Insights and algorithms for the multivariate square-root lasso

Aaron J. Molstad
Department of Statistics and Genetics Institute
University of Florida

June 12th, 2020
Statistical Learning Seminar

Outline of the talk

1. Multivariate response linear regression
2. Considerations in high-dimensional settings
3. The multivariate square-root lasso
   - Motivation/interpretation
   - Theoretical tuning
   - Computation
   - Simulation studies
   - Genomic data example

Multivariate response linear regression model

The multivariate response linear regression model assumes the measured response for the ith subject $y_i \in \mathbb{R}^q$ is a realization of the random vector

$$\mathbf{Y}_i = \beta' x_i + \epsilon_i, \quad (i = 1, \ldots, n),$$

where

- $x_i \in \mathbb{R}^p$ is the p-variate predictor for the ith subject,
- $\beta \in \mathbb{R}^{p \times q}$ is the unknown regression coefficient matrix,
- $\epsilon_i \in \mathbb{R}^q$ are iid random vectors with mean zero and covariance $\mathbf{\Sigma} \equiv \mathbf{\Omega}^{-1} \in \mathbb{S}_+^q$.

Let the observed data be organized into:

- $Y = (y_1, \ldots, y_n)' \in \mathbb{R}^{n \times q}$, $X = (x_1, \ldots, x_n)' \in \mathbb{R}^{n \times p}$.

Multivariate response linear regression model

Most natural estimator when $n > p$ is the least-squares estimator (i.e., squared Frobenius norm):

$$\hat{\beta}_{OLS} = \underset{\beta \in \mathbb{R}^{p \times q}}{\arg\min} \, \| Y - X\beta \|_F^2$$

where $\|A\|_F^2 = \mathrm{tr}(A'A) = \sum_{i,j} A_{i,j}^2$.

Multivariate response linear regression model

Most natural estimator when $n > p$ is the least-squares estimator (i.e., squared Frobenius norm):

$$\hat{\beta}_{OLS} = \underset{\beta \in \mathbb{R}^{p \times q}}{\arg\min} \, \| Y - X\beta \|_F^2$$

where $\|A\|_F^2 = \mathrm{tr}(A'A) = \sum_{i,j} A_{i,j}^2$.

Setting the gradient to zero,

$$X'X\hat{\beta}_{OLS} - X'Y = 0 \implies \hat{\beta}_{OLS} = (X'X)^{-1}X'Y.$$

$\implies$ same estimator we would get if we performed q separate least squares regressions.

Multivariate response linear regression model

If we assume the errors are multivariate normal, then the maximum likelihood estimator is

$$\underset{\beta \in \mathbb{R}^{p \times q}, \Omega \in \mathbb{S}_+^q}{\arg\min} \left[\operatorname{tr} \left\{ n^{-1} (Y - X\beta) \Omega (Y - X\beta)' \right\} - \log \det(\Omega) \right].$$

- Equivalent to least squares only if $\Omega \propto I_q$ is known and fixed.

Multivariate response linear regression model

If we assume the errors are multivariate normal, then the maximum likelihood estimator is

$$\underset{\beta \in \mathbb{R}^{p \times q}, \Omega \in \mathbb{S}_+^q}{\arg\min} \left[\operatorname{tr} \left\{ n^{-1} (Y - X\beta) \Omega (Y - X\beta)' \right\} - \log\det(\Omega) \right].$$

- Equivalent to least squares only if $\Omega \propto I_q$ is known and fixed.

However, the first order optimality conditions for β are

$$X'X\hat{\beta}_{\text{MLE}}\Omega - X'Y\Omega = 0,$$

which implies

$$\hat{\beta}_{\text{MLE}} = (X'X)^{-1}X'Y = \hat{\beta}_{\text{OLS}}.$$

When $n < p$, $\hat{\beta}_{\mathrm{OLS}}$ is non-unique, so we may want to apply some type of shrinkage/regularization, or impose some type of parsimonious parametric restriction.

Estimating β in high-dimensions

When p and q are large, one way to estimate β is to minimize some loss plus a penalty:

$$\underset{\beta \in \mathbb{R}^{p \times q}, \theta \in \Theta}{\arg\min} \left\{ \ell(\beta, \theta) + \lambda \mathcal{P}(\beta) \right\},$$

where the choice of penalty depends on assumptions about β:

- Elementwise sparse: $\mathcal{P}(\beta) = \sum_{j,k} |\beta_{j,k}|$ (Tibshirani, 1996)

Estimating β in high-dimensions

When p and q are large, one way to estimate β is to minimize some loss plus a penalty:

$$\underset{\beta \in \mathbb{R}^{p \times q}, \theta \in \Theta}{\arg\min} \left\{ \ell(\beta, \theta) + \lambda \mathcal{P}(\beta) \right\},$$

where the choice of penalty depends on assumptions about β:

- Elementwise sparse: $\mathcal{P}(\beta) = \sum_{j,k} |\beta_{j,k}|$ (Tibshirani, 1996)
- Row-wise sparse: $\mathcal{P}(\beta) = \sum_{j=1}^{p} \|\beta_{j,\cdot}\|_2$ (Yuan and Lin, 2007; Obozinski et al., 2011)

Estimating β in high-dimensions

When p and q are large, one way to estimate β is to minimize some loss plus a penalty:

$$\underset{\beta \in \mathbb{R}^{p \times q}, \theta \in \Theta}{\arg\min} \left\{ \ell(\beta, \theta) + \lambda \mathcal{P}(\beta) \right\},$$

where the choice of penalty depends on assumptions about β:

- Elementwise sparse: $\mathcal{P}(\beta) = \sum_{j,k} |\beta_{j,k}|$ (Tibshirani, 1996)
- Row-wise sparse: $\mathcal{P}(\beta) = \sum_{j=1}^{p} \|\beta_{j,\cdot}\|_2$ (Yuan and Lin, 2007; Obozinski et al., 2011)
- "Bi-level" sparse: $\mathcal{P}(\beta) = \alpha \sum_{j,k} |\beta_{j,k}| + (1 - \alpha) \sum_{j=1}^{p} \|\beta_{j,\cdot}\|_2$ (Peng et al., 2012; Simon et al., 2013)

Estimating β in high-dimensions

When p and q are large, one way to estimate β is to minimize some loss plus a penalty:

$$\underset{\beta \in \mathbb{R}^{p \times q}, \theta \in \Theta}{\arg\min} \left\{ \ell(\beta, \theta) + \lambda \mathcal{P}(\beta) \right\},$$

where the choice of penalty depends on assumptions about β:

- Elementwise sparse: $\mathcal{P}(\beta) = \sum_{j,k} |\beta_{j,k}|$ (Tibshirani, 1996)
- Row-wise sparse: $\mathcal{P}(\beta) = \sum_{j=1}^{p} \|\beta_{j,\cdot}\|_2$ (Yuan and Lin, 2007; Obozinski et al., 2011)
- "Bi-level" sparse: $\mathcal{P}(\beta) = \alpha \sum_{j,k} |\beta_{j,k}| + (1-\alpha) \sum_{j=1}^{p} \|\beta_{j,\cdot}\|_2$ (Peng et al., 2012; Simon et al., 2013)
- Low-rank: $\mathcal{P}(\beta) = \|\beta\|_* = \sum_{j=1}^{\min(p,q)} \varphi_j(\beta)$ or $\mathcal{P}(\beta) = \mathrm{Rank}(\beta)$ (Yuan et al., 2007; Bunea et al., 2011; Chen et al., 2013)

Can we ignore the error covariance in these high-dimensional settings?

High-dimensional maximum likelihood

The penalized normal maximum likelihood estimator with $\mathbf{\Omega}$ known is

$$\hat{\beta}_{\mathcal{P}} \in \underset{\beta \in \mathbb{R}^{p \times q}}{\arg\min} \left[\operatorname{tr} \left\{ n^{-1} (Y - X\beta) \mathbf{\Omega} (Y - X\beta)' \right\} + \lambda \mathcal{P}(\beta) \right].$$

High-dimensional maximum likelihood

The penalized normal maximum likelihood estimator with $\mathbf{\Omega}$ known is

$$\hat{\beta}_{\mathcal{P}} \in \underset{\beta \in \mathbb{R}^{p \times q}}{\arg\min} \left[\operatorname{tr} \left\{ n^{-1} (Y - X\beta) \mathbf{\Omega} (Y - X\beta)' \right\} + \lambda \mathcal{P}(\beta) \right].$$

Then, the first order optimality conditions are:

$$X'X\hat{\beta}_{\mathcal{P}}\mathbf{\Omega} - X'Y\mathbf{\Omega} + \lambda \partial \mathcal{P}(\hat{\beta}_{\mathcal{P}}) \ni 0,$$

where $\partial \mathcal{P}(\hat{\beta}_{\mathcal{P}})$ is the subgradient of $\mathcal{P}$ evaluated at $\hat{\beta}_{\mathcal{P}}$.

High-dimensional maximum likelihood

The penalized normal maximum likelihood estimator with $\mathbf{\Omega}$ known is

$$\hat{\beta}_{\mathcal{P}} \in \underset{\beta \in \mathbb{R}^{p \times q}}{\arg\min} \left[\operatorname{tr} \left\{ n^{-1} (Y - X\beta) \mathbf{\Omega} (Y - X\beta)' \right\} + \lambda \mathcal{P}(\beta) \right].$$

Then, the first order optimality conditions are:

$$X'X\hat{\beta}_{\mathcal{P}}\mathbf{\Omega} - X'Y\mathbf{\Omega} + \lambda \partial \mathcal{P}(\hat{\beta}_{\mathcal{P}}) \ni 0,$$

where $\partial \mathcal{P}(\hat{\beta}_{\mathcal{P}})$ is the subgradient of $\mathcal{P}$ evaluated at $\hat{\beta}_{\mathcal{P}}$.

