Transposable Regularized Covariance Models with an Application to Missing Data Imputation

Genevera I. Allen
&
Robert Tibshirani

November, 11, 2008

Abstract

Missing data estimation is an important challenge with high-dimensional data arranged in the form of a matrix. Typically this data is *transposable*, meaning that either the rows, columns or both can be treated as features. To model transposable data, we present a modification of the matrix-variate normal, the *mean-restricted matrix-variate normal*, in which the rows and columns each have a separate mean vector and covariance matrix. We extend regularized covariance models, which place an additive penalty on the inverse covariance matrix, to this distribution, by placing separate penalties on the covariances of the rows and columns. These so called transposable regularized covariance models allow for maximum likelihood estimation of the mean and non-singular covariance matrices. Using these models, we formulate EM-type algorithms for missing data imputation in both the multivariate and transposable frameworks. Exploiting the structure of our transposable models, we present techniques enabling use of our models with high-dimensional data and give a computationally feasible one-step approximation for imputation. Simulations and results on microarray data and the Netflix data show that these imputation techniques often outperform existing methods and offer a greater degree of flexibility.

1 Introduction

As large datasets have become more common in biological and data mining applications, missing data imputation is a significant challenge. Imputation is an essential pre-processing step for data where values are missing often because of a technological or measurement error. In these cases, the imputed values can greatly influence the subsequent analysis and results. Another instance of missing data occurs when values are not missing due to error, but instead are simply unobserved, like unrated movies or products. Imputation, or prediction of these unobserved events, is a central goal of this type of data. Hence, good methods for missing data imputation are critical to analysis of high-dimensional data.

We motivate missing data imputation in large, matrix datasets with the example of the Netflix movie rating data (2). This dataset has around 18,000 movies (columns) and several hundred thousand customers (rows). Customers have rated some of the movies, but the data matrix is very sparse with a only small percentage of the ratings present. The goal is to predict the ratings for unrated movies so as to better recommend movies to customers. Many methods, such as the singular value decomposition (SVD) method which uses the right singular vectors (linear combinations of movies) as predictors to estimate missing values, or

k-nearest neighbors and local least squares, which use the k most correlated movies to predict ratings (29; 1; 16), must use only the columns, or movies as "features" from which to base predictions. These methods then ignore the relationships between the customers (or rows), considering the customers to be independent. This assumption, however, may be untrue since groups of customers may act as one unit, and ratings by other customers have an affect on which movies a particular customer chooses to watch. Hence, methods are generally applied to either the rows or the columns. Bell and Koren illustrate this in their discussion of imputation for the Netflix data calling all of their methods either "movie-centric" or "user-centric" (1).

We propose a unified model that incorporates possible dependences between (and among) both the rows (customers) and columns (movies). Thus, our model is *transposable* in the sense that it treats both the rows and columns as features. The model is based on the matrix-variate normal distribution, which has a separate covariance matrix parameters for both the rows and the columns. If matrix-variate normal data is strung out in a long vector, then it is distributed as multivariate normal with the covariance related to the original row and column covariance matrices through their Kronecker product. In practice, however, the matrix-variate distribution has only been of theoretical interest and has rarely applied to real datasets (15). One reason for this is that the mean and covariance parameters can only be estimated if there is a large group of matrix-distributed data (7), and then at least one of the estimated covariances is non-singular. In addition, the high-dimensional nature of the parameters have made the matrix-variate normal too difficult to work with computationally.

In this paper, we introduce modifications of the matrix-variate normal distribution that allow us to fit this transposable model to a single matrix of data. Specifically, restrictions on the means and penalties on the covariances allow for simple (and non-singular) parameter estimation. Also, our parameterization means that every marginal distribution of the matrix-data, is multivariate normal. With this theoretical foundation, we present Expectation Maximization-type (EM) algorithms for missing data imputation. These efficient algorithms exploit the special block structure of the matrix-variate normal distribution, without which the computations would be infeasible even for moderate size datasets. We develop a two-step process for calculating conditional distributions and an algorithm for calculating conditional expectations of scattered missing data that has the computational cost of comparable multivariate methods. These contributions allow one to fit this parametric transposable model to a single data matrix at reasonable computational cost, opening the door to potentially numerous applications.

We organize the paper beginning with a review of the multivariate regularized covariance models, Section 2, that form the foundation for the transposable regularized covariance models introduced in Section 3. We then present new EM-type imputation algorithms for multivariate data, Section 4, and for transposable data, Section 5, along with a one-step approximation in Section 5.2. Simulation and results on microarray and the Netflix data are given in Section 6 and we conclude with a discussion of our methods in Section 7.

2 Regularized Covariance Models

Several recent papers have presented algorithms and discussed applications of regularized covariance models (RCM) for the multivariate normal distribution (12), (30). These models regularize the maximum likelihood estimate of the covariance matrix by placing an additive penalty on the inverse covariance or concentration matrix. The resulting estimates are non-singular, thus enabling covariance estimation when the number of features is greater than the number of observations. We give a brief review of these models.

Take $X_i \sim N(0, \mathbf{\Delta})$ for $i = 1 \dots n$, *i.i.d.* observations and p features. Thus, our data matrix, $\mathbf{X}$ is $n \times p$ with covariance matrix $\mathbf{\Delta} \in \Re^{p \times p}$. The penalized log-likelihood is proportional to

$$\ell(\mathbf{\Delta}) = \frac{n}{2}\log|\,\mathbf{\Delta}^{-1}\,| - \frac{1}{2}\mathrm{tr}(\mathbf{X}^{\mathrm{T}}\,\mathbf{X}\,\mathbf{\Delta}^{-1}) - \rho||\,\mathbf{\Delta}^{-1}\,||^{q} \tag{1}$$

where $||\cdot||^q = \sum_{i=1}^{p^2} |\cdot|^q$ and q is either 1 or 2, i.e. the sum of the absolute value or square of the elements of $\mathbf{\Delta}^{-1}$. The penalty parameter is ρ. With an L_2 penalty, we can write the penalty term as $\rho\,\mathrm{tr}(\mathbf{\Delta}^{-1}\,\mathbf{\Delta}^{-1}) = \rho||\,\mathbf{\Delta}^{-1}\,||_{\mathrm{F}}^2$.

Maximizing $\ell(\mathbf{\Delta})$ gives the penalized-maximum likelihood estimate (MLE) of $\mathbf{\Delta}$. Friedman *et. al* (12) present the graphical lasso algorithm to solve the problem with an L_1 penalty. The graphical lasso uses the lasso method iteratively on the rows of $\hat{\mathbf{\Delta}}^{-1}$, and gives a sparse solution for $\hat{\mathbf{\Delta}}^{-1}$. With an L_2 penalty, the problem has an analytical solution (30). If we take the singular value decomposition (SVD) of $\mathbf{X}$, $\mathbf{X} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^{\mathrm{T}}$, with $d = \mathrm{diag}(\mathbf{D})$, then

$$\hat{\mathbf{\Delta}} = \mathbf{V}\,\mathrm{diag}(\theta)\,\mathbf{V}^{\mathrm{T}}, \qquad \theta_i = \frac{d_i^2 + \sqrt{d_i^4 + 16n\rho}}{2n}. \tag{2}$$

Thus, the inclusion of the L_2 penalty simply regularizes the eigenvalues of the covariance matrix. When $p > n$ and letting r be the rank of $\mathbf{X}$, the final $n - r$ values of θ are constant and are equal to $2\sqrt{\rho/n}$. While a rank-k SVD approximation uses only the first k eigenvalues, the L_2 RCM gives a covariance estimate with all non-zero eigenvalues. Regularized covariance models provide an alternative method of estimating the covariance matrix with many desirable properties (25).

