Sparse multivariate regression with covariance estimation

Adam J. Rothman, Elizaveta Levina, and Ji Zhu
Department of Statistics
University of Michigan

May 18, 2010

Abstract

We propose a procedure for constructing a sparse estimator of a multivariate regression coefficient matrix that accounts for correlation of the response variables. This method, which we call multivariate regression with covariance estimation (MRCE), involves penalized likelihood with simultaneous estimation of the regression coefficients and the covariance structure. An efficient optimization algorithm and a fast approximation are developed for computing MRCE. Using simulation studies, we show that the proposed method outperforms relevant competitors when the responses are highly correlated. We also apply the new method to a finance example on predicting asset returns. An R-package containing this dataset and code for computing MRCE and its approximation are available online.

Keywords: High dimension low sample size; Sparsity; Multiple output regression; Lasso

1 Introduction

Multivariate regression is a generalization of the classical regression model of regressing a single response on p predictors to regressing $q > 1$ responses on p predictors. Applica-

tions of this general model arise in chemometrics, econometrics, psychometrics, and other quantitative disciplines where one predicts multiple responses with a single set of prediction variables. For example, predicting several measures of quality of paper with a set of variables relating to its production, or predicting asset returns for several companies using the vector auto-regressive model (Reinsel, 1997), both result in multivariate regression problems.

Let $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^T$ denote the predictors, let $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{iq})^T$ denote the responses, and let $\boldsymbol{\epsilon}_i = (\epsilon_1, \ldots, \epsilon_q)^T$ denote the errors, all for the ith sample. The multivariate regression model is given by,

$$\boldsymbol{y}_i = \boldsymbol{B}^T \boldsymbol{x}_i + \boldsymbol{\epsilon}_i, \qquad \text{for} \quad i = 1, \ldots, n,$$

where $\boldsymbol{B}$ is a $p \times q$ regression coefficient matrix and n is the sample size. Column k of $\boldsymbol{B}$ is the regression coefficient vector from regressing the kth response on the predictors. We make the standard assumption that $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_n$ are i.i.d $N_q(0, \boldsymbol{\Sigma})$. Thus, given a realization of the predictor variables, the covariance matrix of the response variables is $\boldsymbol{\Sigma}$.

The model can be expressed in matrix notation. Let $\boldsymbol{X}$ denote the $n \times p$ predictor matrix where its ith row is $\boldsymbol{x}_i^T$, let $\boldsymbol{Y}$ denote the $n \times q$ random response matrix where its ith row is $\boldsymbol{y}_i^T$, and let $\boldsymbol{E}$ denote the $n \times q$ random error matrix where its ith row is $\boldsymbol{\epsilon}_i^T$, then the model is,

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} + \boldsymbol{E}.$$

Note that if $q = 1$, the model simplifies to the classical regression problem where $\boldsymbol{B}$ is a p dimensional regression coefficient vector. For simplicity of notation we assume that columns of $\boldsymbol{X}$ and $\boldsymbol{Y}$ have been centered and thus the intercept terms are omitted.

The negative log-likelihood function of $(\boldsymbol{B}, \boldsymbol{\Omega})$, where $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$, can be expressed up to a constant as,

$$g(\boldsymbol{B}, \boldsymbol{\Omega}) = \operatorname{tr}\left[\frac{1}{n}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B})^T(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B})\boldsymbol{\Omega}\right] - \log|\boldsymbol{\Omega}|. \tag{1}$$

The maximum likelihood estimator for $\boldsymbol{B}$ is simply $\hat{\boldsymbol{B}}^{\text{OLS}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$, which amounts to performing separate ordinary least squares estimates for each of the q response variables and does not depend on $\boldsymbol{\Omega}$.

Prediction with the multivariate regression model requires one to estimate pq parameters which becomes challenging when there are many predictors and responses. Criterion-based model selection has been extended to multivariate regression by Bedrick and Tsai (1994) and Fujikoshi and Satoh (1997). For a review of Bayesian approaches for model selection and prediction with the multivariate regression model see Brown et al. (2002) and references therein. A dimensionality reduction approach called reduced-rank regression (Anderson, 1951; Izenman, 1975; Reinsel and Velu, 1998) minimizes (1) subject to $\text{rank}(\boldsymbol{B}) = r$ for some $r \leq \min(p, q)$. The solution involves canonical correlation analysis, and combines information from all of the q response variables into r canonical response variates that have the highest canonical correlation with the corresponding predictor canonical variates. As in the case of principal components regression, the interpretation of the reduced rank model is typically impossible in terms of the original predictors and responses.

Other approaches aimed at reducing the number of parameters in the coefficient matrix $\boldsymbol{B}$ involve solving,

$$\hat{\boldsymbol{B}} = \underset{\boldsymbol{B}}{\text{argmin}}\, \text{tr}\left[(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B})^T(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B})\right] \quad \text{subject to:} \quad C(\boldsymbol{B}) \leq t, \tag{2}$$

where $C(\boldsymbol{B})$ is some constraint function. A method called factor estimation and selection (FES) was proposed in Yuan et al. (2007), who apply the constraint function $C(\boldsymbol{B}) = \sum_{j=1}^{\min(p,q)} \sigma_j(\boldsymbol{B})$, where $\sigma_j(\boldsymbol{B})$ is the jth singular value of $\boldsymbol{B}$. This constraint encourages sparsity in the singular values of $\hat{\boldsymbol{B}}$, and hence reduces the rank of $\hat{\boldsymbol{B}}$; however, unlike reduced rank regression, FES offers a continuous regularization path. A novel approach for imposing sparsity in the entries of $\hat{\boldsymbol{B}}$ was taken by Turlach et al. (2005), who proposed

the constraint function, $C(\boldsymbol{B}) = \sum_{j=1}^{p} \max(|b_{j1}|, \ldots, |b_{jq}|)$. This method was recommended for model selection (sparsity identification), and not for prediction because of the bias of the L_∞-norm penalty. Imposing sparsity in $\hat{\boldsymbol{B}}$ for the purposes of identifying "master predictors" was proposed by Peng et al. (2009), who applied a combined constraint function $C(\boldsymbol{B}) = \lambda C_1(\boldsymbol{B}) + (1-\lambda) C_2(\boldsymbol{B})$ for $\lambda \in [0, 1]$, where $C_1(\boldsymbol{B}) = \sum_{j,k} |b_{jk}|$, the lasso constraint (Tibshirani, 1996) on the entries of $\boldsymbol{B}$ and $C_2(\boldsymbol{B}) = \sum_{j=1}^{p} (b_{j1}^2 + \cdots + b_{jq}^2)^{0.5}$, the sum of the L_2-norms of the rows of $\boldsymbol{B}$ (Yuan and Lin, 2006). The first constraint introduces sparsity in the entries of $\hat{\boldsymbol{B}}$ and the second constraint introduces zeros for all entires in some rows of $\hat{\boldsymbol{B}}$, meaning that some predictors are irrelevant for all q responses. Asymptotic properties for an estimator using this constraint with $\lambda = 0$ have also been established (Obozinski et al., 2008). This combined constraint approach provides highly interpretable models in terms of the prediction variables. However, all of the methods above that solve (2) do not account for correlated errors.