$\implies$ $\hat{\beta}_{\mathcal{P}}$, the shrinkage estimator, depends on $\mathbf{\Omega}$.

High-dimensional maximum likelihood

The penalized normal maximum likelihood estimator with $\mathbf{\Omega}$ known is

$$\hat{\beta}_{\mathcal{P}} \in \underset{\beta \in \mathbb{R}^{p \times q}}{\arg\min} \left[\operatorname{tr} \left\{ n^{-1} (Y - X\beta) \mathbf{\Omega} (Y - X\beta)' \right\} + \lambda \mathcal{P}(\beta) \right].$$

Then, the first order optimality conditions are:

$$X'X\hat{\beta}_{\mathcal{P}}\mathbf{\Omega} - X'Y\mathbf{\Omega} + \lambda \partial \mathcal{P}(\hat{\beta}_{\mathcal{P}}) \ni 0,$$

where $\partial \mathcal{P}(\hat{\beta}_{\mathcal{P}})$ is the subgradient of $\mathcal{P}$ evaluated at $\hat{\beta}_{\mathcal{P}}$.

$\implies$ $\hat{\beta}_{\mathcal{P}}$, the shrinkage estimator, depends on $\mathbf{\Omega}$.

Equivalent to penalized least squares if $\mathbf{\Omega} \propto I_q$.

Can we ignore the error covariance in these high-dimensional settings?

No! But of course, Ω is unknown in practice.

Penalized normal maximum likelihood

When $\mathbf{\Omega}$ is unknown and the ϵ_i are normally distributed, the (doubly) penalized maximum likelihood estimator is:

$$\underset{\beta \in \mathbb{R}^{p \times q}, \Omega \in \mathbb{S}_+^q}{\arg\min} \left\{ \mathcal{F}(\beta, \Omega) + \lambda_\beta \mathcal{P}_\beta(\beta) + \lambda_\Omega \mathcal{P}_\Omega(\Omega) \right\},$$

where

$$\mathcal{F}(\beta, \Omega) = \operatorname{tr}\left\{ n^{-1}(Y - X\beta)\Omega(Y - X\beta)' \right\} - \log\det(\Omega).$$

Penalized normal maximum likelihood

When $\mathbf{\Omega}$ is unknown and the ϵ_i are normally distributed, the (doubly) penalized maximum likelihood estimator is:

$$\underset{\beta \in \mathbb{R}^{p \times q}, \Omega \in \mathbb{S}_+^q}{\arg\min} \left\{ \mathcal{F}(\beta, \Omega) + \lambda_\beta \mathcal{P}_\beta(\beta) + \lambda_\Omega \mathcal{P}_\Omega(\Omega) \right\},$$

where

$$\mathcal{F}(\beta, \Omega) = \operatorname{tr}\left\{ n^{-1}(Y - X\beta)\Omega(Y - X\beta)' \right\} - \log\det(\Omega).$$

- Rothman et al. (2010) and Yin and Li (2011) use ℓ_1-penalties on both β and Ω.

Penalized normal maximum likelihood

When $\mathbf{\Omega}$ is unknown and the ϵ_i are normally distributed, the (doubly) penalized maximum likelihood estimator is:

$$\underset{\beta \in \mathbb{R}^{p \times q}, \Omega \in \mathbb{S}_+^q}{\arg\min} \left\{ \mathcal{F}(\beta, \Omega) + \lambda_\beta \mathcal{P}_\beta(\beta) + \lambda_\Omega \mathcal{P}_\Omega(\Omega) \right\},$$

where

$$\mathcal{F}(\beta, \Omega) = \operatorname{tr}\left\{ n^{-1}(Y - X\beta)\Omega(Y - X\beta)' \right\} - \log\det(\Omega).$$

- Rothman et al. (2010) and Yin and Li (2011) use ℓ_1-penalties on both β and Ω.
- Chen and Huang (2016) impose low-rank and sparsity inducing penalties on β.
- ...

Penalized normal maximum likelihood

When $\mathbf{\Omega}$ is unknown and the ϵ_i are normally distributed, the (doubly) penalized maximum likelihood estimator is:

$$\underset{\beta \in \mathbb{R}^{p \times q}, \Omega \in \mathbb{S}_+^q}{\arg\min} \left\{ \mathcal{F}(\beta, \Omega) + \lambda_\beta \mathcal{P}_\beta(\beta) + \lambda_\Omega \mathcal{P}_\Omega(\Omega) \right\},$$

where

$$\mathcal{F}(\beta, \Omega) = \operatorname{tr}\left\{ n^{-1}(Y - X\beta)\Omega(Y - X\beta)' \right\} - \log\det(\Omega).$$

If estimating β is the primary goal, we may want to avoid these methods since they can require

Penalized normal maximum likelihood

When $\mathbf{\Omega}$ is unknown and the ϵ_i are normally distributed, the (doubly) penalized maximum likelihood estimator is:

$$\underset{\beta \in \mathbb{R}^{p \times q}, \Omega \in \mathbb{S}_+^q}{\arg\min} \left\{ \mathcal{F}(\beta, \Omega) + \lambda_\beta \mathcal{P}_\beta(\beta) + \lambda_\Omega \mathcal{P}_\Omega(\Omega) \right\},$$

where

$$\mathcal{F}(\beta, \Omega) = \operatorname{tr}\left\{ n^{-1}(Y - X\beta)\Omega(Y - X\beta)' \right\} - \log\det(\Omega).$$

If estimating β is the primary goal, we may want to avoid these methods since they can require

- estimating $O(q^2)$ nuisance parameters,

Penalized normal maximum likelihood

When $\mathbf{\Omega}$ is unknown and the ϵ_i are normally distributed, the (doubly) penalized maximum likelihood estimator is:

$$\underset{\beta \in \mathbb{R}^{p \times q}, \Omega \in \mathbb{S}_+^q}{\arg\min} \left\{ \mathcal{F}(\beta, \Omega) + \lambda_\beta \mathcal{P}_\beta(\beta) + \lambda_\Omega \mathcal{P}_\Omega(\Omega) \right\},$$

where

$$\mathcal{F}(\beta, \Omega) = \operatorname{tr}\left\{ n^{-1}(Y - X\beta)\Omega(Y - X\beta)' \right\} - \log\det(\Omega).$$

If estimating β is the primary goal, we may want to avoid these methods since they can require

- estimating $O(q^2)$ nuisance parameters,
- solving a non-convex optimization problem,

Penalized normal maximum likelihood

When $\mathbf{\Omega}$ is unknown and the ϵ_i are normally distributed, the (doubly) penalized maximum likelihood estimator is:

$$\underset{\beta \in \mathbb{R}^{p \times q}, \Omega \in \mathbb{S}_+^q}{\arg\min} \left\{ \mathcal{F}(\beta, \Omega) + \lambda_\beta \mathcal{P}_\beta(\beta) + \lambda_\Omega \mathcal{P}_\Omega(\Omega) \right\},$$

where

$$\mathcal{F}(\beta, \Omega) = \operatorname{tr}\left\{ n^{-1}(Y - X\beta)\Omega(Y - X\beta)' \right\} - \log\det(\Omega).$$

If estimating β is the primary goal, we may want to avoid these methods since they can require

- estimating $O(q^2)$ nuisance parameters,
- solving a non-convex optimization problem,
- **extremely** long computing times.

Can we estimate β in high-dimensional settings and account for the error dependence

1. without sacrificing convexity,
2. without requiring an explicit estimate of the error precision matrix?

Multivariate square-root lasso

The *multivariate square-root lasso* (MSRL) estimator is:

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

where

- $\|A\|_* = \mathrm{tr}\left\{(A'A)^{1/2}\right\} = \sum \varphi_j(A)$ is the nuclear norm (trace norm) which sums the singular values of its matrix argument,

Multivariate square-root lasso

The *multivariate square-root lasso* (MSRL) estimator is:

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

where

- $\|A\|_* = \mathrm{tr}\left\{(A'A)^{1/2}\right\} = \sum \varphi_j(A)$ is the nuclear norm (trace norm) which sums the singular values of its matrix argument,
- $\mathcal{P}$ is a convex penalty function, so that (1) is convex.

Past work and our contributions

The ℓ_1-penalized version of MSRL was proposed by van de Geer (2016) and van de Geer and Stucky (2016) as a means for constructing confidence intervals for univariate response regression coefficients.

Past work and our contributions

The ℓ_1-penalized version of MSRL was proposed by van de Geer (2016) and van de Geer and Stucky (2016) as a means for constructing confidence intervals for univariate response regression coefficients.

Our contributions:

- establish statistical theory and as a consequence, a new direct tuning procedure,
- propose two specialized algorithms with convergence guarantees to compute MSRL,
- demonstrate usefulness of MSRL for multivariate response linear regression with dependent errors.

Remainder of the talk

1. Review of multivariate response linear regression ✓
2. Considerations in high dimensions ✓
3. The multivariate square-root lasso
   - Motivation/interpretation
   - Theoretical tuning
   - Computation
   - Simulation studies
   - Genomic data example

Why the nuclear norm?