3 Transposable Regularized Covariance Models

Our aim is to find an appropriate model for transposable data to capture the possible dependencies among the rows and columns. For this, we turn to the matrix-variate normal distribution which models the row/column dependence as a Kronecker product between the separate row covariance matrix and column covariance matrix. This distribution is the matrix equivalent of multivariate normal distribution, thus allowing for straightforward extensions of the multivariate regularized covariance models.

In this section, we first present a modification of the matrix-variate normal, the mean-restricted matrix-variate normal distribution. We confine the means to limit the total number of parameters and to provide interpretable marginal distributions. We then introduce our transposable regularized covariance models by applying penalties to the covariances of our matrix-variate distribution. Finally, we present the penalized-maximum likelihood estimates for our models and briefly assess the performance of our estimates.

3.1 Mean-Restricted Matrix-variate Normal Distribution

We introduce the mean-restricted matrix-variate normal, a variation on the matrix-variate normal, presented by Gupta and Nagar (15). A restriction on the means is needed because the matrix-variate normal has a mean matrix, $\mathbf{M}$, of the same dimension as $\mathbf{X}$, meaning that there are $n \times p$ mean parameters. Since the matrix-variate normal is mostly applied in instances where there are several independent samples of the random matrix $\mathbf{X}$ (7), this parameter formulation is appropriate. We propose, however, to use the model when we only have one matrix $\mathbf{X}$ from which to estimate the parameters. Also, we wish to parameterize

our model so that the marginals are multivariate normal, thus easing computations and improving interpret-ability.

We denote the mean-restricted matrix-variate normal distribution by $\mathbf{X} \sim N_{n,p}(\nu, \mu, \mathbf{\Sigma}, \mathbf{\Delta})$ with $\mathbf{X} \in \Re^{n \times p}$, the row mean $\nu \in \Re^n$, the column mean $\mu \in \Re^p$, the row covariance $\mathbf{\Sigma} \in \Re^{n \times n}$ and the column covariance $\mathbf{\Delta} \in \Re^{p \times p}$. If we place the matrix $\mathbf{X}$ into a vector of length np, we have $\text{vec}(\mathbf{X}) \sim N(\text{vec}(\mathbf{M}), \mathbf{\Omega})$ where $\mathbf{M} = \nu \mathbf{1}_{(p)}^T + \mathbf{1}_{(n)} \mu^T$, and $\mathbf{\Omega} = \mathbf{\Delta} \otimes \mathbf{\Sigma}$. Thus, our mean-restricted matrix-variate normal model is a multivariate normal with a mean matrix composed of additive elements from the row and column mean vectors and a covariance matrix given by the Kronecker product between the row and column covariance matrices. To further illustrate this model, we note that the rows and columns are both marginally multivariate normal. The i^{th} row, denoted as $X_{i\cdot}$, is distributed as, $X_{i\cdot} \sim N(\nu_i + \mu, \Sigma_{ii}\, \mathbf{\Delta})$ and the j^{th} column denoted by $X_{\cdot j}$, is distributed as $X_{\cdot j} \sim N(\nu + \mu_j, \Delta_{jj}\, \mathbf{\Sigma})$. Thus, a single element, X_{ij} has mean $\mu_i + \nu_j$ along with variance $\mathbf{\Sigma}_{ii}\, \mathbf{\Delta}_{jj}$, a mean and variance component from each the row and column to which it belongs. Similarly, two elements from different rows or columns are distributed as a bivariate normal, $(X_{ij}, X_{i'j'}) \sim N\left(\begin{pmatrix} \mu_i + \nu_j \\ \mu_{i'} + \nu_{j'} \end{pmatrix}, \begin{pmatrix} \mathbf{\Sigma}_{ii}\, \mathbf{\Delta}_{jj} & \mathbf{\Sigma}_{ii'}\, \mathbf{\Delta}_{jj'} \\ \mathbf{\Sigma}_{i'i}\, \mathbf{\Delta}_{j'j} & \mathbf{\Sigma}_{i'i'}\, \mathbf{\Delta}_{j'j'} \end{pmatrix} \right)$. In fact, the familiar multivariate normal distribution is a special case of the mean-restricted matrix-variate normal as seen by the following two statements. If $\mathbf{\Sigma} = \mathbf{I}$ and $\nu = \mathbf{0}$ then, $\mathbf{X} \sim N(\mu, \mathbf{\Delta})$, and if $\mathbf{\Delta} = \mathbf{I}$ and $\mu = \mathbf{0}$ then, $\mathbf{X} \sim N(\nu, \mathbf{\Sigma})$.

For completeness, the density function of this distribution is

$$p(\nu, \mu, \mathbf{\Sigma}, \mathbf{\Delta}) = (2\pi)^{-\frac{np}{2}} \mid \mathbf{\Sigma} \mid^{-\frac{p}{2}} \mid \mathbf{\Delta} \mid^{-\frac{n}{2}} \\ \text{etr}\left(-\frac{1}{2} (\mathbf{X} - \nu \mathbf{1}_{(p)}^T - \mathbf{1}_{(n)} \mu^T)\, \mathbf{\Delta}^{-1} (\mathbf{X} - \nu \mathbf{1}_{(p)}^T - \mathbf{1}_{(n)} \mu^T)^T\, \mathbf{\Sigma}^{-1} \right),$$

where etr() is the exponential of the trace function. Hence, our formulation of the matrix-variate normal distribution adds restrictions on the means, giving the distribution desirable properties in terms of its marginals and easing computation of parameter estimates, discussed in the following section.

3.2 Transposable Regularized Covariance Model (TRCM)

The matrix-variate normal distribution has been used in the theoretical literature and has been well studied from this perspective (15; 7). Application of the model, however, is lacking due to several reasons. One of the most significant drawbacks is the large number of parameters relative to the size of the data. In the previous section, we have reformulated the distribution to limit the mean parameters and in this section, we regularize the covariance parameters. This allows us to obtain non-singular covariance estimates which are important for use in any application, including missing data imputation.

As in the multivariate case, we seek to penalize the inverse covariance matrix. Instead of penalizing the overall covariance, $\mathbf{\Omega}$, we add two separate penalty terms, penalizing the inverse covariance of the rows and of the columns. The penalized log-likelihood is thus

$$\ell(\nu, \mu, \mathbf{\Sigma}, \mathbf{\Delta}) = \frac{p}{2} \log \mid \mathbf{\Sigma}^{-1} \mid + \frac{n}{2} \log \mid \mathbf{\Delta}^{-1} \mid \\ - \frac{1}{2} \text{tr}\left(\mathbf{\Sigma}^{-1} (\mathbf{X} - \nu \mathbf{1}_{(p)}^T - \mathbf{1}_{(n)} \mu^T)\, \mathbf{\Delta}^{-1} (\mathbf{X} - \nu \mathbf{1}_{(p)}^T - \mathbf{1}_{(n)} \mu^T)^T \right) \\ - \rho_r || \mathbf{\Sigma}^{-1} ||^{q_r} - \rho_c || \mathbf{\Delta}^{-1} ||^{q_c}. \tag{3}$$

where $|| \cdot ||^{q_r} = \sum_{i=1}^{m^2} | \cdot |^{q_r}$ and q_r and q_c are either 1 or 2, i.e. the sum of the absolute value of the matrix elements or squared elements. ρ_r and ρ_c are the two penalty parameters.