To directly exploit the correlation in the response variables to improve prediction performance, a method called Curds and Whey (C&W) was proposed by Breiman and Friedman (1997). C&W predicts the multivariate response with an optimal linear combination of the ordinary least squares predictors. The C&W linear predictor has the form $\tilde{\boldsymbol{Y}} = \hat{\boldsymbol{Y}}^{\mathrm{OLS}} \boldsymbol{M}$, where $\boldsymbol{M}$ is a $q \times q$ shrinkage matrix estimated from the data. This method exploits correlation in the responses arising from shared random predictors as well as correlated errors.

In this paper, we propose a method that combines some of the strengths of the estimators discussed above to improve prediction in the multivariate regression problem while allowing for interpretable models in terms of the predictors. We reduce the number of parameters using the lasso penalty on the entries of $\boldsymbol{B}$ while accounting for correlated errors. We accomplish this by simultaneously optimizing (1) with penalties on the entries of $\boldsymbol{B}$ and $\boldsymbol{\Omega}$. We call our new method multivariate regression with covariance estimation (MRCE). The method assumes predictors are not random; however, the resulting formulas for the estimates

would be the same with random predictors. Our focus is on the conditional distribution of $\boldsymbol{Y}$ given $\boldsymbol{X}$ and thus, unlike in the Curds and Whey framework, the correlation of the response variables arises only from the correlation in the errors.

We also note that the use of lasso penalty on the entries of Ω has been considered by several authors in the context of covariance estimation (Yuan and Lin, 2007; d'Aspremont et al., 2008; Rothman et al., 2008; Friedman et al., 2008). However, here we use it in the context of a regression problem, thus making it an example of what one could call *supervised* covariance estimation: the covariance matrix here is estimated in order to improve prediction, rather than as a stand-alone parameter. This is a natural next step from the extensive covariance estimation literature, which has been given surprisingly little attention to date; one exception is the joint regression approach of Witten and Tibshirani (2009). Another less directly relevant example of such supervised estimation is the supervised principal components by Bair et al. (2006).

The remainder of the paper is organized as follows: Section 2 describes the MRCE method and associated computational algorithms, Section 3 presents simulation studies comparing MRCE to competing methods, Section 4 presents an application of MRCE for predicting asset returns, and Section 5 concludes with a summary and discussion.

2 Joint estimation of $\boldsymbol{B}$ and $\boldsymbol{\Omega}$ via penalized normal likelihood

2.1 The MRCE method

We propose a sparse estimator for $\boldsymbol{B}$ that accounts for correlated errors using penalized normal likelihood. We add two penalties to the negative log-likelihood function g to construct

a sparse estimator of $\boldsymbol{B}$ depending on $\boldsymbol{\Omega} = [\omega_{j'j}]$,

$$(\hat{\boldsymbol{B}}, \hat{\boldsymbol{\Omega}}) = \underset{\boldsymbol{B},\boldsymbol{\Omega}}{\operatorname{argmin}} \left\{ g(\boldsymbol{B}, \boldsymbol{\Omega}) + \lambda_1 \sum_{j' \neq j} |\omega_{j'j}| + \lambda_2 \sum_{j=1}^{p} \sum_{k=1}^{q} |b_{jk}| \right\}, \tag{3}$$

where $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ are tuning parameters.

We selected the lasso penalty on the off-diagonal entries of the inverse error covariance $\boldsymbol{\Omega}$ for two reasons. First, it ensures that an optimal solution for $\boldsymbol{\Omega}$ has finite objective function value when there are more responses than samples ($q > n$); second, the penalty has the effect of reducing the number of parameters in the inverse error covariance, which is useful when q is large (Rothman et al., 2008). Other penalties such as the ridge penalty could be used when it is unreasonable to assume that the inverse error covariance matrix is sparse. If q is large, estimating a dense $\boldsymbol{\Omega}$ means that the MRCE regression method has $O(q^2)$ additional parameters in $\boldsymbol{\Omega}$ to estimate compared with doing separate lasso regressions for each response variable. Thus estimating a sparse $\boldsymbol{\Omega}$ has considerably lower variability, and so we focus on the lasso penalty on $\boldsymbol{\Omega}$. We show in simulations that when the inverse error covariance matrix is not sparse, the lasso penalty on $\boldsymbol{\Omega}$ still considerably outperforms ignoring covariance estimation altogether (i.e., doing a separate lasso regression for each response).

The lasso penalty on $\boldsymbol{B}$ introduces sparsity in $\hat{\boldsymbol{B}}$, which reduces the number of parameters in the model and provides interpretation. In classical regression ($q = 1$), the lasso penalty can offer major improvement in prediction performance when there is a relatively small number of relevant predictors. This penalty also ensures that an optimal solution for $\boldsymbol{B}$ is a function of $\boldsymbol{\Omega}$. Without a penalty on $\boldsymbol{B}$ (i.e., $\lambda_2 = 0$), the optimal solution for $\boldsymbol{B}$ is always $\hat{\boldsymbol{B}}^{\text{OLS}}$.

To see the effect of including the error covariance when estimating an L_1-penalized $\boldsymbol{B}$, assume that we know $\boldsymbol{\Omega}$ and also assume $p < n$. Solving (3) for $\boldsymbol{B}$ with $\boldsymbol{\Omega}$ fixed is a convex problem (see Section 2.2) and thus there exists a global minimizer $\boldsymbol{B}^{\text{opt}}$. This implies that

there exists a zero subgradient of the objective function at $\boldsymbol{B}^{\text{opt}}$ (see Theorem 3.4.3 page 127 in Bazaraa et al. (2006)). We express this in matrix notation as,

$$\mathbf{0} = 2n^{-1}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{B}^{\text{opt}}\boldsymbol{\Omega} - 2n^{-1}\boldsymbol{X}^T\boldsymbol{Y}\boldsymbol{\Omega} + \lambda_2\boldsymbol{\Gamma},$$

which gives,

$$\boldsymbol{B}^{\text{opt}} = \hat{\boldsymbol{B}}^{\text{OLS}} - \lambda_2(2n^{-1}\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{\Gamma}\boldsymbol{\Omega}^{-1}, \tag{4}$$

where $\boldsymbol{\Gamma} \equiv \boldsymbol{\Gamma}(\boldsymbol{B}^{\text{opt}})$ is a $p \times q$ matrix with entries $\gamma_{ij} = \text{sign}(b_{ij}^{\text{opt}})$ if $b_{ij}^{\text{opt}} \neq 0$ and otherwise $\gamma_{ij} \in [-1, 1]$ with specific values chosen to solve (4). Ignoring the correlation in the error is equivalent to assuming that $\boldsymbol{\Omega}^{-1} = \boldsymbol{I}$. Thus having highly correlated errors will have greater influence on the amount of shrinkage of each entry of $\boldsymbol{B}^{\text{opt}}$ than having mildly correlated errors.

2.2 Computational algorithms

The optimization problem in (3) is not convex; however, solving for either $\boldsymbol{B}$ or $\boldsymbol{\Omega}$ with the other fixed is convex. We present an algorithm for solving (3) and a fast approximation to it.