Motivation

$$\|A\|_* = \mathrm{tr}\left\{(A'A)^{1/2}\right\} = \mathrm{tr}\left\{A(A'A)^{-1/2}A'\right\}.$$

Motivation

$$\|A\|_* = \text{tr}\left\{(A'A)^{1/2}\right\} = \text{tr}\left\{A(A'A)^{-1/2}A'\right\}.$$

Hence, we can write

$$\frac{1}{\sqrt{n}}\|Y - X\beta\|_* = \text{tr}\left\{n^{-1}(Y - X\beta)\Omega_\beta^{1/2}(Y - X\beta)'\right\}$$

where

$$\Omega_\beta = \left\{n^{-1}(Y - X\beta)'(Y - X\beta)\right\}^{-1}.$$

Motivation

$$\|A\|_* = \text{tr}\left\{(A'A)^{1/2}\right\} = \text{tr}\left\{A(A'A)^{-1/2}A'\right\}.$$

Hence, we can write

$$\frac{1}{\sqrt{n}}\|Y - X\beta\|_* = \text{tr}\left\{n^{-1}(Y - X\beta)\Omega_\beta^{1/2}(Y - X\beta)'\right\}$$

where

$$\Omega_\beta = \left\{n^{-1}(Y - X\beta)'(Y - X\beta)\right\}^{-1}.$$

Nuclear norm of residuals is a weighted residual sum of squares where the weight is

- an estimate of the square-root of the error precision matrix,
- a function of the optimization variable β only.

Multivariate square-root lasso

Define $(\bar{\beta}, \bar{\Sigma})$ as

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

Let $\hat{\beta}$ be the multivariate square-root lasso estimator with the same tuning parameter.

Multivariate square-root lasso

Define $(\bar{\beta}, \bar{\Sigma})$ as

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

Let $\hat{\beta}$ be the multivariate square-root lasso estimator with the same tuning parameter.

If the residual matrix $Y - X\hat{\beta}$ has q nonzero singular values, then $(\bar{\beta}, \bar{\Sigma})$ satisfies

$$\bar{\Sigma} = n^{-1}(Y - X\hat{\beta})'(Y - X\hat{\beta}) \quad \text{and} \quad \bar{\beta} = \hat{\beta}.$$

Multivariate square-root lasso

Define $(\bar{\beta}, \bar{\Sigma})$ as

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

Let $\hat{\beta}$ be the multivariate square-root lasso estimator with the same tuning parameter.

If the residual matrix $Y - X\hat{\beta}$ has q nonzero singular values, then $(\bar{\beta}, \bar{\Sigma})$ satisfies

$$\bar{\Sigma} = n^{-1}(Y - X\hat{\beta})'(Y - X\hat{\beta}) \quad \text{and} \quad \bar{\beta} = \hat{\beta}.$$

MSRL solves the joint optimization problem!

Relation to univariate square-root lasso

MSRL generalizes the univariate square-root lasso (Owen, 2007; Belloni et al., 2011), i.e., when $q = 1$ so that $y \in \mathbb{R}^n$, MSRL is equivalent to

$$\underset{\eta \in \mathbb{R}^p}{\arg\min} \left\{ \sqrt{\frac{1}{n}\|y - X\eta\|_2^2} + \gamma\|\eta\|_1 \right\}. \tag{2}$$

Relation to univariate square-root lasso

MSRL generalizes the univariate square-root lasso (Owen, 2007; Belloni et al., 2011), i.e., when $q = 1$ so that $y \in \mathbb{R}^n$, MSRL is equivalent to

$$\underset{\eta \in \mathbb{R}^p}{\arg\min} \left\{ \sqrt{\frac{1}{n}\|y - X\eta\|_2^2} + \gamma\|\eta\|_1 \right\}. \tag{2}$$

- Belloni et al. (2011) proved that value of γ leading to near oracle error bounds does not depend on any unknown quantities, unlike ℓ_1-penalized least squares.

Relation to univariate square-root lasso

MSRL generalizes the univariate square-root lasso (Owen, 2007; Belloni et al., 2011), i.e., when $q = 1$ so that $y \in \mathbb{R}^n$, MSRL is equivalent to

$$\underset{\eta \in \mathbb{R}^p}{\arg\min} \left\{ \sqrt{\frac{1}{n}\|y - X\eta\|_2^2} + \gamma \|\eta\|_1 \right\}. \tag{2}$$

- Belloni et al. (2011) proved that value of γ leading to near oracle error bounds does not depend on any unknown quantities, unlike ℓ_1-penalized least squares.
- Many extensions and improvements of (2), e.g., Bunea, Lederer, and She (2013); Liu, Wang, and Zhao (2015); Ndiaye et al. (2016); Tian et al. (2019).

Assumptions

Definitions:

- Let s denote the number of nonzero entries in β
- $\mathcal{P}(\beta) = \sum_{j,k} |\beta_{j,k}|$
- Normalize predictors so that $\|X_j\|_2 = 1$ for $j = 1, \ldots, p$
- Define $\|A\|_{\max} = \max_{j,k} |A_{j,k}|$

Assumptions

Definitions:

- Let s denote the number of nonzero entries in β
- $\mathcal{P}(\beta) = \sum_{j,k} |\beta_{j,k}|$
- Normalize predictors so that $\|X_j\|_2 = 1$ for $j = 1, \ldots, p$
- Define $\|A\|_{\max} = \max_{j,k} |A_{j,k}|$

Assumptions:

A1. $\mathcal{E}$, the $n \times q$ error matrix (i.e., $Y = X\beta + \mathcal{E}$), has q nonzero singular values almost surely.

A2. The error matrix $\mathcal{E}$ is left-spherical, i.e., for any $n \times n$ orthogonal matrix O, $O\mathcal{E}$ has the same matrix-variate distribution as $\mathcal{E}$.

Assumptions

Definitions:

- Let s denote the number of nonzero entries in β
- $\mathcal{P}(\beta) = \sum_{j,k} |\beta_{j,k}|$
- Normalize predictors so that $\|X_j\|_2 = 1$ for $j = 1, \ldots, p$
- Define $\|A\|_{\max} = \max_{j,k} |A_{j,k}|$

Assumptions:

A1. $\mathcal{E}$, the $n \times q$ error matrix (i.e., $Y = X\beta + \mathcal{E}$), has q nonzero singular values almost surely.

A2. The error matrix $\mathcal{E}$ is left-spherical, i.e., for any $n \times n$ orthogonal matrix O, $O\mathcal{E}$ has the same matrix-variate distribution as $\mathcal{E}$.

- A1 and A2 would hold if $n > q$ and rows of $\mathcal{E}$ were iid, mean zero multivariate normal random vectors with covariance $\mathbf{\Sigma} \in \mathbb{S}_+^q$.

Frobenius norm error bound

Proposition: Suppose A1 is true. Let $U_* D_* V_*' = Y - X\beta$ be the singular value decomposition. If $\lambda \geq \frac{c}{\sqrt{n}} \|X' U_* V_*'\|_{\max}$ for some constant $c > 1$, then

$$\|\hat{\beta} - \beta\|_F \leq \frac{\bar{c}}{\kappa(\mathcal{E}, c)} \lambda \sqrt{s},$$

where $\bar{c} = (c+1)/(c-1)$.

Frobenius norm error bound

Proposition: Suppose A1 is true. Let $U_* D_* V_*' = Y - X\beta$ be the singular value decomposition. If $\lambda \geq \frac{c}{\sqrt{n}} \|X' U_* V_*'\|_{\max}$ for some constant $c > 1$, then

$$\|\hat{\beta} - \beta\|_F \leq \frac{\bar{c}}{\kappa(\mathcal{E}, c)} \lambda \sqrt{s},$$

where $\bar{c} = (c+1)/(c-1)$.

Empirically:

$$\kappa(\mathcal{E}, c) \geq \frac{k_c}{M \varphi_1(\mathbf{\Sigma})}$$

where k_c is the restricted eigenvalue, $\varphi_1(\mathbf{\Sigma})$ is largest eigenvalue of error covariance matrix, and M is some positive constant.

Frobenius norm error bound

Proposition: Suppose A1 is true. Let $U_* D_* V_*' = Y - X\beta$ be the singular value decomposition. If $\lambda \geq \frac{c}{\sqrt{n}} \|X' U_* V_*'\|_{\max}$ for some constant $c > 1$, then

$$\|\hat{\beta} - \beta\|_F \leq \frac{\bar{c}}{\kappa(\mathcal{E}, c)} \lambda \sqrt{s},$$

where $\bar{c} = (c+1)/(c-1)$. If A2 is also true, then the distribution of $\|X' U_* V_*'\|_{\max}$ does not depend on Ω.

Frobenius norm error bound

Proposition: Suppose A1 is true. Let $U_* D_* V_*' = Y - X\beta$ be the singular value decomposition. If $\lambda \geq \frac{c}{\sqrt{n}} \|X' U_* V_*'\|_{\max}$ for some constant $c > 1$, then

$$\|\hat{\beta} - \beta\|_F \leq \frac{\bar{c}}{\kappa(\mathcal{E}, c)} \lambda \sqrt{s},$$

where $\bar{c} = (c+1)/(c-1)$. If A2 is also true, then the distribution of $\|X' U_* V_*'\|_{\max}$ does not depend on Ω.

Under A1 and A2, $U_* V_*'$ has a uniform distribution on the set of $n \times q$ semiorthogonal matrices (e.g., Eaton, (1989)).

Frobenius norm error bound

Proposition: Suppose A1 is true. Let $U_* D_* V_*' = Y - X\beta$ be the singular value decomposition. If $\lambda \geq \frac{c}{\sqrt{n}} \|X' U_* V_*'\|_{\max}$ for some constant $c > 1$, then

$$\|\hat{\beta} - \beta\|_F \leq \frac{\bar{c}}{\kappa(\mathcal{E}, c)} \lambda \sqrt{s},$$

where $\bar{c} = (c+1)/(c-1)$. If A2 is also true, then the distribution of $\|X' U_* V_*'\|_{\max}$ does not depend on $\mathbf{\Omega}$.