Note that we will refer to the four possible types of penalties as $L_{q_r} : L_{q_c}$. Placing separate penalties on the two covariance matrices is not equivalent to placing a single penalty on the Kronecker product covariance matrix $\mathbf{\Omega}$. Using two separate penalties gives greater flexibility, as the covariance of the rows and columns can be modeled separately using differing penalties and penalty parameters. Also, having two penalty terms leads to simple parameter estimation strategies.

3.3 Parameter Estimation

We estimate the means and covariances via penalized maximum likelihood estimation. The estimates, however, are not unique, but the overall mean, $\hat{\mathbf{M}}$, and overall covariance $\hat{\mathbf{\Omega}}$ are unique. Hence, $\hat{\nu}$ and $\hat{\mu}$ are unique up to an additive constant and $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Delta}}$ are unique up to a multiplicative constant. We first begin with the maximum likelihood estimation of the mean parameters.

Proposition 1. *The MLE estimates for ν and μ are*

$$\hat{\nu} = \sum_{j=1}^{p} \frac{(X_{\cdot j} - \hat{\mu}_j)}{p}, \qquad \hat{\mu} = \sum_{i=1}^{n} \frac{(X_{i\cdot} - \hat{\nu}_i)}{n}. \tag{4}$$

where $X_{\cdot j}$ denotes the j^{th} column and $X_{i\cdot}$ the i^{th} row of $\mathbf{X} \in \Re^{n \times p}$.

Proof. See Appendix A.

The estimates for ν and μ are obtained by centering with respect to the rows and then the columns. Note that centering by the columns first will change $\hat{\mu}$ and $\hat{\nu}$, but will still give the same additive result. Thus, the order in which we center is unimportant.

Maximum likelihood estimation of the covariance matrices is more difficult. Here, we will assume that the data has been centered, $\mathbf{M} = \mathbf{0}$. Then, the penalized log-likelihood, $\ell(\mathbf{\Sigma}, \mathbf{\Delta})$, is a biconcave function of $\mathbf{\Sigma}^{-1}$ and $\mathbf{\Delta}^{-1}$. In words, this means that for any fixed $\mathbf{\Sigma}^{-1\prime}$, $\ell(\mathbf{\Sigma}', \mathbf{\Delta})$ is a concave function of $\mathbf{\Delta}^{-1}$, and for any fixed $\mathbf{\Delta}^{-1\prime}$, $\ell(\mathbf{\Sigma}, \mathbf{\Delta}')$ is a concave function of $\mathbf{\Sigma}^{-1}$. We exploit this structure to maximize the penalized likelihood by iteratively maximizing along each coordinate, either $\mathbf{\Sigma}^{-1}$ or $\mathbf{\Delta}^{-1}$.

Proposition 2. *Iterative block coordinate-wise maximization of $\ell(\mathbf{\Sigma}, \mathbf{\Delta})$ with respect to $\mathbf{\Sigma}^{-1}$ and $\mathbf{\Delta}^{-1}$ converges to a stationary point of $\ell(\mathbf{\Sigma}, \mathbf{\Delta})$ for both L_1 and L_2 penalty types.*

Proof. See Appendix A.

While block coordinate-wise maximization (Proposition 2) reaches a stationary point of $\ell(\mathbf{\Sigma}, \mathbf{\Delta})$, it is not guaranteed to reach the global maximum. There are potentially many stationary points, especially with L_1 penalties, due to the high-dimensional nature of the parameter space. Thus, starting the coordinate-wise maximization at random initial values often yields different stationary points. We make note of a few straightforward properties of the coordinate-wise maximization procedure, namely that each iteration monotonically increases the penalized log-likelihood and the order of maximization is unimportant.

The coordinate-wise maximization is accomplished by setting the gradients with respect to $\mathbf{\Sigma}^{-1}$ and $\mathbf{\Delta}^{-1}$ equal to zero and solving. We list the gradients with L_2 penalties. With L_1 penalties only the third term is changed and is given in parentheses.

$$\begin{aligned}
\frac{\partial \ell}{\partial \mathbf{\Sigma}^{-1}} &= \mathbf{\Sigma} - \mathbf{X}\,\mathbf{\Delta}^{-1}\,\mathbf{X}^{\mathrm{T}}/p - \frac{4\rho_r}{p}\,\mathbf{\Sigma}^{-1}\left(\frac{2\rho_r}{p}\mathrm{sign}(\mathbf{\Sigma}^{-1})\right) \\
\frac{\partial \ell}{\partial \mathbf{\Delta}^{-1}} &= \mathbf{\Delta} - \mathbf{X}^{\mathrm{T}}\,\mathbf{\Sigma}^{-1}\,\mathbf{X}/n - \frac{4\rho_c}{n}\,\mathbf{\Delta}^{-1}\left(\frac{2\rho_c}{n}\mathrm{sign}(\mathbf{\Delta}^{-1})\right)
\end{aligned} \tag{5}$$

Maximization with L_1 penalties is achieved by applying the graphical lasso algorithm to the second term with the coefficient of the third term the as the penalty parameter. With L_2 penalties, we maximize by taking the eigenvalue decomposition of the second term and regularizing the eigenvalues as in the multivariate case, (2). Thus, coordinate-wise maximization leads to a simple iterative algorithm, but it comes at a cost since it does not necessarily converge to the global maximum. When both penalty terms are L_2 penalties, however, we can find the global maximum.

3.3.1 Covariance Estimation for L_2 Penalties

Covariance estimation when both penalties of the transposable regularized covariance model are L_2 penalties reduces to a minimization problem involving the eigenvalues of the covariance matrices. This problem has a unique analytical solution, and thus our estimates, $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Delta}}$, are globally optimal.

Theorem 1. *The global unique solution maximizing $\ell(\mathbf{\Sigma}, \mathbf{\Delta})$ with L_2 penalties on both covariance parameters is given by the following: Denote the SVD of $\mathbf{X}$ as $\mathbf{X} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^{\mathrm{T}}$ with $d = \mathrm{diag}(\mathbf{D})$ and let r be the rank of $\mathbf{X}$, then*

$$\mathbf{\Sigma}^* = \mathbf{U}\,\mathrm{diag}(\beta^*)\,\mathbf{U}^{\mathrm{T}} \quad \& \quad \mathbf{\Delta}^* = \mathbf{V}\,\mathrm{diag}(\theta^*)\,\mathbf{V}^{\mathrm{T}} \tag{6}$$

where $\beta^* \in \Re^{n+}$ and $\theta^* \in \Re^{p+}$ given by

$$\beta_i^* = \begin{cases} 2\sqrt{\frac{\rho_r}{p}} & \text{if } i \geq r, \\ \sqrt{\frac{-c_2^{(i)} - \sqrt{{c_2^{(i)}}^2 - 4c_1^{(i)}c_3^{(i)}}}{2c_1^{(i)}}} & \text{otherwise.} \end{cases} \qquad \theta_i^* = \begin{cases} 2\sqrt{\frac{\rho_c}{n}} & \text{if } i \geq r \\ \frac{d_i^2\beta_i^*}{p{\beta_i^*}^2 - 4\rho_r} & \text{otherwise.} \end{cases}$$

with coefficients

$$c_1^{(i)} = -4\rho_c p^2, \quad c_2^{(i)} = 32\rho_r\rho_c p + d_i^4(n-p), \quad \& \quad c_3^{(i)} = 4\rho_r(d_i^4 - 16\rho_r\rho_c).$$

Proof. See Appendix A.