Solving (3) for $\boldsymbol{\Omega}$ with $\boldsymbol{B}$ fixed at a chosen point $\boldsymbol{B}_0$ yields the optimization problem,

$$\hat{\boldsymbol{\Omega}}(\boldsymbol{B}_0) = \underset{\boldsymbol{\Omega}}{\text{argmin}} \left\{ \text{tr}\left(\hat{\boldsymbol{\Sigma}}_R\boldsymbol{\Omega}\right) - \log|\boldsymbol{\Omega}| + \lambda_1 \sum_{j' \neq j} |\omega_{j'j}| \right\}, \tag{5}$$

where $\hat{\boldsymbol{\Sigma}}_R = \frac{1}{n}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}_0)^T(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}_0)$. This is exactly the L_1-penalized covariance estimation problem considered by d'Aspremont et al. (2008), Yuan and Lin (2007), Rothman et al. (2008), Friedman et al. (2008), Lu (2009), and Lu (2010). We use the graphical lasso (glasso) algorithm of Friedman et al. (2008) to solve (5) since it is fast and the most commonly used algorithm for solving (5).

Solving (3) for $\boldsymbol{B}$ with $\boldsymbol{\Omega}$ fixed at a chosen point $\boldsymbol{\Omega}_0$ yields the optimization problem,

$$\hat{\boldsymbol{B}}(\boldsymbol{\Omega}_0) = \underset{\boldsymbol{B}}{\operatorname{argmin}} \left\{ \operatorname{tr}\left[\frac{1}{n}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B})^T(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B})\boldsymbol{\Omega}_0\right] + \lambda_2 \sum_{j=1}^{p}\sum_{k=1}^{q} |b_{jk}| \right\}, \tag{6}$$

which is convex if $\boldsymbol{\Omega}_0$ is non-negative definite. This follows because the trace term in the objective function has the Hessian $2n^{-1}\boldsymbol{\Omega}_0 \otimes \boldsymbol{X}^T\boldsymbol{X}$, which is non-negative definite because the Kronecker product of two symmetric non-negative definite matrices is also non-negative definite. A solution can be efficiently computed using cyclical-coordinate descent analogous to that used for solving the single output lasso problem (Friedman et al., 2007). We summarize the optimization procedure in Algorithm 1. We use the ridge penalized least-squares estimate $\hat{\boldsymbol{B}}^{\text{RIDGE}} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda_2\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$ to scale our test of parameter convergence since it is always well defined (including when $p > n$).

Algorithm 1. *Given $\boldsymbol{\Omega}$ and an initial value $\hat{\boldsymbol{B}}^{(0)}$, let $\boldsymbol{S} = \boldsymbol{X}^T\boldsymbol{X}$ and $\boldsymbol{H} = \boldsymbol{X}^T\boldsymbol{Y}\boldsymbol{\Omega}$.*

Step 1: *Set $\hat{\boldsymbol{B}}^{(m)} \leftarrow \hat{\boldsymbol{B}}^{(m-1)}$. Visit all entries of $\hat{\boldsymbol{B}}^{(m)}$ in some sequence and for entry (r, c) update $\hat{b}_{rc}^{(m)}$ with the minimizer of the objective function along its coordinate direction given by,*

$$\hat{b}_{rc}^{(m)} \leftarrow \operatorname{sign}\left(\hat{b}_{rc}^{(m)} + \frac{h_{rc} - u_{rc}}{s_{rr}\omega_{cc}}\right)\left(\left|\hat{b}_{rc}^{(m)} + \frac{h_{rc} - u_{rc}}{s_{rr}\omega_{cc}}\right| - \frac{n\lambda_2}{s_{rr}\omega_{cc}}\right)_+,$$

where $u_{rc} = \sum_{j=1}^{p}\sum_{k=1}^{q} \hat{b}_{jk}^{(m)} s_{rj}\omega_{kc}$.

Step 2: *If $\sum_{j,k}|\hat{b}_{jk}^{(m)} - \hat{b}_{jk}^{(m-1)}| < \epsilon \sum_{j,k}|\hat{b}_{jk}^{\text{RIDGE}}|$ then stop, otherwise goto Step 1.*

A full derivation of Algorithm 1 is found in the Appendix. Algorithm 1 is guaranteed to converge to the global minimizer if the given $\boldsymbol{\Omega}$ is non-negative definite. This follows from the fact that the trace term in the objective function is convex and differentiable and the penalty term decomposes into a sum of convex functions of individual parameters (Tseng, 1988; Friedman et al., 2007). We set the convergence tolerance parameter $\epsilon = 10^{-4}$.

In terms of computational cost, we need to cycle through pq parameters, and for each compute u_{rc}, which costs at most $O(pq)$ flops, and if the least sparse iterate has v non-zeros, then computing u_{rc} costs $O(v)$. The worst case cost for the entire algorithm is $O(p^2q^2)$.

Using (5) and (6) we can solve (3) using block-wise coordinate descent, that is, we iterate minimizing with respect to $\boldsymbol{B}$ and minimizing with respect to $\boldsymbol{\Omega}$.

Algorithm 2 (MRCE)**.** *For fixed values of* λ_1 *and* λ_2*, initialize* $\hat{\boldsymbol{B}}^{(0)} = \mathbf{0}$ *and* $\hat{\boldsymbol{\Omega}}^{(0)} = \hat{\boldsymbol{\Omega}}(\hat{\boldsymbol{B}}^{(0)})$.

Step 1: *Compute* $\hat{\boldsymbol{B}}^{(m+1)} = \hat{\boldsymbol{B}}(\hat{\boldsymbol{\Omega}}^{(m)})$ *by solving* (6) *using Algorithm 1.*

Step 2: *Compute* $\hat{\boldsymbol{\Omega}}^{(m+1)} = \hat{\boldsymbol{\Omega}}(\hat{\boldsymbol{B}}^{(m+1)})$ *by solving* (5) *using the glasso algorithm.*

Step 3: *If* $\sum_{j,k} |\hat{b}_{jk}^{(m+1)} - \hat{b}_{jk}^{(m)}| < \epsilon \sum_{j,k} |\hat{b}_{jk}^{\text{RIDGE}}|$ *then stop, otherwise goto Step 1.*

Algorithm 2 uses block-wise coordinate descent to compute a local solution for (3). Steps 1 and 2 both ensure a decrease in the objective function value. In practice we found that for certain values of the penalty tuning parameters (λ_1, λ_2), the algorithm may take many iterations to converge for high-dimensional data. For such cases, we propose a faster approximate solution to (3).