Under A1 and A2, $U_* V_*'$ has a uniform distribution on the set of $n \times q$ semiorthogonal matrices (e.g., Eaton, (1989)).

- We can use simulation to approximate the distribution of $\frac{c}{\sqrt{n}} \|X' U_* V_*'\|_{\max}$, and set λ equal to some empirical quantile (e.g., 95th).

Frobenius norm error bound

Theorem: Suppose A1 and A2 are true. If n is sufficiently large wrt q and $\alpha \in (0,1)$; and $\lambda = 3\left\{2n^{-1}\log(4pq/\alpha)\right\}^{1/2}$, then

$$\|\hat{\beta} - \beta\|_F \leq \frac{9}{\kappa(\mathcal{E},2)}\sqrt{\frac{2s\log(4pq/\alpha)}{n}},$$

with probability at least $1-\alpha$.

Frobenius norm error bound

Theorem: Suppose A1 and A2 are true. If n is sufficiently large wrt q and $\alpha \in (0,1)$; and $\lambda = 3\left\{2n^{-1}\log(4pq/\alpha)\right\}^{1/2}$, then

$$\|\hat{\beta} - \beta\|_F \leq \frac{9}{\kappa(\mathcal{E},2)}\sqrt{\frac{2s\log(4pq/\alpha)}{n}},$$

with probability at least $1-\alpha$.

Unlike penalized squared Frobenius norm estimator, tuning parameter λ does not depend on any unknown quantities!

What's left?

1. Review of multivariate response linear regression ✓
2. Considerations in high dimensions ✓
3. The multivariate square-root lasso ✓
   - Motivation/interpretation ✓
   - Theoretical tuning ✓
   - Computation
   - Simulation studies
   - Genomic data example

Constrained problem

Computing MSRL is difficult because, although convex, it is the sum of two non-differentiable functions:

$$\underset{\beta \in \mathbb{R}^{p \times q}}{\arg\min} \left\{ \| Y - X\beta \|_* + \tilde{\lambda} \mathcal{P}(\beta) \right\}.$$

Constrained problem

Computing MSRL is difficult because, although convex, it is the sum of two non-differentiable functions:

$$\underset{\beta \in \mathbb{R}^{p \times q}}{\arg\min} \left\{ \| Y - X\beta \|_* + \tilde{\lambda} \mathcal{P}(\beta) \right\}.$$

We split the two functions by rewriting MSRL as a constrained optimization problem:

$$\underset{\beta \in \mathbb{R}^{p \times q}, \Phi \in \mathbb{R}^{n \times q}}{\text{minimize}} \left\{ \| \Phi \|_* + \tilde{\lambda} \mathcal{P}(\beta) \right\} \quad \text{subject to} \quad \Phi = Y - X\beta,$$

and solve the constrained problem using dual ascent via alternating direction methods of multipliers (ADMM).

Prox-linear ADMM

Let $\Gamma \in \mathbb{R}^{n \times q}$ be a Lagrangian dual variable and $\rho > 0$ a step size parameter. The augmented Lagrangian for the constrained problem is:

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

Prox-linear ADMM

Let $\Gamma \in \mathbb{R}^{n \times q}$ be a Lagrangian dual variable and $\rho > 0$ a step size parameter. The augmented Lagrangian for the constrained problem is:

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

Following Boyd et al. (2011), the updating equations of ADMM are:

$$\Phi_{k+1} = \underset{\Phi \in \mathbb{R}^{n \times q}}{\arg\min}\, \mathcal{G}_\rho(\beta_k, \Phi, \Gamma_k)$$

$$\beta_{k+1} = \underset{\beta \in \mathbb{R}^{p \times q}}{\arg\min}\, \mathcal{G}_\rho(\beta, \Phi_{k+1}, \Gamma_k)$$

$$\Gamma_{k+1} = \Gamma_k + \rho(Y - X\beta_{k+1} - \Phi_{k+1}),$$

Prox-linear ADMM

Let $\Gamma \in \mathbb{R}^{n \times q}$ be a Lagrangian dual variable and $\rho > 0$ a step size parameter. The augmented Lagrangian for the constrained problem is:

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

Following Boyd et al. (2011), the updating equations of ADMM are:

$$\Phi_{k+1} = \underset{\Phi \in \mathbb{R}^{n \times q}}{\arg\min}\, \mathcal{G}_\rho(\beta_k, \Phi, \Gamma_k)$$

$$\beta_{k+1} = \underset{\beta \in \mathbb{R}^{p \times q}}{\arg\min}\, \mathcal{G}_\rho(\beta, \Phi_{k+1}, \Gamma_k)$$

$$\Gamma_{k+1} = \Gamma_k + \rho(Y - X\beta_{k+1} - \Phi_{k+1}),$$

Prox-linear ADMM

Let $\Gamma \in \mathbb{R}^{n \times q}$ be a Lagrangian dual variable and $\rho > 0$ a step size parameter. The augmented Lagrangian for the constrained problem is:

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

Following Boyd et al. (2011), the updating equations of ADMM are:

$$\Phi_{k+1} = \underset{\Phi \in \mathbb{R}^{n \times q}}{\arg\min}\, \mathcal{G}_\rho(\beta_k, \Phi, \Gamma_k)$$

$$\beta_{k+1} = \underset{\beta \in \mathbb{R}^{p \times q}}{\arg\min}\, \mathcal{G}_\rho(\beta, \Phi_{k+1}, \Gamma_k)$$

$$\Gamma_{k+1} = \Gamma_k + \rho(Y - X\beta_{k+1} - \Phi_{k+1}),$$

Prox-linear ADMM

We approximate $\mathcal{G}_\rho(\beta, \Phi_{k+1}, \Gamma_k)$ at β_k with

$$\mathcal{M}_{\rho,\eta}(\beta, \Phi_{k+1}, \Gamma_k; \beta_k) \equiv \mathcal{G}_\rho(\beta, \Phi_{k+1}, \Gamma_k) + \frac{\rho}{2} \operatorname{tr}\{(\beta - \beta_k)'(\eta I_p - X'X)(\beta - \beta_k)\},$$

with $\eta \in \mathbb{R}$ chosen so that $\eta I_p - X'X \succeq 0$.

Prox-linear ADMM

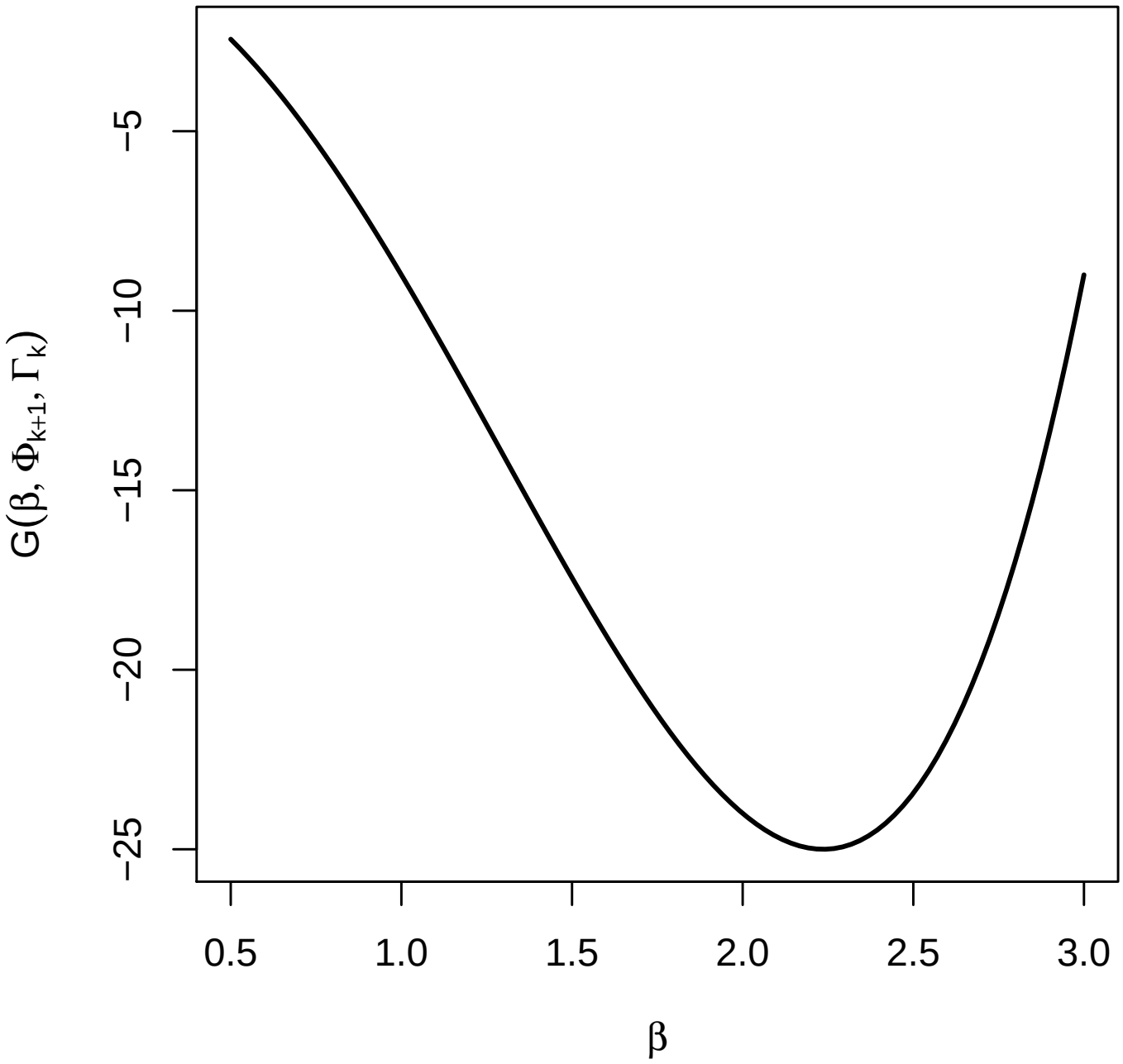
We approximate $\mathcal{G}_\rho(\beta, \Phi_{k+1}, \Gamma_k)$ at β_k with