With L_2 penalties, maximum likelihood covariance estimates $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Delta}}$ have eigenvectors given by the left and right singular vectors of $\mathbf{X}$ respectively. To reveal some intuition as to how these covariance estimates compare to other possible eigenvalue regularization methods, we present the two gradient equations in terms of the eigenvalues β and θ. (These are discussed fully in the proof of Theorem 1).

$$p\theta\beta^2 - d^2\beta - 4\rho_r\theta = 0 \quad \& \quad n\beta\theta^2 - d^2\theta - 4\rho_c\beta = 0.$$

These are two quadratic functions in β and θ, so using the quadratic formula, this gives us the eigenvalues in terms of each other.

$$\beta = \frac{d^2 + \sqrt{d^4 + 16p\rho + r\theta}}{2p\theta} \quad \& \quad \theta = \frac{d^2 + \sqrt{d^4 + 16n\rho_c\beta}}{2n\beta}.$$

From this we see that the eigenvalues regularize the square of the singular values by a function of the dimensions, the penalty parameters and the eigenvalues of the other covariance estimate. Thus from Theorem 1, $L_2 : L_2$ covariance estimation has a unique and globally optimal solution, which cannot be said of the other combinations of penalties.

3.3.2 Covariance Estimation Results

We investigate the accuracy of our transposable regularized covariance estimates through simulation using the Kullback-Leibler divergence as the metric. As the focus of this paper is on the application of our models to missing data imputation, we compare our covariance estimates to simple shrinkage estimates for completeness. We will assume that our data has mean zero an has previously been centered as in Proposition 1, and the Kullback-Leibler (K-L) divergence is given in Proposition 3.

Proposition 3. *The Kullback-Leibler divergence for the mean-restricted matrix-variate normal with mean zero is*

$$\mathrm{E}_{(\mathbf{\Sigma},\mathbf{\Delta})}\left[\ell(\mathbf{\Sigma},\mathbf{\Delta}) - \ell(\hat{\mathbf{\Sigma}},\hat{\mathbf{\Delta}})\right] = \frac{p}{2}\log|\hat{\mathbf{\Sigma}}\,\mathbf{\Sigma}^{-1}| + \frac{n}{2}\log|\hat{\mathbf{\Delta}}\,\mathbf{\Delta}^{-1}| - \frac{np}{2} + \frac{1}{2}\mathrm{tr}\left(\hat{\mathbf{\Sigma}}^{-1}\,\mathbf{\Sigma}\right)\mathrm{tr}\left(\hat{\mathbf{\Delta}}^{-1}\,\mathbf{\Delta}\right).$$

Proof. See Appendix A.

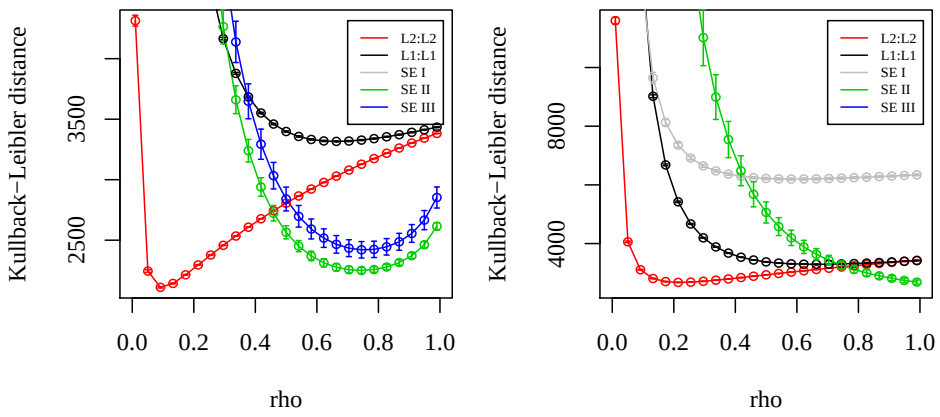

Figure 1: Mean Kullback-Leibler divergence with standard errors of 50 datasets of dimension 50×50 simulated from the matrix-variate normal distribution. Autoregressive (left) and blocked diagonal (right) covariance matrices are used as given in Section 3.3.2. The divergence is compared between two transposable regularized covariance estimates, $L_2 : L_2$ and $L_1 : L_1$,with the penalty parameters on both covariance matrices equal to ρ and three families of shrinkage estimators, SE I, SE II, and SE III described in Section 3.3.2 indexed by ρ.

In Figure 1, we compare covariance estimates of the $L_1 : L_1$ and $L_2 : L_2$ transposable regularized covariance models to three other shrinkage covariance estimates parameterized by ρ using the Kullback-Leibler (K-L) distance. The data of dimension 50×50 was simulated under the matrix-variate normal model with autoregressive covariances, left, and blocked diagonal covariances, right. The covariances are as follows:

- Autoregressive: $\mathbf{\Sigma}_{ij} = 0.8^{|i-j|}$ and $\mathbf{\Delta}_{ij} = 0.6^{|i-j|}$.
- Blocked diagonal: Off-diagonal elements of 5×5 blocks of $\mathbf{\Sigma}$ are 0.8 and of $\mathbf{\Delta}$, 0.6.

In the TRCM estimates, both penalty parameters, ρ_r and ρ_c are set to ρ for comparability. The three comparison transposable shrinkage covariance estimators are straightforward extensions of simple multivariate shrinkage estimates.

- SE I: Denote the SVD of $\mathbf{X}$ by $\mathbf{X} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^{\mathrm{T}}$, then
  - $\hat{\mathbf{\Sigma}}_{\mathrm{SE\,I}} = \mathbf{U}(\mathbf{D}^2 + \rho\mathbf{I})\,\mathbf{U}^{\mathrm{T}}$, & $\hat{\mathbf{\Delta}}_{\mathrm{SE\,I}} = \mathbf{V}(\mathbf{D}^2 + \rho\mathbf{I})\,\mathbf{V}^{\mathrm{T}}$.
- SE II & III: Let S be the unpenalized MLE of $\mathbf{\Sigma}$, i.e. $S = \mathbf{X}\,\mathbf{X}^{\mathrm{T}}/p$ and D be the unpenalized MLE of $\mathbf{\Delta}$, $D = \mathbf{X}^{\mathrm{T}}\,\mathbf{X}/n$, then
  - $\hat{\mathbf{\Sigma}}_{\mathrm{SE\,II}} = \rho\,\frac{\mathrm{tr}(S)}{n}\mathbf{I} + (1-\rho)S$, & $\hat{\mathbf{\Delta}}_{\mathrm{SE\,II}} = \rho\,\frac{\mathrm{tr}(D)}{p}\mathbf{I} + (1-\rho)D$.
  - $\hat{\mathbf{\Sigma}}_{\mathrm{SE\,III}} = \rho\,\mathrm{diag}(S) + (1-\rho)S$, & $\hat{\mathbf{\Delta}}_{\mathrm{SE\,III}} = \rho\,\mathrm{diag}(D) + (1-\rho)D$.