Algorithm 3 (Approximate MRCE)**.** *For fixed values of* λ_1 *and* λ_2*,*

Step 1: *Perform* q *separate lasso regressions each with the same optimal tuning parameter* $\hat{\lambda}_0$ *selected with a cross validation procedure. Let* $\hat{\boldsymbol{B}}_{\hat{\lambda}_0}^{\text{lasso}}$ *denote the solution.*

Step 2: *Compute* $\hat{\boldsymbol{\Omega}} = \hat{\boldsymbol{\Omega}}(\hat{\boldsymbol{B}}_{\hat{\lambda}_0}^{\text{lasso}})$ *by solving* (5) *using the glasso algorithm.*

Step 3: *Compute* $\hat{\boldsymbol{B}} = \hat{\boldsymbol{B}}(\hat{\boldsymbol{\Omega}})$ *by solving* (6) *using Algorithm 1.*

The approximation summarized in Algorithm 3 is only iterative inside its steps. The algorithm begins by finding the optimally tuned lasso solution $\hat{\boldsymbol{B}}_{\hat{\lambda}_0}^{\text{lasso}}$ (using cross validation to

select the tuning parameter $\hat{\lambda}_0$), then computes an estimate for $\boldsymbol{\Omega}$ using the glasso algorithm with $\hat{\boldsymbol{B}}^{\text{lasso}}_{\hat{\lambda}_0}$ plugged in, and then solves (6) using this inverse covariance estimate. Note that one still must select two tuning parameters (λ_1, λ_2). The performance of the approximation is studied in Section 3.

2.3 Tuning parameter selection

For the MRCE methods, the tuning parameters λ_1 and λ_2 could be selected using K-fold cross validation, where validation prediction error is accumulated over all q responses for each fold. Specifically, select the optimal tuning parameters $\hat{\lambda}_1$ and $\hat{\lambda}_2$ using,

$$(\hat{\lambda}_1, \hat{\lambda}_2) = \underset{\lambda_1,\lambda_2}{\operatorname{argmin}} \sum_{k=1}^{K} \|\boldsymbol{Y}^{(k)} - \boldsymbol{X}^{(k)} \boldsymbol{B}^{(-k)}_{\lambda_1,\lambda_2}\|^2_F$$

where $\boldsymbol{Y}^{(k)}$ is the matrix of responses with observations in the kth fold, $\boldsymbol{X}^{(k)}$ is the matrix of predictors of observations in the kth fold, and $\boldsymbol{B}^{(-k)}_{\lambda_1,\lambda_2}$ is the estimated regression coefficient matrix computed with observations outside the kth fold, with tuning parameters λ_1 and λ_2. We have found in simulations that λ_2, which controls the penalization on the regression coefficient matrix, has greater influence on prediction performance than λ_1, which controls the penalization of the inverse error covariance matrix.

3 Simulation study

3.1 Estimators

We compare the performance of the MRCE method, computed with the exact and the approximate algorithms, to other multivariate regression estimators that produce sparse estimates of $\boldsymbol{B}$. We report results for the following methods:

- *Lasso*: Performing q separate lasso regressions, each with the same tuning parameter λ.
- *Separate lasso*: Perform q separate lasso regressions, each with its own tuning parameter.
- *MRCE*: The solution to (3) (Algorithm 2).
- *Approx. MRCE*: An approximate solution to (3) (Algorithm 3).

The ordinary least squares estimator $\hat{\boldsymbol{B}}^{\text{OLS}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$ and the Curds and Whey method of Breiman and Friedman (1997) are computed as a benchmark for low-dimensional models (they are not directly applicable when $p > n$).

We select the tuning parameters by minimizing the squared prediction error, accumulated over all q responses, of independently generated validation data of the same sample size ($n = 50$). This is similar to performing the cross-validation approach described in Section 2.3, and is used to save computing time for the simulations. For the MRCE methods, the two tuning parameters are selected simultaneously.

3.2 Models

In each replication for each model, we generate an $n \times p$ predictor matrix $\boldsymbol{X}$ with rows drawn independently from $N_p(0, \boldsymbol{\Sigma}_X)$ where $\boldsymbol{\Sigma}_X = [\sigma_{Xij}]$ is given by $\sigma_{Xij} = 0.7^{|i-j|}$. This model for the predictors was also used by Yuan et al. (2007) and Peng et al. (2009). Note that all of the predictors are generated with the same unit marginal variance. The error matrix $\boldsymbol{E}$ is generated independently with rows drawn independently from $N_q(0, \boldsymbol{\Sigma}_E)$. We consider two models for the error covariance,

- AR(1) error covariance: $\sigma_{Eij} = \rho_E^{|i-j|}$, with values of ρ_E ranging from 0 to 0.9.

- Fractional Gaussian Noise (FGN) error covariance:

$$\sigma_{Eij} = 0.5\left((|i-j|+1)^{2H} - 2|i-j|^{2H} + (|i-j|-1)^{2H}\right),$$

  with values of the Hurst parameter $H = 0.9, 0.95$.

The inverse error covariance for the AR(1) model is a tri-diagonal sparse matrix while its covariance matrix is dense, and thus this error covariance model completely satisfies the regularizing assumptions for the MRCE method, which exploits the correlated error and the sparse inverse error covariance. The FGN model is a standard example of long-range dependence and both the error covariance and its inverse are dense matrices. Varying H gives different degree of dependence, from $H = 0.5$ corresponding to an i.i.d. sequence to $H = 1$ corresponding to a perfectly correlated one. Thus the introduction of sparsity in the inverse error covariance by the MRCE method should not help; however, since the errors are highly correlated the MRCE method may still perform better than the lasso penalized regressions for each response, which ignore correlation among the errors. The sample size is fixed at $n = 50$ for all models.

We generate sparse coefficient matrices $\boldsymbol{B}$ in each replication using the matrix element-wise product,

$$\boldsymbol{B} = \boldsymbol{W} * \boldsymbol{K} * \boldsymbol{Q},$$

where $\boldsymbol{W}$ is generated with independent draws for each entry from $N(0,1)$, $\boldsymbol{K}$ has entries with independent Bernoulli draws with success probability s_1, and $\boldsymbol{Q}$ has rows that are either all one or all zero, where p independent Bernoulli draws with success probability s_2 are made to determine whether each row is the ones vector or the zeros vector. Generating $\boldsymbol{B}$ in this manner, we expect $(1 - s_2)p$ predictors to be irrelevant for all q responses, and we expect each relevant predictor to be relevant for $s_1 q$ of the response variables.

3.3 Performance evaluation

We measure performance using model error, following the approach in Yuan et al. (2007), which is defined as,

$$\mathrm{ME}(\hat{\boldsymbol{B}}, \boldsymbol{B}) = \mathrm{tr}\left[(\hat{\boldsymbol{B}} - \boldsymbol{B})^T \boldsymbol{\Sigma}_X (\hat{\boldsymbol{B}} - \boldsymbol{B})\right].$$

We also measure the sparsity recognition performance using true positive rate (TPR) and true negative rate (TNR),

$$\mathrm{TPR}(\hat{\boldsymbol{B}}, \boldsymbol{B}) = \frac{\#\{(i,j) : \hat{b}_{ij} \neq 0 \quad \text{and} \quad b_{ij} \neq 0\}}{\#\{(i,j) : b_{ij} \neq 0\}}, \tag{7}$$

$$\mathrm{TNR}(\hat{\boldsymbol{B}}, \boldsymbol{B}) = \frac{\#\{(i,j) : \hat{b}_{ij} = 0 \quad \text{and} \quad b_{ij} = 0\}}{\#\{(i,j) : b_{ij} = 0\}}. \tag{8}$$

Both the true positive rate and true negative rates must be considered simultaneously since OLS always has perfect TPR and $\hat{\boldsymbol{B}} = \boldsymbol{0}$ always has perfect TNR.