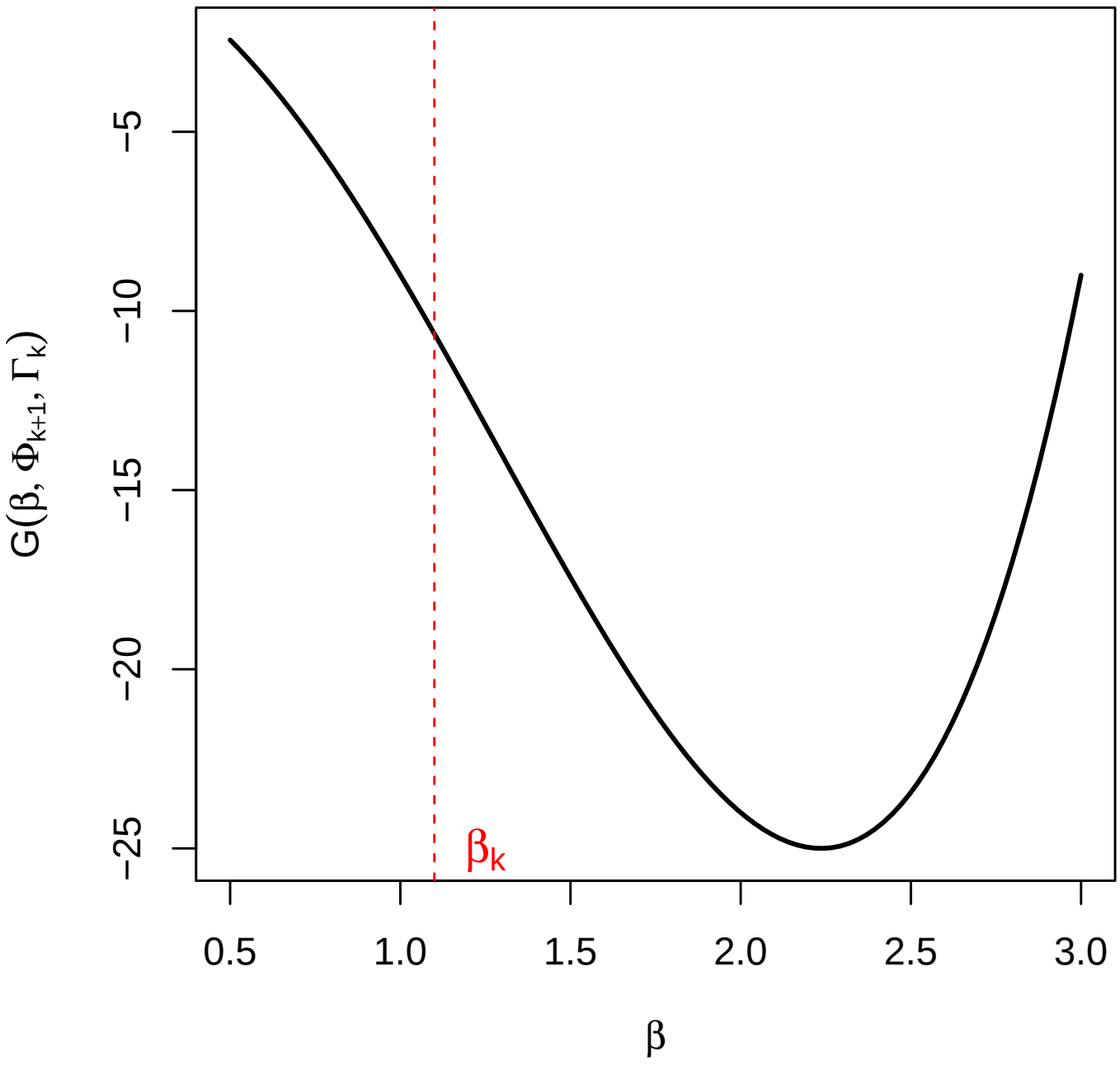
$$\mathcal{M}_{\rho,\eta}(\beta, \Phi_{k+1}, \Gamma_k; \beta_k) \equiv \mathcal{G}_\rho(\beta, \Phi_{k+1}, \Gamma_k) + \frac{\rho}{2} \operatorname{tr}\{(\beta - \beta_k)'(\eta I_p - X'X)(\beta - \beta_k)\},$$

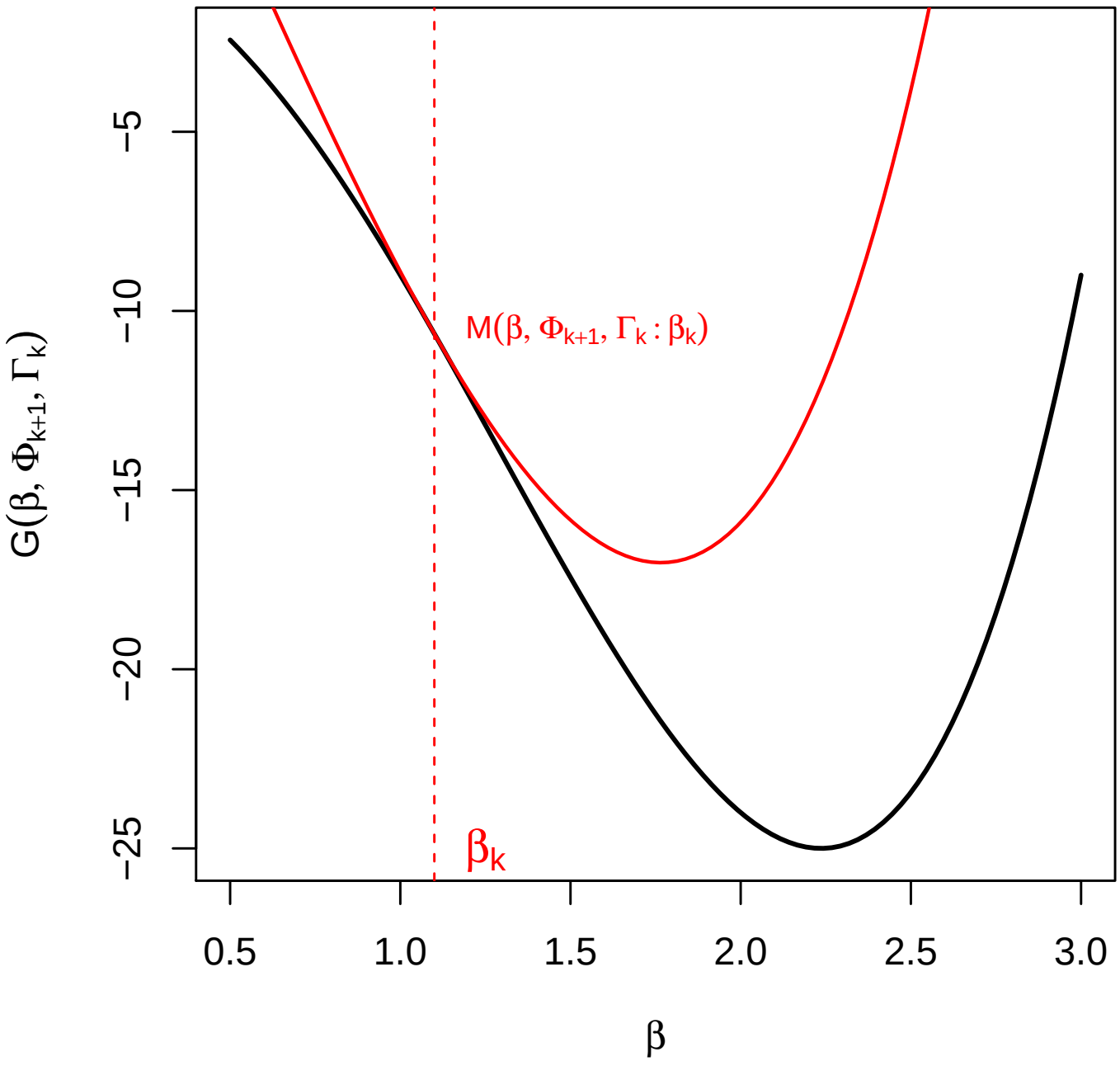
with $\eta \in \mathbb{R}$ chosen so that $\eta I_p - X'X \succeq 0$. Then, we update