These are taken in the spirit of the shrinkage estimators of Daniels and Kass (6), which either shrink the eigenvalues (SE I) or shrink towards a specific structure(SE II and SE III).

According to the Kullback-Leibler distance, the $L_2 : L_2$ TRCM covariance estimates are more accurate than our comparison estimators and the $L_1 : L_1$ TRCM estimates even with underlying sparsity. The latter is not unexpected since from Theorem 1, we have a unique solution compared with the possibly many sub-optimal stationary points of the $L_1 : L_1$ TRCM solution, Proposition 2. From these results, we conjecture that the $L_2 : L_2$ TRCM estimates may lead to significant improvements in missing data imputation, presented in the following sections.

4 Imputation for Multivariate Data

Missing data imputation for transposable data is the main application of this paper. It is important, however, to first discuss imputation methods in the simpler multivariate case using regularized covariance models.

The EM algorithm for the multivariate normal has been traditionally used to impute missing values. First proposed in 1972, this method is based on maximum likelihood estimation of parameters in the presence of missing values. Imputation is simply a side result (22; 19). Many have noted that this approach is not well suited to high-dimensional data, especially when the number of features is large. Hence, a common remedy to this problem, seen predominately in image processing, is to penalize the log-likelihood, (11), (23), (14), (10).

We present a new penalized EM algorithm for imputation with regularized covariance models, methods that provide the necessary background for the transposable methods and later form an integral part of our one-step approximation (Section 5). Our method maximizes the penalized log-likelihood of the observed data given below. First, however, we assume that our data $x_i \sim N(\mu, \mathbf{\Delta})$ with $i = 1 \ldots n$ observations and p features, with values missing at random. We let x_{i,o_i} denote the observed components of observation i and x_{i,m_i} denote the missing components. For each observation, we partition the mean and covariance to correspond to the observed parts of observation i and denote these as μ_{o_i} and $\mathbf{\Delta}_{o_i,o_i}$.

The penalized observed data log-likelihood is:

$$\ell_{\mathrm{obs}}(\mu, \mathbf{\Delta}) = \frac{1}{2}\sum_{i=1}^{n}\Big[\log|\,\mathbf{\Delta}_{o_i,o_i}\,| - (x_{o_i} - \mu_{o_i})^T\,\mathbf{\Delta}^{-1}{}_{o_i,o_i}(x_{o_i} - \mu_{o_i})\Big] - \rho||\,\mathbf{\Delta}^{-1}\,||^q. \tag{7}$$

Notice that except for the last penalty term, this is the same observed log-likelihood that is maximized by the original multivariate normal EM algorithm (19).

Missing data imputation is part of the E step of the algorithm in which the conditional expectation of the complete-data log-likelihood is taken given the current parameter estimates as in un-penalized EM algorithms. This is denoted by $Q(\theta|\theta^{(k)}) = \mathrm{E}\left(\ell(\mu, \mathbf{\Delta})|X_o, \mu', \mathbf{\Delta}'\right)$,

letting $\theta = (\mu, \mathbf{\Delta})$ and where X_o denotes the totality of observed elements of X. We can break this down into two parts which we term the imputation and covariance correction steps. These steps come directly from the conditional distribution formulas for the multivariate normal.
Imputation Step:

$$\hat{x}_{i,j} = \mathrm{E}\left(x_{i,j}|x_{i,o_i}, \mu', \mathbf{\Delta}'\right) = \begin{cases} \mu'_{m_i} + \mathbf{\Delta}'_{m_i,o_i}\, \mathbf{\Delta}^{-1'}_{o_i,o_i}\left(x_{i,o_i} - \mu'_{o_i}\right), & \text{if } j \in m_i \\ x_{i,j}, & \text{if } j \in o_i. \end{cases}$$

Covariance Correction Step: $\mathrm{E}\left(x_{i,j}x_{i,j'}|x_{i,o_i}, \mu', \mathbf{\Delta}'\right) = \hat{x}_{i,j}\hat{x}_{i,j'} + c_{i,jj'}$,

$$c_{i,jj'} = \begin{cases} \mathbf{\Delta}'_{m_i,m_i} - \mathbf{\Delta}'_{m_i,o_i}\, \mathbf{\Delta}^{-1'}_{o_i,o_i}\, \mathbf{\Delta}'_{o_i,m_i}, & \text{if } j, j' \in m_i \\ 0, & \text{else.} \end{cases}$$

We call $c_{i,jj'}$ the covariance correction term because it is added to the cross products forming the covariance matrix. Notice that $c_{i,jj'}$ is only non-zero if both j and j' are missing. Intuitively, this correction term is needed because in the imputation step, we set the missing values equal to their conditional expectations and hence lose some of the variance associated with these elements.

The Maximization step is where our algorithm differs from that of Little and Rubin (19). In the M step, we must maximize $Q(\theta|\theta^{(k)})$ with respect to θ to obtain the new estimate $\theta^{(k+1)}$, giving

$$Q(\theta|\theta^{(k)}) = \frac{n}{2}\log|\,\mathbf{\Delta}^{-1}\,| - \frac{1}{2}\mathrm{tr}\left(\hat{\mathbf{\Delta}}'\,\mathbf{\Delta}^{-1}\right) - \rho||\,\mathbf{\Delta}^{-1}\,||^q,$$

$$\text{where} \qquad \hat{\mathbf{\Delta}}'_{jj'} = \sum_{i=1}^{n}\left[(\hat{x}_{ij} - \mu_j)(\hat{x}_{ij} - \mu_j) + c_{i,jj'}\right].$$

The computations in the E step come into Q through $\hat{\mathbf{\Delta}}'$. Maximizing with respect to μ, gives the estimate $\hat{\mu}_j = \sum_{i=1}^{n} = \hat{x}_{ij}/n$. Replacing μ with $\hat{\mu}$ in $\hat{\mathbf{\Delta}}'$, we see that Q has the structure of the regularized covariance models. Hence, our estimate of $\hat{\mathbf{\Delta}}$ is obtained by applying either the L_1 or L_2 solvers of the RCM problem. We break this M step into two parts which we present in the imputation algorithm, Algorithm 1.

This regularized covariance model imputation approach (RCMimpute) is closely related to other penalized EM methods for missing value estimation. These algorithms give non-singular covariance estimates (14), thus enabling use of the EM framework when $p > n$. Also, our algorithm has a unified theoretical framework based on the regularized covariance models. Results on simulated and example data are given for this imputation method in Section 6.

5 Imputation for Transposable Data

Imputation methods for transposable data are the main focus of this paper. Our aim is to formulate methods based on the transposable regularized covariance models introduced in Section 3. As previously suggested, the matrix-variate normal distribution has not been widely used in applied settings largely because computational costs are prohibitive. Thus, we let computational considerations motivate the formulation of our imputation methods.