3.4 Results

The model error performance for AR(1) error covariance model is displayed in Figure 1 for low-dimensional models, and Figure 2 and Table 1 for high-dimensional models. Standard errors are omitted in the figures because of visibility issues, and we note that they are less than 4% of the corresponding average model error. We see that the margin by which MRCE and its approximation outperform the lasso and separate lasso in terms of model error increases as the error correlation ρ_E increases. This trend is consistent with the analysis of the subgradient equation given in (4), since the manner by which MRCE performs lasso shrinkage exploits highly correlated errors. Additionally, the MRCE method and its approximation outperform the lasso and separate lasso more for sparser coefficient matrices. We omitted the exact MRCE method for $p = 60, q = 20$ and $p = q = 100$ because these cases were computationally intractable. For a single realization of the model with $p = 20, q = 60$

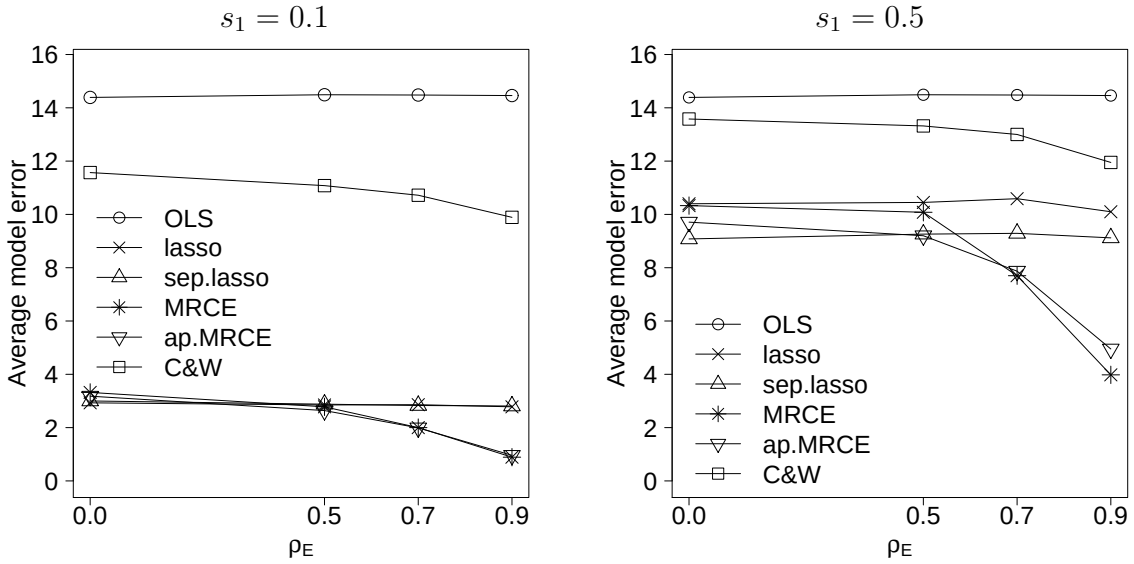

Figure 1: Average model error versus AR(1) correlation ρ_E, based on 50 replications with $n = 50$, $p = q = 20$, and $s_2 = 1$.

and $\rho_E = 0.9$, using the tuning parameters selected with cross validation, MRCE took 4.1 seconds, approximate MRCE took 1.7 seconds, lasso took 0.5 seconds and separate lasso took 0.4 seconds to compute on a workstation with a 2 GHz processor with 4GB of RAM. All of the sparse estimators outperform the ordinary least squares method by a considerable margin. The Curds and Whey method, although designed to exploit correlation in the responses, is outperformed here because it does not introduce sparsity in $\boldsymbol{B}$.

Table 1: Model error for the AR(1) error covariance models of high dimension, with $p = q = 100$, $s_1 = 0.5$, and $s_2 = 0.1$. Averages and standard errors in parenthesis are based on 50 replications with $n = 50$.

ρ_E	lasso	sep.lasso	ap.MRCE
0.9	58.79	59.32	34.87
	(2.29)	(2.35)	(1.54)
0.7	59.09	59.60	60.12
	(2.22)	(2.30)	(2.02)

The model error performance for FGN error covariance model is reported in Table 2

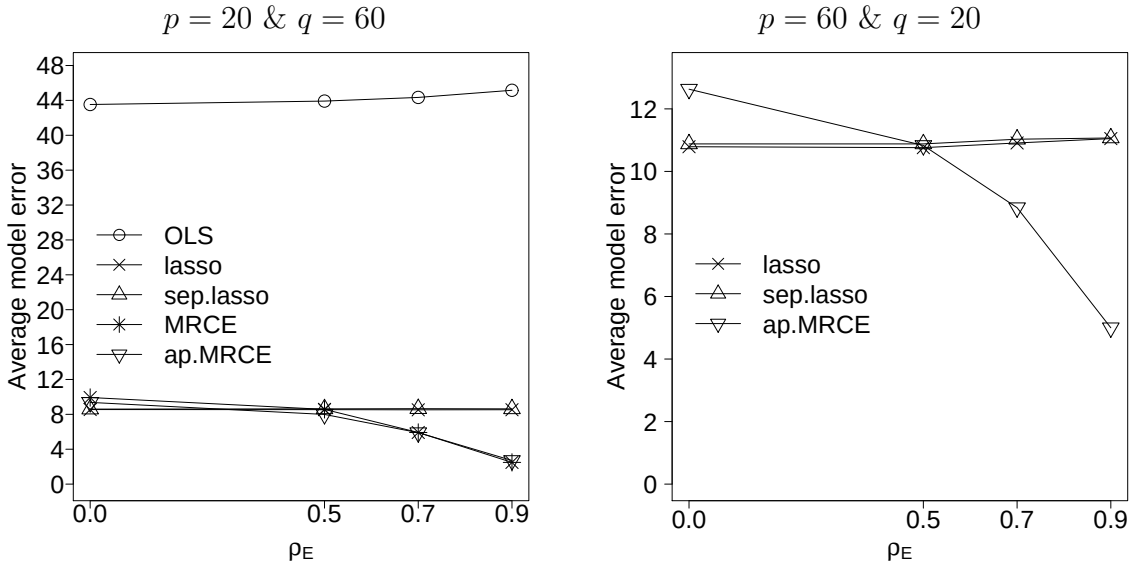

Figure 2: Average model error versus AR(1) correlation ρ_E, based on 50 replications with $n = 50$ and $s_1 = 0.1, s_2 = 1$.

for low-dimensional models and in Table 3 for high-dimensional models. Although there is no sparsity in the inverse error covariance for the MRCE method and its approximation to exploit, we see that both methods are still able to provide considerable improvement over the lasso and separate lasso methods by exploiting the highly correlated error. As seen with the AR(1) error covariance model, as the amount of correlation increases (i.e., larger values of H), the margin by which the MRCE method and its approximation outperform competitors increases.