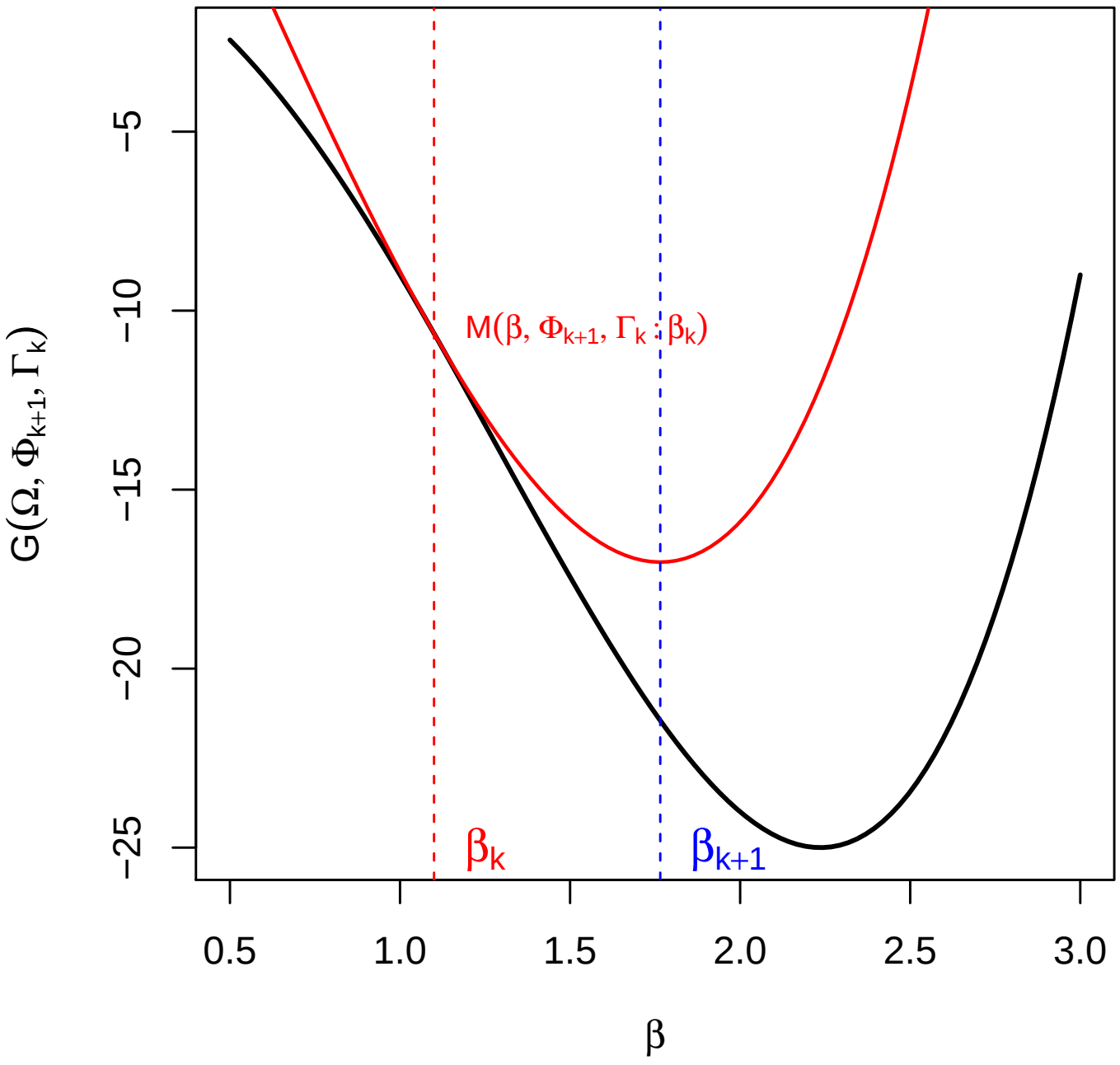
$$\begin{aligned} \beta_{k+1} &= \underset{\beta \in \mathbb{R}^{p \times q}}{\arg\min}\, \mathcal{M}_{\rho,\eta}(\beta, \Phi_{k+1}, \Gamma_k; \beta_k) \\ &= \underset{\beta \in \mathbb{R}^{p \times q}}{\arg\min} \left\{ \frac{1}{2} \|\beta - Z_k\|_2^2 + \tilde{\lambda} \mathcal{P}(\beta) \right\} \end{aligned}$$

where $Z_k = \beta_k + \eta^{-1} X' \left(Y + \rho^{-1} \Gamma_k - \Phi_{k+1} - X\beta_k \right)$, which can be solved in closed form for all aforementioned penalties.


G(β, Φ_{k+1}, Γ_k)
−5
−10
−15
−20
−25
0.5
1.0
1.5
2.0
2.5
3.0
β


G(β, Φ_{k+1}, Γ_k)
−5
−10
−15
−20
−25
β_k
0.5
1.0
1.5
2.0
2.5
3.0
β


G(β, Φ_{k+1}, Γ_k)
−5
−10
−15
−20
−25
M(β, Φ_{k+1}, Γ_k : β_k)
β_k
0.5
1.0
1.5
2.0
2.5
3.0
β


G(Ω, Φk+1, Γk)
−5
−10
−15
−20
−25
M(β, Φk+1, Γk : βk)
βk
βk+1
0.5
1.0
1.5
2.0
2.5
3.0
β

Prox-linear ADMM

By the majorize-minimize principle,

$$\begin{aligned}\mathcal{G}_\rho(\beta_{k+1}, \Phi_{k+1}, \Gamma_k) &\leq \mathcal{M}_{\rho,\eta}(\beta_{k+1}, \Phi_{k+1}, \Gamma_k; \beta_k) \\ &\leq \mathcal{M}_{\rho,\eta}(\beta_k, \Phi_{k+1}, \Gamma_k; \beta_k) \\ &= \mathcal{G}_\rho(\beta_k, \Phi_{k+1}, \Gamma_k),\end{aligned}$$

so that we are guaranteed a non-increasing augmented Lagrangian with this simple approximation scheme.

Prox-linear ADMM

By the majorize-minimize principle,

$$\begin{aligned}\mathcal{G}_\rho(\beta_{k+1}, \Phi_{k+1}, \Gamma_k) &\leq \mathcal{M}_{\rho,\eta}(\beta_{k+1}, \Phi_{k+1}, \Gamma_k; \beta_k) \\ &\leq \mathcal{M}_{\rho,\eta}(\beta_k, \Phi_{k+1}, \Gamma_k; \beta_k) \\ &= \mathcal{G}_\rho(\beta_k, \Phi_{k+1}, \Gamma_k),\end{aligned}$$

so that we are guaranteed a non-increasing augmented Lagrangian with this simple approximation scheme.

Proposition: Iterates of our proposed ADMM with the MM-approximation are guaranteed to converge to their optimal values.

Algorithm summary for ℓ_1-penalized version

1. Decompose with SVD: $UDV' = Y + \rho^{-1}\Gamma_k - X\beta_k$
2. Compute $\Phi_{k+1} \leftarrow U \operatorname{Diag}\left[\max(D - \rho^{-1}, 0)\right] V'$
3. Compute $Z_k \leftarrow \beta_k + \eta^{-1}X'\left(Y + \rho^{-1}\Gamma_k - \Phi_{k+1} - X\beta_k\right)$
4. For all $(l, m) \in [1, \ldots p] \times [1, \ldots, q]$

   Compute $[\beta_{k+1}]_{l,m} \leftarrow \max(|[Z_k]_{l,m}| - \tilde{\lambda}, 0) \operatorname{sign}([Z_k]_{l,m})$
5. Compute $\Gamma_{k+1} \leftarrow \Gamma_k + \rho(Y - X\beta_{k+1} - \Phi_{k+1})$
6. If not converged, set $k \leftarrow k + 1$ and return to 1.

- Everything can be computed in closed form assuming proximal operator of $\mathcal{P}$ has closed form.
- github.com/ajmolstad/MSRL

Comparison to off-the-shelf solver

Prior way to compute MSRL was the generic solver `CVX`, which was used by van de Geer and Stucky (2016).

	Normal errors		t_5 errors	
ξ	ADMM	CVX	ADMM	CVX
0.30	3.74	405.62	4.57	546.55
0.50	3.84	500.26	3.94	482.23
0.70	3.40	512.32	3.69	506.57
0.90	2.43	436.76	2.94	531.84
0.95	2.28	460.67	2.28	459.05

Table: Average computing times (in secs) using prox-linear ADMM versus CVX with $n = 200$, $p = 500$, and $q = 50$ with errors having correlation matrix $\Sigma_{*j,k} = \xi \mathbf{1}(j \neq k) + \mathbf{1}(j = k)$.

Simulations

For 100 independent replications with $p = 500$ and $q = 50$

- Generate $X \sim \mathrm{N}_{500}(0, \Sigma_{*X})$ where $[\Sigma_{*X}]_{j,k} = .5^{|j-k|}$;
- Given $X = x$, generate $y = \beta' x + \epsilon$ where

$$\epsilon \sim \mathrm{N}_{50}(0, \mathbf{\Sigma}),$$

  with $\mathbf{\Sigma} = D\tilde{\Sigma}D$ where D is diagonal with entries equally spaced from 3 to .50 and $[\tilde{\Sigma}]_{j,k} = \xi \mathbf{1}(j = k) + \mathbf{1}(j \neq k)$;
- Each column of β has 5 randomly selected entries set equal to -1 or 1, zeros elsewhere.

Simulations

For 100 independent replications with $p = 500$ and $q = 50$

- Generate $X \sim \mathrm{N}_{500}(0, \Sigma_{*X})$ where $[\Sigma_{*X}]_{j,k} = .5^{|j-k|}$;
- Given $X = x$, generate $y = \beta' x + \epsilon$ where

$$\epsilon \sim \mathrm{N}_{50}(0, \mathbf{\Sigma}),$$

  with $\mathbf{\Sigma} = D\tilde{\Sigma}D$ where D is diagonal with entries equally spaced from 3 to .50 and $[\tilde{\Sigma}]_{j,k} = \xi \mathbf{1}(j = k) + \mathbf{1}(j \neq k)$;
- Each column of β has 5 randomly selected entries set equal to -1 or 1, zeros elsewhere.