We propose an Multi-Cycle Expectation Conditional Maximization
(MCECM) algorithm, given by Meng and Rubin (21), maximizing the observed penalized log-likelihood of the transposable regularized covariance models. The algorithm exploits the

Algorithm 1 Imputation for Regularized Covariance Models (RCMimpute)

1. Initialization:
   (a) Set the missing values to the mean: $\hat{x}_{i,m_i} = \sum_{i \in o_i} x_{ij}/n_i$
   (b) Set $\mu^{(0)}$ and $\mathbf{\Delta}^{(0)}$ to the empirical mean and covariance.
2. E Step:
   (a) Imputation: Compute $\mathrm{E}\left(x_{i,j}|x_{i,o_i}, \mu^{(k)}, \mathbf{\Delta}^{(k)}\right)$.
   (b) Cross Products: Compute $\mathrm{E}\left(x_{i,j}x_{i,j'}|x_{i,o_i}, \mu^{(k)}, \mathbf{\Delta}^{(k)}\right)$.
3. M Step:
   (a) Update Estimates: $\hat{\mu}_j$ & $\hat{\mathbf{\Delta}}'_{jj'}$.
   (b) Maximize penalized log-likelihood with respect to $\mathbf{\Delta}^{-1}$ to obtain the new estimate $\hat{\mathbf{\Delta}}$.
4. Repeat steps 2-3 until convergence.

structure of our model by maximizing with respect to one block of coordinates at a time, saving considerable mathematical and computational time. First, we develop the Expectation and Conditional Maximization steps of the algorithm mathematically and present the algorithm in Section 5.1.1. We then give the rationale for structuring our algorithm as an ECM-type algorithm and support this with a brief numerical example in Section 5.1.2. Computational strategies for the algorithm are addressed in Section 5.1.3.

In high dimensional data, however, the MCECM algorithm we propose for imputation is not computationally feasible. Hence, we suggest a computation-saving one-step approximation in Section 5.2. The foundation of our approximation lies in new methods, given in Theorems 2 and 3, for calculating conditional distributions with the mean-restricted matrix-variate normal, Section 5.2.2. We demonstrate the utility of this one-step procedure in numerical examples in Section 5.2.3. Finally, we conclude with a Bayesian variation of the one-step approximation using Gibbs sampling in Section 5.2.4.

Prior to formulating the imputation algorithm for transposable models, we pause to address a logical question: Why do we not use a multivariate imputation method, such as RCMimpute, given that the mean-restricted matrix normal distribution can be written as a multivariate normal with $\mathrm{vec}(\mathbf{X}) \sim N(\mathrm{vec}(\mathbf{M}), \mathbf{\Omega})$? There are two reasons why this is inadvisable. First, notice that TRCMs place an additive penalty on both the inverse covariance matrices of the rows and the columns. The overall covariance matrix, $\mathbf{\Omega}$, however, is their Kronecker product. Thus, converting the TRCM into a multivariate form yields a messy penalty term leading to a difficult maximization step. The second reason to avoid the multivariate methods is computational. Recall that $\mathbf{\Omega}$ is a $np \times np$ matrix which is expensive to repeatedly invert. We will see that the mathematical form of the ECM imputation algorithm we propose leads to computational strategies that avoid the expensive inversion of $\mathbf{\Omega}$.

5.1 Multi-Cycle ECM Algorithm for Imputation

Before presenting the algorithm, we first review the notation used throughout the remainder of this paper. As previously mentioned, we use i to denote the rows and j the columns. The observed and missing parts of row i are o_i and m_i respectively, and o_j and m_j are the analogous parts of column j. We let m and o denote the totality of missing and observed elements respectively. Since with transposable data there is no natural orientation, we set n to always be the larger dimension of $\mathbf{X}$ and p the smaller.

5.1.1 Algorithm

We develop the ECM-type algorithm for imputation mathematically, beginning with the observed data log-likelihood which we seek to maximize. Letting $x^*_{o_j,j} = \mathbf{\Sigma}^{-1/2}_{o_j,o_j}(x_{o_j,j} - \nu_{o_j})$,

$$\ell(\nu,\mu,\mathbf{\Sigma},\mathbf{\Delta}) = \frac{1}{2}\left[\sum_{j=1}^{p}\log|\,\mathbf{\Sigma}^{-1}{}_{o_j,o_j}\,| + \sum_{i=1}^{n}|\,\mathbf{\Delta}^{-1}{}_{o_i,o_i}\,|\right] \tag{8}$$
$$-\frac{1}{2}\mathrm{tr}\left(\sum_{i=1}^{n}(x^*_{i,o_i}-\mu_{o_i})^T(x^*_{i,o_i}-\mu_{o_i})\,\mathbf{\Delta}^{-1}{}_{o_i,o_i}\right) - \rho_r||\,\mathbf{\Sigma}^{-1}\,||^{q_r} - \rho_c||\,\mathbf{\Delta}^{-1}\,||^{q_c}.$$

One can show that this is indeed the observed log-likelihood by starting with (7) and using $\mathrm{vec}(\mathbf{X})$ and the corresponding $\mathrm{vec}(\mathbf{M})$ and $\mathbf{\Omega}$. Again, we maximize (8) via the EM algorithm which, similarly to the multivariate case, gives the imputed values as a part of the Expectation step.

We present two forms of the E step, one which leads to simple maximization with respect to $\mathbf{\Sigma}^{-1}$ and the other with respect to $\mathbf{\Delta}^{-1}$. This is possible because of the structure of the matrix-variate model, specifically the trace term. Below, we give the Q function, defined in the previous section, and our equivalent forms of the trace term. Here, we assume that $\mathbf{X}$ is centered.

$$\begin{aligned} Q(\theta|\theta^{(k)}) &= \mathrm{E}\left(\ell(\nu,\mu,\mathbf{\Sigma},\mathbf{\Delta})|X_o,\nu^{(k)},\mu^{(k)},\mathbf{\Sigma}^{(k)},\mathbf{\Delta}^{(k)}\right) \\ &\propto \mathrm{E}\left[\mathrm{tr}\left(\mathbf{X}^{\mathrm{T}}\,\mathbf{\Sigma}^{-1}\,\mathbf{X}\,\mathbf{\Delta}^{-1}\right)|X_o,\theta'\right] \\ &\propto \mathrm{tr}\left[\mathrm{E}\left(\mathbf{X}^{\mathrm{T}}\,\mathbf{\Sigma}^{-1}\,\mathbf{X}\,|X_o,\theta'\right)\mathbf{\Delta}^{-1}\right] \\ &\propto \mathrm{tr}\left[\mathrm{E}\left(\mathbf{X}\,\mathbf{\Delta}^{-1}\,\mathbf{X}^{\mathrm{T}}\,|X_o,\theta'\right)\mathbf{\Sigma}^{-1}\right]. \end{aligned} \tag{9}$$

Thus, we have two equivalent forms of the Q function. We give the conditional expectations for the E step below.