We report the true positive rate and true negative rates in Table 4 for the AR(1) error covariance models and in Table 5 for the FGN error covariance models. We see that as the error correlation increases (larger values of ρ_E and H), the true positive rate for the MRCE method and its approximation increases, while the true negative rate tends to decrease. While all methods perform comparably on these sparsity measures, the substantially lower prediction errors obtained by the MRCE methods give them a clear advantage over other methods.

Table 2: Model error for the FGN error covariance models of low dimension. Averages and standard errors in parenthesis are based on 50 replications with $n = 50$. Tuning parameters were selected using a 10^x resolution.

p	q	H	s_1, s_2	OLS	lasso	sep.lasso	MRCE	ap.MRCE	C&W
20	20	0.95	0.1, 1	14.51	2.72	2.71	1.03	1.01	9.86
				(0.69)	(0.10)	(0.11)	(0.02)	(0.03)	(0.46)
20	20	0.90	0.1, 1	14.49	2.76	2.77	1.78	1.71	10.29
				(0.53)	(0.09)	(0.09)	(0.05)	(0.05)	(0.36)
20	20	0.95	0.5, 1	14.51	9.89	8.94	3.63	4.42	11.72
				(0.69)	(0.26)	(0.21)	(0.09)	(0.16)	(0.45)
20	20	0.90	0.5, 1	14.49	10.01	9.03	6.11	6.34	12.29
				(0.53)	(0.21)	(0.18)	(0.14)	(0.13)	(0.34)

Table 3: Model error for the FGN error covariance models of high dimension. Averages and standard errors in parenthesis are based on 50 replications with $n = 50$. Tuning parameters were selected using a 10^x resolution.

p	q	H	s_1, s_2	OLS	lasso	sep.lasso	MRCE	ap.MRCE
20	60	0.95	0.1, 1	46.23	8.56	8.63	3.31	3.20
				(2.04)	(0.36)	(0.37)	(0.19)	(0.18)
20	60	0.90	0.1, 1	45.41	8.60	8.69	5.31	5.03
				(1.42)	(0.24)	(0.25)	(0.15)	(0.14)
60	20	0.95	0.1, 1	NA	11.15	11.23	-	4.84
					(0.35)	(0.36)		(0.12)
60	20	0.90	0.1, 1	NA	11.14	11.21	-	7.44
					(0.30)	(0.30)		(0.16)
100	100	0.95	0.5, 0.1	NA	58.28	58.86	-	31.85
					(2.36)	(2.44)		(1.26)
100	100	0.95	0.5, 0.1	NA	58.10	58.63	-	47.37
					(2.27)	(2.36)		(1.68)

4 Example: Predicting Asset Returns

We consider a dataset of weekly log-returns of 9 stocks from 2004, analyzed in Yuan et al. (2007). We selected this dataset because it is the most recent dataset analyzed in the multivariate regression literature. The data are modeled with a first-order vector autoregressive

Table 4: True Positive Rate / True Negative Rate for the AR(1) error covariance models, averaged over 50 replications; $n = 50$. Standard errors are omitted (the largest standard error is 0.04 and most are less than 0.01). Tuning parameters were selected using a 10^x resolution.

p	q	ρ_E	s_1, s_2	lasso	sep.lasso	MRCE	ap.MRCE
20	20	0.9	0.1, 1	0.83/0.72	0.82/0.74	0.95/0.59	0.94/0.62
20	20	0.7	0.1, 1	0.83/0.71	0.82/0.73	0.89/0.60	0.89/0.63
20	20	0.5	0.1, 1	0.83/0.70	0.81/0.73	0.86/0.62	0.87/0.63
20	20	0	0.1, 1	0.84/0.70	0.82/0.72	0.85/0.63	0.85/0.64
20	20	0.9	0.5, 1	0.86/0.44	0.87/0.44	0.93/0.42	0.91/0.45
20	20	0.7	0.5, 1	0.85/0.47	0.87/0.42	0.86/0.51	0.86/0.52
20	20	0.5	0.5, 1	0.83/0.52	0.87/0.44	0.83/0.54	0.85/0.48
20	20	0	0.5, 1	0.84/0.50	0.87/0.43	0.84/0.51	0.82/0.56
20	60	0.9	0.1, 1	0.83/0.70	0.80/0.74	0.94/0.58	0.93/0.61
20	60	0.7	0.1, 1	0.84/0.71	0.81/0.73	0.89/0.61	0.89/0.62
20	60	0.5	0.1, 1	0.84/0.70	0.82/0.73	0.86/0.64	0.86/0.64
20	60	0	0.1, 1	0.83/0.71	0.81/0.74	0.85/0.63	0.85/0.65
60	20	0.9	0.1, 1	0.79/0.76	0.79/0.76	-	0.89/0.66
60	20	0.7	0.1, 1	0.79/0.76	0.78/0.76	-	0.85/0.65
60	20	0.5	0.1, 1	0.79/0.76	0.79/0.76	-	0.83/0.66
60	20	0	0.1, 1	0.79/0.76	0.79/0.76	-	0.81/0.66
100	100	0.9	0.5, 0.1	0.77/0.81	0.76/0.82	-	0.87/0.72
100	100	0.7	0.5, 0.1	0.78/0.81	0.76/0.82	-	0.82/0.72

model,

$$\boldsymbol{Y} = \tilde{\boldsymbol{Y}}\boldsymbol{B} + \boldsymbol{E},$$

where the response $\boldsymbol{Y} \in \mathbb{R}^{T-1 \times q}$ has rows $\boldsymbol{y}_2, \ldots, \boldsymbol{y}_T$ and the predictor $\tilde{\boldsymbol{Y}} \in \mathbb{R}^{T-1 \times q}$ has rows $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{T-1}$. Here $\boldsymbol{y}_t$ corresponds to the vector of log-returns for the 9 companies at week t. Let $\boldsymbol{B} \in \mathbb{R}^{q \times q}$ denote the transition matrix. Following the approach of Yuan et al. (2007), we use log-returns from the first 26 weeks of the year ($T = 26$) as the training set, and the log-returns from the remaining 26 weeks of the year as the test set. Prediction performance is measured by the average mean-squared prediction error over the test set for each stock, with the model fitted using the training set. Tuning parameters were selected with 10-fold

Table 5: True Positive Rate / True Negative Rate for the FGN error covariance models averaged over 50 replications; $n = 50$. Standard errors are omitted (the largest standard error is 0.04 and most are less than 0.01). Tuning parameters were selected using a 10^x resolution.

p	q	H	s_1, s_2	lasso	sep.lasso	MRCE	ap.MRCE
20	20	0.95	0.1, 1	0.83/0.72	0.81/0.75	0.94/0.55	0.93/0.59
20	20	0.90	0.1, 1	0.84/0.71	0.83/0.73	0.90/0.59	0.89/0.61
20	20	0.95	0.5, 1	0.87/0.40	0.87/0.45	0.93/0.39	0.92/0.39
20	20	0.90	0.5, 1	0.86/0.43	0.87/0.45	0.88/0.51	0.90/0.43
20	60	0.95	0.1, 1	0.83/0.70	0.81/0.73	0.93/0.55	0.93/0.58
20	60	0.90	0.1, 1	0.83/0.70	0.81/0.73	0.90/0.58	0.90/0.60
60	20	0.95	0.1, 1	0.79/0.76	0.79/0.76	-	0.89/0.66
60	20	0.90	0.1, 1	0.79/0.76	0.78/0.76	-	0.87/0.65
100	100	0.95	0.5, 0.1	0.77/0.81	0.75/0.82	-	0.87/0.72
100	100	0.90	0.5, 0.1	0.77/0.81	0.75/0.82	-	0.83/0.71

CV.