Measure performance of the various estimators using mean squared error:

$$\frac{\|\hat{\beta} - \beta\|_F^2}{pq}.$$

Penalized maximum likelihood estimators

When $\mathbf{\Omega}$ is unknown and the ϵ_i are normally distributed, the (doubly) penalized maximum likelihood estimator is:

$$\underset{\beta \in \mathbb{R}^{p \times q}, \Omega \in \mathbb{S}_+^q}{\arg\min} \left\{ \mathcal{F}(\beta, \Omega) + \lambda_\beta \mathcal{P}_\beta(\beta) + \lambda_\Omega \mathcal{P}_\Omega(\Omega) \right\},$$

where

$$\mathcal{F}(\beta, \Omega) = \operatorname{tr}\left[n^{-1} (Y - X\beta) \Omega (Y - X\beta)' \right] - \log\det(\Omega).$$

- The doubly penalized maximum likelihood estimator uses ℓ_1-penalties on both β and Ω, but can take hours to compute.
- `Oracle.MaxLik` uses an ℓ_1-penalty on β and fixes $\Omega = \mathbf{\Omega}$.
- `Pen.MaxLik` uses two step approximation to the doubly penalized maximum likelihood estimator.

Simulations: Validated tuning

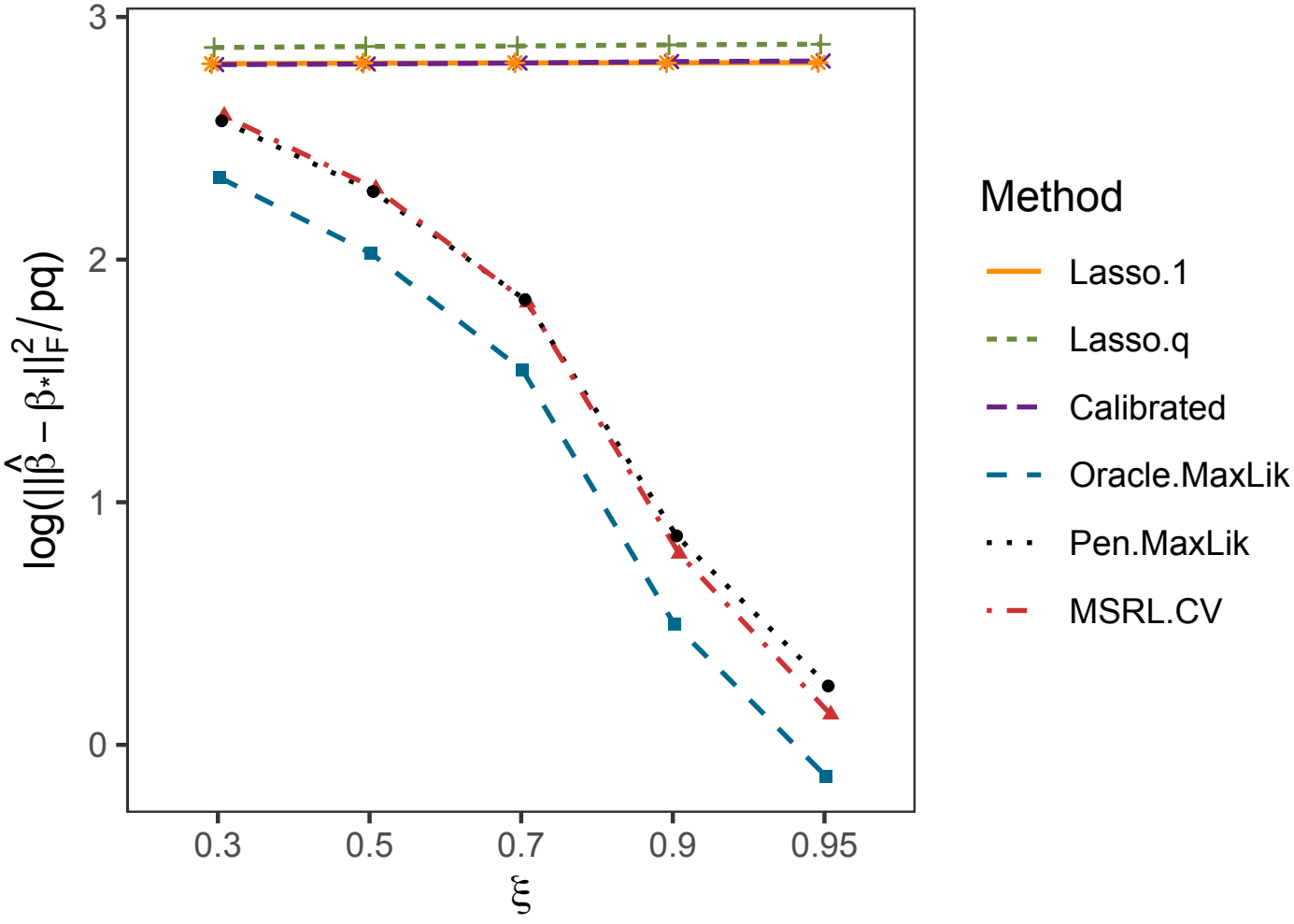


Figure: Average log-model error over 100 independent replications with $n = 200$, $p = 500$, $q = 50$, and $[\Sigma_*]_{j,k} = \xi \mathbf{1}(j \neq k) + \mathbf{1}(j = k)$.

Simulations: Validated tuning computing times

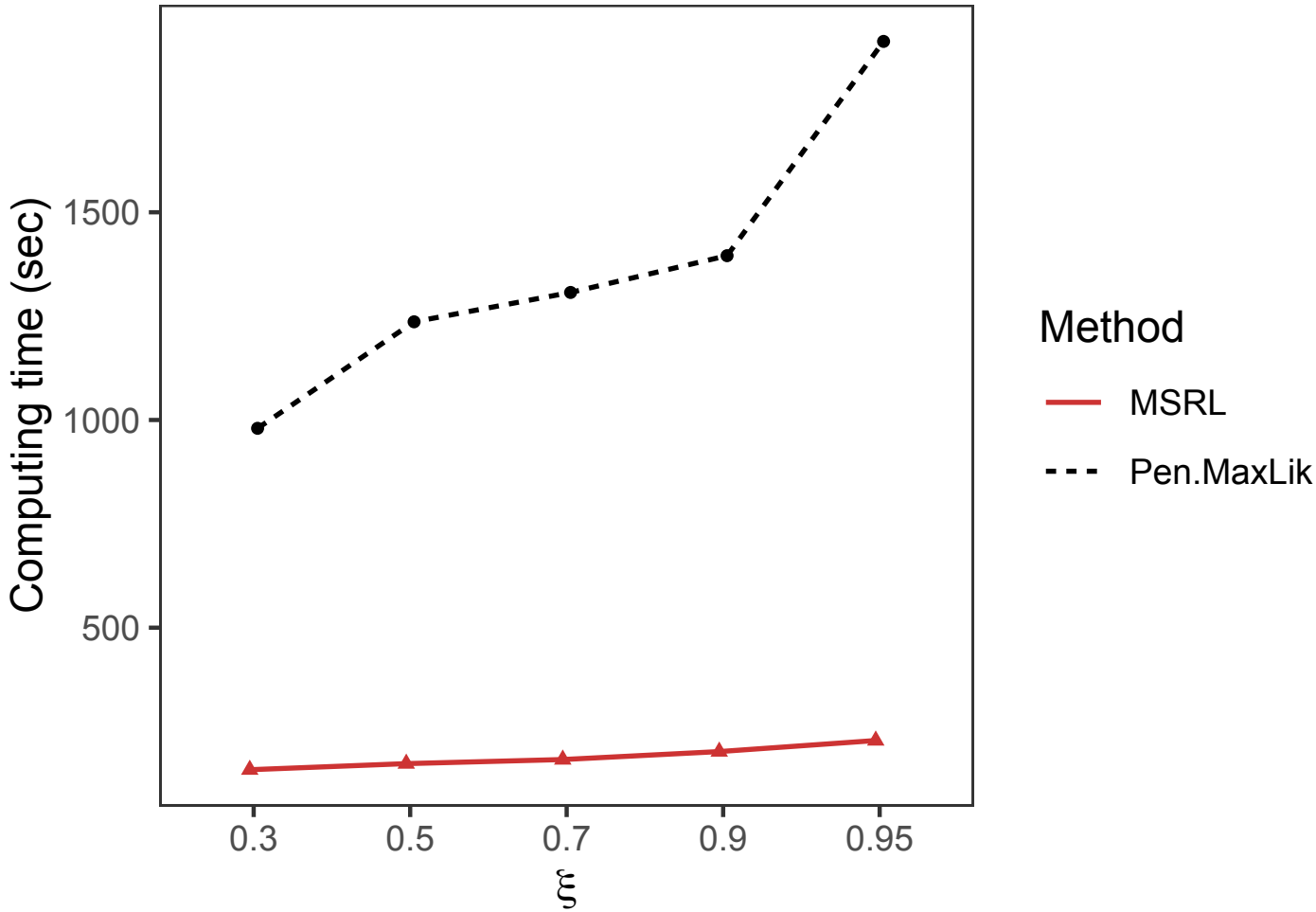


Figure: Average solution path computing times over 100 independent replications with $n = 200$, $p = 500$, $q = 50$, and $[\Sigma_*]_{j,k} = \xi \mathbf{1}(j \neq k) + \mathbf{1}(j = k)$.