Proposition 4. *The E step has the following form*

$$\begin{aligned} \mathrm{E}\left[\mathrm{tr}\left(\mathbf{X}^{\mathrm{T}}\,\mathbf{\Sigma}^{-1}\,\mathbf{X}\,\mathbf{\Delta}^{-1}\right)|X_o,\theta'\right] &= \mathrm{tr}\left[\left(\hat{\mathbf{X}}^T\,\mathbf{\Sigma}^{-1}\,\hat{\mathbf{X}} + \mathbf{G}(\mathbf{\Sigma}^{-1})\right)\mathbf{\Delta}^{-1}\right] \\ &= \mathrm{tr}\left[\left(\hat{\mathbf{X}}\,\mathbf{\Delta}^{-1}\,\hat{\mathbf{X}}^T + \mathbf{F}(\mathbf{\Delta}^{-1})\right)\mathbf{\Sigma}^{-1}\right], \end{aligned} \tag{10}$$

where $\hat{\mathbf{X}} = \mathrm{E}\left(\mathbf{X}\,|X_o,\theta'\right)$ *and*

$$\mathbf{G}(\mathbf{\Sigma}^{-1}) = \begin{pmatrix} \mathrm{tr}(\mathbf{C}^{(11)}\,\mathbf{\Sigma}^{-1}) & \dots & \mathrm{tr}(\mathbf{C}^{(1p)}\,\mathbf{\Sigma}^{-1}) \\ \vdots & \ddots & \vdots \\ \mathrm{tr}(\mathbf{C}^{(p1)}\,\mathbf{\Sigma}^{-1}) & \dots & \mathrm{tr}(\mathbf{C}^{(pp)}\,\mathbf{\Sigma}^{-1}) \end{pmatrix}, \quad \mathbf{C}^{(jj')} = \mathrm{Cov}\left(X_{\cdot j}, X_{\cdot j'}|X_o,\theta'\right),$$

$$\mathbf{F}(\mathbf{\Delta}^{-1}) = \begin{pmatrix} \mathrm{tr}(\mathbf{D}^{(11)}\,\mathbf{\Delta}^{-1}) & \dots & \mathrm{tr}(\mathbf{D}^{(1n)}\,\mathbf{\Delta}^{-1}) \\ \vdots & \ddots & \vdots \\ \mathrm{tr}(\mathbf{D}^{(n1)}\,\mathbf{\Delta}^{-1}) & \dots & \mathrm{tr}(\mathbf{D}^{(nn)}\,\mathbf{\Delta}^{-1}) \end{pmatrix}, \quad \mathbf{D}^{(ii')} = \mathrm{Cov}\left(X_{i\cdot}, X_{i'\cdot}|X_o,\theta'\right).$$

Proof. See Appendix A.

The E step in the matrix-variate normal framework has a similar structure to that of the multivariate normal with an imputation step ($\hat{\mathbf{X}}$) and a covariance correction step ($\mathbf{C}^{(jj')}$ and $\mathbf{D}^{(ii')}$). The matrices $\mathbf{C}^{(jj')} \in \Re^{n\times n}$ and $\mathbf{D}^{(ii')} \in \Re^{p\times p}$, while $\mathbf{G}(\mathbf{\Sigma}^{-1}) \in \Re^{p\times p}$ and $\mathbf{F}(\mathbf{\Delta}^{-1}) \in \Re^{n\times n}$. Note that $\mathbf{C}^{(jj')}$ is sparse and only non-zero at $\mathbf{C}^{(jj')}_{ii'}$ when x_{ij} and $x_{i'j'}$ are both missing. $\mathbf{C}^{(jj')}$ is not symmetric, but $\mathbf{C}^{(jj')^T} = \mathbf{C}^{(j'j)}$, hence $\mathbf{G}(\mathbf{\Sigma}^{-1})$ is symmetric. The matrices $\mathbf{D}^{(ii')}$ and $\mathbf{F}(\mathbf{\Delta}^{-1})$ are structured analogously. Thus, we have two equivalent forms of the Q function needed for the E step, which will be inserted between the two Conditional Maximization (CM) steps to form the MCECM algorithm.

The CM steps which maximize the Q function along either $\mathbf{\Sigma}^{-1}$ or $\mathbf{\Delta}^{-1}$ are direct extensions of the MLE solvers for the multivariate RCMs. This is easily seen from the gradients. Note that we only show the gradients with an L_2 penalty, since an L_1 penalty differs only in the last term.

$$\frac{\partial Q}{\partial \mathbf{\Sigma}^{-1}} = \mathbf{\Sigma} - \left[\hat{\mathbf{X}}\,\mathbf{\Delta}^{-1}\,\hat{\mathbf{X}}^T + \mathbf{F}(\mathbf{\Delta}^{-1})\right]/p - \frac{4\rho_r}{p}\,\mathbf{\Sigma}^{-1}\,.$$
$$\frac{\partial Q}{\partial \mathbf{\Delta}^{-1}} = \mathbf{\Delta} - \left[\hat{\mathbf{X}}^T\,\mathbf{\Sigma}^{-1}\,\hat{\mathbf{X}} + \mathbf{G}(\mathbf{\Sigma}^{-1})\right]/n - \frac{4\rho_c}{n}\,\mathbf{\Delta}^{-1}\,.$$

With an L_2 penalty, the estimate is given by taking the eigenvalue decomposition of the second term and regularizing the eigenvalues as in (2). The graphical lasso algorithm applied to the second term gives the estimate in the case with an L_1 penalty.

Algorithm 2 Imputation with Transposable Regularized Covariance Models (TRCMimpute)

1. Initialization:
   (a) Estimate $\hat{\nu}$ and $\hat{\mu}$ from the observed data.
   (b) If x_{ij} is missing, set $x_{ij} = \hat{\nu}_i + \hat{\mu}_j$.
   (c) Start with non-singular estimates $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Delta}}$.
2. E Step ($\mathbf{\Delta}$): Calculate $\hat{\mathbf{X}}^T\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{X}} + \mathbf{G}(\hat{\mathbf{\Sigma}}^{-1})$.
3. M Step ($\mathbf{\Delta}$):
   (a) Update estimates of $\hat{\nu}$ and $\hat{\mu}$.
   (b) Maximize Q with respect to $\mathbf{\Delta}^{-1}$ to obtain $\hat{\mathbf{\Delta}}$.
4. E Step ($\mathbf{\Sigma}$): Calculate $\hat{\mathbf{X}}\hat{\mathbf{\Delta}}^{-1}\hat{\mathbf{X}}^T + \mathbf{F}(\hat{\mathbf{\Delta}}^{-1})$,
5. M Step ($\mathbf{\Sigma}$):
   (a) Update estimates of $\hat{\nu}$ and $\hat{\mu}$.
   (b) Maximize Q with respect to $\mathbf{\Sigma}^{-1}$ to obtain $\hat{\mathbf{\Sigma}}$.
6. Repeat Steps 2-5 until convergence.

We now put these steps together and present the Multi-Cycle ECM algorithm for imputation with transposable data, TRCMimpute, in Algorithm 2. A brief comment regarding the

initialization of parameter estimates is needed. Estimating the mean parameters when missing values are present is not as simple as centering the rows and columns as in (4). Instead, we iterate centering by rows and columns, ignoring the missing values until convergence. This is a straightforward extension of (4) in which we simply sum over the observed values. Secondly, the initial estimates of $\hat{\mathbf{\Sigma}}^{-1}$ and $\hat{\mathbf{\Delta}}^{-1}$ must be non-singular in order to preform the needed computations in the E step. While any non-singular matrices will work, we find that the algorithm converges faster if we start with the MLE estimates with the missing values fixed and set to the estimated mean.

We have presented the mathematical derivations of the E and CM steps of our ECM-type imputation algorithm for transposable data. As stated earlier, this algorithm maximizes the observed penalized log-likelihood, (8), a point which we explain in the following section by discussing the algorithm structure and relating the algorithm to other EM-type methods. We leave all computational methods and concerns to Section 5.1.3.