Average test squared error over the 26 test points is reported in Table 6, where we see that the MRCE method and its approximation have somewhat better performance than the lasso and separate lasso methods. The lasso estimate of the transition matrix $\boldsymbol{B}$ was all zeros, yielding the null model. Nonetheless, this results in prediction performance comparable, (i.e., within a standard error), to the FES method of Yuan et al. (2007) (copied directly from Table 3 on page 341), which was shown to be the best of several competitors for these data. This comparable performance of the null model suggests that the signal is very weak in this dataset. Separate lasso, MRCE, and its approximation estimated 3/81, 4/81, and 12/81 coefficients as non-zero, respectively.

We report the estimate of the unit lag coefficient matrix $\boldsymbol{B}$ for the approximate MRCE method in Table 7, which is the least sparse estimate, identifying 12 non-zero entries. The estimated unit lag coefficient matrix for separate lasso, MRCE, and approximate MRCE all identified the log-return for Walmart at week $t-1$ as a relevant predictor for the log-return

Table 6: Average testing squared error for each output (company) $\times 1000$, based on 26 testing points. Standard errors are reported in parenthesis. The results for the FES method where copied from Table 3 in Yuan et al. (2007).

	OLS	sep.lasso	lasso	MRCE	ap.MRCE	FES
Walmart	0.98(0.27)	0.44(0.10)	0.42(0.12)	0.41(0.11)	0.41(0.11)	0.40
Exxon	0.39(0.08)	0.31(0.07)	0.31(0.07)	0.31(0.07)	0.31(0.07)	0.29
GM	1.68(0.42)	0.71(0.17)	0.71(0.17)	0.71(0.17)	0.69(0.17)	0.62
Ford	2.15(0.61)	0.77(0.25)	0.77(0.25)	0.77(0.25)	0.77(0.25)	0.69
GE	0.58(0.15)	0.45(0.09)	0.45(0.09)	0.45(0.09)	0.45(0.09)	0.41
ConocoPhillips	0.98(0.24)	0.79(0.22)	0.79(0.22)	0.79(0.22)	0.78(0.22)	0.79
Citigroup	0.65(0.17)	0.61(0.13)	0.66(0.14)	0.62(0.13)	0.62(0.13)	0.59
IBM	0.62(0.14)	0.49(0.10)	0.49(0.10)	0.49(0.10)	0.47(0.09)	0.51
AIG	1.93(0.93)	1.88(1.02)	1.88(1.02)	1.88(1.02)	1.88(1.02)	1.74
AVE	1.11(0.14)	0.72(0.12)	0.72(0.12)	0.71(0.12)	0.71(0.12)	0.67

of GE at week t, and the log-return for Ford at week $t-1$ as a relevant predictor for the log return of Walmart at week t. The FES does not provide any interpretation.

We also report the estimate for the inverse error covariance matrix for the MRCE method in Table 8. A non-zero entry (i, j) means that we estimate that ϵ_i is correlated with ϵ_j given the other errors (or ϵ_i is partially correlated with ϵ_j). We see that AIG (an insurance company) is estimated to be partially correlated with most of the other companies, and companies with similar products are partially correlated, such as Ford and GM (automotive), GE and IBM (technology), as well as Conoco Phillips and Exxon (oil). These results make sense in the context of financial data.

5 Summary and discussion

We proposed the MRCE method to produce a sparse estimate of the multivariate regression coefficient matrix $\boldsymbol{B}$. Our method explicitly accounts for the correlation of the response variables. We also developed a fast approximate algorithm for computing MRCE which

Table 7: Estimated coefficient matrix $\boldsymbol{B}$ for approximate MRCE

	Wal	Exx	GM	Ford	GE	CPhil	Citi	IBM	AIG
Walmart	0	0	0	0	0	0	0.123	0.078	0
Exxon	0	0	0	0	0	0	0	0	0
GM	0	0	0	0	0	0	0	0	0
Ford	-0.093	0.035	0.012	0	0	0	0	-0.040	-0.010
GE	0	0	0	0	0	0.044	0	0	0
ConocoPhillips	0	0.007	0	0	0	0	0	-0.005	0
Citigroup	0	0	0.025	0	0	0	0	0	0
IBM	0	0	0	0	0	0	0	0	0
AIG	0	0	0.031	0	0	0	0	0	0

Table 8: Inverse error covariance estimate for MRCE

	Wal	Exx	GM	Ford	GE	CPhil	Citi	IBM	AIG
Walmart	1810.0	0	-378.0	0	0	0	0	0	-10.8
Exxon	0	4409.2	0	0	0	-1424.1	0	0	-8.4
GM	-378.0	0	2741.3	-1459.2	-203.5	0	-363.7	-56.0	-104.9
Ford	0	0	-1459.2	1247.4	0	0	0	0	0
GE	0	0	-203.4	0	2599.1	0	-183.7	-1358.1	-128.5
CPhillips	0	-1424.1	0	0	0	2908.2	0	0	-264.3
Citigroup	0	0	-363.7	0	-183.7	0	4181.7	0	-718.1
IBM	0	0	-56.1	0	-1358.1	0	0	3353.5	-3.6
AIG	-10.8	-8.4	-104.9	0	-128.5	-264.3	-718.1	-3.6	1714.2

has roughly the same performance in terms of model error. These methods were shown to outperform q separate lasso penalized regressions (which ignore the correlation in the responses) in simulations when the responses are highly correlated, even when the inverse error covariance is dense.

Although we considered simultaneous L_1-penalization of $\boldsymbol{B}$ and $\boldsymbol{\Omega}$, one could use other penalties that introduce less bias instead, such as SCAD (Fan and Li, 2001; Lam and Fan, 2009). In addition, this work could be extended to the situation when the response vector samples have serial correlation, in which case the model would involve both the error

covariance and the correlation among the samples.

6 Acknowledgments

We thank Ming Yuan for providing the weekly log-returns dataset. We also thank the Associate Editor and two referees for their helpful suggestions. This research has been supported in part by the Yahoo PhD student fellowship (A.J. Rothman) and National Science Foundation grants DMS-0805798 (E.Levina), DMS-0705532 and DMS-0748389 (J.Zhu).