Theoretical tuning

Recall that our error bounds held if we selected, for some $c > 1$,

$$(i)\ \ \lambda \geq \frac{c}{\sqrt{n}} \|X' U_* V_*'\|_{\max}$$

for where $U_* D_* V_*' = \mathcal{E}$ is the SVD of the error matrix. Further, using the distribution of $X' U_* V_*'$, we showed that if

$$(ii)\ \ \lambda = c \left(\frac{2 \log(4pq/\alpha)}{n} \right)^{1/2}$$

then (i) holds with probability at least $1 - \alpha$.

Theoretical tuning

Based on these results, in each replication we also tried both

$$\lambda = \frac{1.01}{\sqrt{n}} Q_{95}\left(\|X'O\|_{\max}\right), \quad (\texttt{MSRL-q95})$$

where we approximate the 95th percentile of $\|X'O\|_{\max}$ through simulation, and

$$\lambda = 1.01\sqrt{\frac{2\log(4pq/.05)}{n}}, \quad (\texttt{MSRL-Asymp}).$$

Following Belloni et al. (2011), we use refitting (i.e., SUR MLE based only on selected predictors) to mitigate extra bias.

Simulations: Theoretical tuning with all methods refit

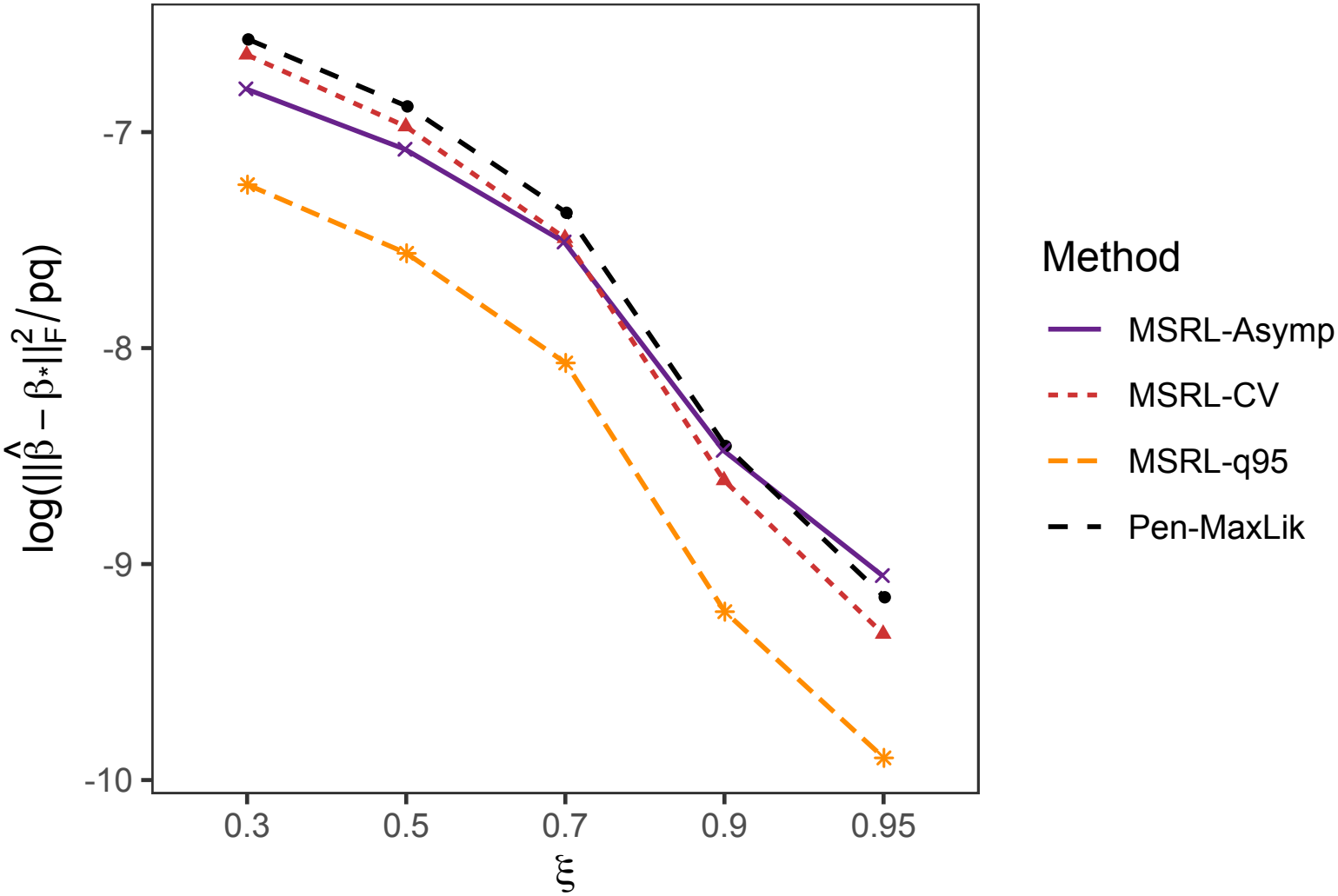


Figure: Average log-model error over 100 independent replications with $n = 200$, $p = 500$, $q = 50$, and $[\tilde{\Sigma}_*]_{j,k} = \xi \mathbf{1}(j \neq k) + \mathbf{1}(j = k)$.

Simulations: variable selection accuracy

	(True positive / false positive)				
Method	$\xi = 0.3$	$\xi = 0.5$	$\xi = 0.7$	$\xi = 0.9$	$\xi = 0.95$
`Pen.MaxLik`	85.52 / 5.18	87.76 / 5.27	90.44 / 5.35	94.59 / 5.64	96.12 / 5.89
`Oracle.MaxLik`	85.60 / 5.96	87.76 / 6.09	90.40 / 6.23	94.42 / 6.34	95.89 / 5.37
`MSRL.CV`	84.94 / 4.53	87.08 / 4.64	90.01 / 4.76	94.16 / 5.02	95.82 / 5.23
`MSRL.Asymp`	47.78 / 0.01	52.60 / 0.01	59.35 / 0.01	71.81 / 0.01	77.80 / 0.01
`MSRL.Asymp-2`	79.61 / 0.88	82.64 / 0.89	86.42 / 0.90	91.80 / 0.91	94.18 / 0.92
`MSRL.q95`	56.03 / 0.02	60.76 / 0.02	67.45 / 0.03	79.13 / 0.03	84.33 / 0.03
`MSRL.q95-2`	81.72 / 1.60	84.48 / 1.61	87.71 / 1.62	92.70 / 1.63	94.82 / 1.64

Table: Average true positive and false positive rates over 100 independent replications for identifying nonzero entries of β.

Direct tuning

Direct tuning without refitting often led to slightly larger squared norm error than did validation-based tuning.

- Rescaling both direct tuning parameters by $1/2$ led to nearly the same accuracy as validation-based tuning parameter – a result also observed by Belloni et al. (2011).
- Direct tuning does not require cross-validation: it requires computing MSRL for a single value of the tuning parameter.
  - Computing times were less than 4 seconds in all replications (recalling that $pq = 25000$).

GBM data example

- We used our method to model the linear relationship between microRNA expression and gene expression in patients with glimoblastoma multiforme, an aggressive brain cancer, collected by the Cancer Genome Atlas Project (TCGA).

GBM data example

- We used our method to model the linear relationship between microRNA expression and gene expression in patients with glimoblastoma multiforme, an aggressive brain cancer, collected by the Cancer Genome Atlas Project (TCGA).

- Since microRNAs can act as either oncogenes (a gene which may cause cancer) or tumor suppressors, it is of scientific interest to quantify how gene expression regulates microRNA expression.

GBM data example

- We used our method to model the linear relationship between microRNA expression and gene expression in patients with glimoblastoma multiforme, an aggressive brain cancer, collected by the Cancer Genome Atlas Project (TCGA).

- Since microRNAs can act as either oncogenes (a gene which may cause cancer) or tumor suppressors, it is of scientific interest to quantify how gene expression regulates microRNA expression.

- Following Wang (2015), we predict expression in microRNAs with the m largest median absolute deviations with the expression of g genes with largest median absolute deviations.

GBM data example

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

	Weighted prediction error				Nuclear norm prediction error			
m	20		40		20		40	
g	500	1000	500	1000	500	1000	500	1000
`MSRL.cv`	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
`Pen.MaxLik`*	0.6416	0.6060	0.6659	0.6354	0.2130	0.2069	0.3387	0.3314

GBM data example

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

	Weighted prediction error				Nuclear norm prediction error			
m	20		40		20		40	
g	500	1000	500	1000	500	1000	500	1000
`MSRL.cv`	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
`Pen.MaxLik`*	0.6416	0.6060	0.6659	0.6354	0.2130	0.2069	0.3387	0.3314

GBM data example

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

	Weighted prediction error				Nuclear norm prediction error			
m	20		40		20		40	
g	500	1000	500	1000	500	1000	500	1000
`MSRL.cv`	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
`Pen.MaxLik`*	0.6416	0.6060	0.6659	0.6354	0.2130	0.2069	0.3387	0.3314

Conclusion

- In high-dimensional multivariate response linear regression, error dependence should not be ignored.
- The multivariate square-root lasso is a convex alternative to doubly penalized normal maximum likelihood estimators.
  - MSRL performed as well or better than the penalized MLE in the settings we considered.
  - MSRL is significantly faster to compute since we avoid estimating the error precision matrix.
  - Directly tuned version can be computed almost instanteously for even large p, although bias may be an issue.

Thank you!

web: ajmolstad.github.io
email: amolstad@ufl.edu