Appendix: Derivation of Algorithm 1

The objective function for $\boldsymbol{\Omega}$ fixed at $\boldsymbol{\Omega}_0$ is now,

$$f(\boldsymbol{B}) = g(\boldsymbol{B}, \boldsymbol{\Omega}_0) + \lambda_2 \sum_{j=1}^{p} \sum_{k=1}^{q} |b_{jk}|.$$

We can solve for $\boldsymbol{B}$ with cyclical coordinate descent. Express the directional derivatives as,

$$\begin{aligned}
\frac{\partial f^+}{\partial \boldsymbol{B}} &= \frac{2}{n}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{B}\boldsymbol{\Omega} - \frac{2}{n}\boldsymbol{X}^T\boldsymbol{Y}\boldsymbol{\Omega} + \lambda_2 \mathbf{1}(b_{ij} \geq 0) - \lambda_2 \mathbf{1}(b_{ij} < 0) \\
\frac{\partial f^-}{\partial \boldsymbol{B}} &= -\frac{2}{n}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{B}\Omega + \frac{2}{n}\boldsymbol{X}^T\boldsymbol{Y}\boldsymbol{\Omega} - \lambda_2 \mathbf{1}(b_{ij} > 0) + \lambda_2 \mathbf{1}(b_{ij} \leq 0),
\end{aligned}$$

where the indicator $\mathbf{1}()$ is understood to be a matrix. Let $\boldsymbol{S} = \boldsymbol{X}^T\boldsymbol{X}$ and $\boldsymbol{H} = \boldsymbol{X}^T\boldsymbol{Y}\boldsymbol{\Omega}$ and $u_{rc} = \sum_{j=1}^{p}\sum_{k=1}^{q} b_{jk}s_{rj}\omega_{kc}$. To update a single parameter b_{rc} we have the directional derivatives,

$$\begin{aligned}
\frac{\partial f^+}{\partial b_{rc}} &= u_{rc} - h_{rc} + n\lambda_2 1(b_{ij} \geq 0) - n\lambda_2 1(b_{ij} < 0) \\
\frac{\partial f^-}{\partial b_{rc}} &= -u_{rc} + h_{rc} - n\lambda_2 1(b_{ij} > 0) + n\lambda_2 1(b_{ij} \leq 0).
\end{aligned}$$

Let b^0_{rc} be our current iterate. The unpenalized univariate minimizer $\hat{b}^*_{rc}$ solves,

$$\hat{b}^*_{rc} s_{rr}\omega_{cc} - b^0_{rc} s_{rr}\omega_{cc} + u_{rc} - h_{rc} = 0,$$

implying $\hat{b}^*_{rc} = b^0_{rc} + \frac{h_{rc}-u_{rc}}{s_{rr}\omega_{cc}}$. If $\hat{b}^*_{rc} > 0$, then we look leftward and by convexity the penalized minimizer is $\hat{b}_{rc} = \max(0, \hat{b}^*_{rc} - \frac{n\lambda_2}{s_{rr}\omega_{cc}})$. Similarly if $\hat{b}^*_{rc} < 0$ then we look to the right and by convexity the penalized univariate minimizer is $\hat{b}_{rc} = \min(0, \hat{b}^*_{rc} + \frac{n\lambda_2}{s_{rr}\omega_{cc}})$, thus $\hat{b}_{rc} = \text{sign}(\hat{b}^*_{rc})(|\hat{b}^*_{rc}| - \frac{n\lambda_2}{s_{rr}\omega_{cc}})_+$. Also if $\hat{b}^*_{rc} = 0$, which has probability zero, then both the loss and penalty part of the objective function are minimized and the parameter stays at 0. We can write this solution as,

$$\hat{b}_{rc} = \text{sign}\left(b^0_{rc} + \frac{h_{rc}-u_{rc}}{s_{rr}\omega_{cc}}\right)\left(\left|b^0_{rc} + \frac{h_{rc}-u_{rc}}{s_{rr}\omega_{cc}}\right| - \frac{n\lambda_2}{s_{rr}\omega_{cc}}\right)_+ .$$

Supplemental Materials

R-package for MRCE: R-package "MRCE" containing functions to compute MRCE and its approximation as well as the dataset of weekly log-returns of 9 stocks from 2004 analyzed in Section 4. (MRCE_1.0.tar.gz, GNU zipped tar file)

References

Anderson, T. (1951). Estimating linear restrictions on regression coefficients for multivariate normal distributions. *Ann. Math. Statist.*, 22:327–351.

Bair, E., Hastie, T., Paul, D., and Tibshirani, R. (2006). Prediction by supervised principal components. *J. Amer. Statist. Assoc.*, 101(473):119–137.

Bazaraa, M. S., Sherali, H. D., and Shetty, C. M. (2006). *Nonlinear Programming: Theory and Algorithms*. Wiley, New Jersey, 3rd edition.

Bedrick, E. and Tsai, C. (1994). Model selection for multivariate regression in small samples. *Biometrics*, 50:226–231.

Breiman, L. and Friedman, J. H. (1997). Predicting multivariate responses in multiple linear regression (Disc: p37-54). *J. Roy. Statist. Soc., Ser. B*, 59:3–37.

Brown, P., Vannucci, M., and Fearn, T. (2002). Bayes model averaging with selection of regressors. *J. R. Statist. Soc. B*, 64:519–536.

d'Aspremont, A., Banerjee, O., and El Ghaoui, L. (2008). First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and its Applications*, 30(1):56–66.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.*, 96(456):1348–1360.

Friedman, J., Hastie, T., and Tibshirani, R. (2007). Pathwise coordinate optimization. *Annals of Applied Statistics*, 1(2):302–332.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Fujikoshi, Y. and Satoh, K. (1997). Modified AIC and C_p in multivariate linear regression. *Biometrika*, 84:707–716.

Izenman, A. J. (1975). Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248–264.

Lam, C. and Fan, J. (2009). Sparsistency and rates of convergence in large covariance matrices estimation. *Annals of Statistics*. To appear.

Lu, Z. (2009). Smooth optimization approach for sparse covariance selection. *SIAM Journal on Optimization*, 19(4):1807–1827.

Lu, Z. (2010). Adaptive first-order methods for general sparse inverse covariance selection. *SIAM Journal on Matrix Analysis and Applications*.

Obozinski, G., Wainwright, M. J., and Jordan, M. I. (2008). Union support recovery in high-dimensional multivariate regression. Technical Report 761, UC Berkeley, Department of Statistics.

Peng, J., Zhu, J., Bergamaschi, A., Han, W., Noh, D.-Y., Pollack, J. R., and Wang, P. (2009). Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer. *Annals of Applied Statistics*. To appear.

Reinsel, G. (1997). *Elements of Multivariate Time Series Analysis*. Springer, New York, 2nd edition.

Reinsel, G. and Velu, R. (1998). *Multivariate Reduced-rank Regression: Theory and Applications*. Springer, New York.

Rothman, A. J., Bickel, P. J., Levina, E., and Zhu, J. (2008). Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc., Ser. B*, 58:267–288.

Tseng, P. (1988). Coordinate ascent for maximizing nondifferentiable concave functions. Technical Report LIDS-P, 1840, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems.

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

Witten, D. M. and Tibshirani, R. (2009). Covariance-regularized regression and classification for high-dimensional problems. *Journal of the Royal Statistical Society, Series B*, 71(3):615–636.

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*, 69(3):329–346.

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

Yuan, M. and Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35.