5.1.2 Properties and Comparisons

We present some properties and the rationale for structuring our imputation approach with transposable models as a Multi-Cycle ECM algorithm. The MCECM algorithm is a special case of the ECM algorithm of Meng and Rubin (21), where the Maximization step is split into several Conditional Maximization (CM) steps. Expectation steps are inserted between the CM steps to form multi-cycles. Our algorithm uses a specific type of conditional maximization, maximization with respect to one block of coordinates, either $\mathbf{\Sigma}^{-1}$ or $\mathbf{\Delta}^{-1}$. These ECM-type algorithms are, as Meng and Rubin say, a type of "extended GEM" (Generalized EM) algorithm (21), and thus retain many of the properties of GEM algorithms. Most importantly, our algorithm is monotonic, increasing the observed likelihood each step, and converges to a stationary point of the observed likelihood function.

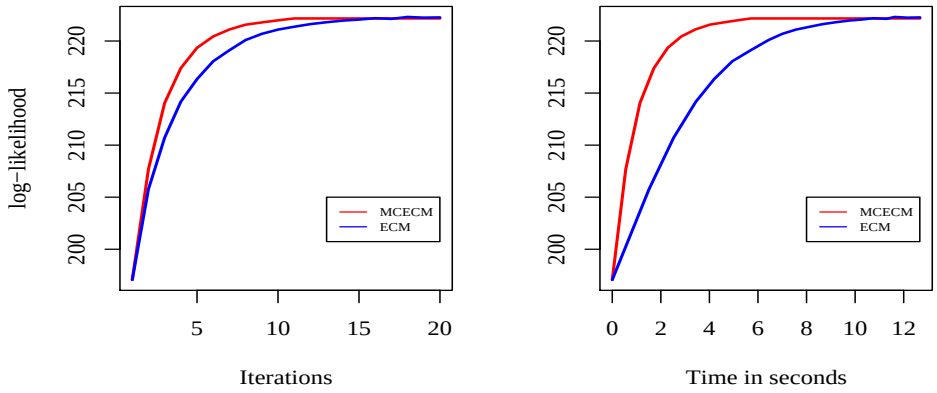

Figure 2: Comparison of convergence of the observed log-likelihood by iterations (left) and by time in seconds (right) of the two structural approaches to the transposable EM algorithm: the Multi-cycle ECM and the ECM algorithms. The MCECM algorithm took 6.01 seconds of CPU time and the ECM algorithm took 12.375 seconds of CPU time until convergence. The data, 25×25 with 20% missing values, was simulated from the matrix-variate normal distribution with autoregressive covariances given in Section 5.1.2.

We note that the ECM and also the MCECM algorithm can lead to a substantial computational savings in terms of the rate of convergence (20). We investigate this possibility with simulated data in Figure 2 where we compare the convergence of the MCECM and the ECM approach to imputation with transposable data. The data, 25×25 with 20% of the data missing, is simulated from the matrix-variate normal with the autoregressive covariance parameters:

- Autoregressive: $\mathbf{\Sigma}_{ij} = 0.8^{|i-j|}$ and $\mathbf{\Delta}_{ij} = 0.6^{|i-j|}$.

For consistency, we used the $L_2 : L_2$ algorithm with $\rho_r = \rho_c = 1$ for both methods. From this simulation, we see that the MCECM algorithm converges in fewer iterations and in less time than the ECM algorithm. The MCECM algorithm reaches its maximum in 11 iterations, whereas the ECM algorithm needs 20 iterations.

5.1.3 Computational Strategies

We have presented our imputation algorithm for transposable data, TRCMimpute, but have not yet discussed the computations required. Calculation of the terms for the two E steps can be especially troublesome and thus we concentrate on these. Particularly, we need to find $\hat{\mathbf{X}} = \mathrm{E}\left(\mathbf{X} \,|X_o, \theta'\right)$, and the covariance terms, $\mathbf{C}^{(jj')} = \mathrm{Cov}\left(X_{\cdot j}, X_{\cdot j'}|X_o, \theta'\right)$ and $\mathbf{D}^{(ii')} = \mathrm{Cov}\left(X_{i\cdot}, X_{i'\cdot}|X_o, \theta'\right)$. The simplest but not always the most efficient way to compute these is to use the multivariate normal conditional formulas with the Kronecker covariance matrix $\mathbf{\Omega}$, i.e. if we let m be the indices of the missing values of $\mathrm{vec}(\mathbf{X})$ and o be the observed,

$$\mathrm{vec}(\hat{\mathbf{X}})_k = \begin{cases} \mathrm{vec}(\mathbf{M})_k + \mathbf{\Omega}_{ko}\mathbf{\Omega}_{oo}^{-1}\left(\mathrm{vec}(\mathbf{X})_o - \mathrm{vec}(\mathbf{M})_o\right) & \text{if } k \in m \\ \mathrm{vec}(X)_k & \text{if } k \notin o \end{cases} \tag{11}$$

and, the non-zero elements of $\mathbf{C}$ and $\mathbf{D}$ corresponding to covariances between pairs of missing values come from

$$\mathrm{Cov}\left(\mathrm{vec}(\mathbf{X})_m, \mathrm{vec}(\mathbf{X})_m|X_o, \theta'\right) = \mathbf{\Omega}_{mm} - \mathbf{\Omega}_{mo}\mathbf{\Omega}_{oo}^{-1}\mathbf{\Omega}_{om}. \tag{12}$$

This computational strategy may be appropriate for small data matrices, but even when n and p are medium-sized, this approach can be computationally expensive. Inverting $\mathbf{\Omega}$ can be of order $O(n^3p^3)$, depending on the amount of missing data. So, even if we have a relatively small matrix of dimension 100×50, this inversion costs around $O(10^{10})$! Using Gibbs sampling to approximate the calculations of the E steps in either a Stochastic or Stochastic Approximation EM-type algorithm (5) is one computational approach (we present Gibbs sampling as part of our Bayesian one-step approximation in Section 5.2.4). A stochastic approach, however, is still computationally expensive and thus, an approximation to our MCECM algorithm is needed.

5.2 One-Step Approximation to TRCMimpute

For high-dimensional transposable data, the imputation algorithm, TRCMimpute, can be computationally prohibitive. Thus, we propose a one-step approximation and a Bayesian one-step approximation which have computational costs comparable to multivariate imputation methods. We begin in Section 5.2.1, by presenting the one-step approximation algorithm and discussing its structure. The computational savings of this algorithm are due to new methods for calculating conditional distributions with the matrix-variate normal, presented in Section 5.2.2. In Section 5.2.3, we present some numerical examples supporting the choice of a one-step approximation for imputation. Finally, we conclude with the Bayesian one-step algorithm using Gibbs sampling in Section 5.2.4.

5.2.1 One-Step Algorithm (TRCMAimpute)

The MCECM algorithm for imputation with transposable regularized covariance models iterates between the E step, taking conditional expectations, and the CM steps, maximizing with respect to the covariances. Both of these steps are computationally intensive for high-dimensional data. While each iterate increases the observed log-likelihood, the first step usually produces the steepest increase in the objective. Thus, we propose an algorithm that instead of iterating between E and CM steps, approximates the solution of the MCECM algorithm by stopping after only one step.

Many have noted in other iterative maximum likelihood-type algorithms that a one-step algorithm from a good initial starting point often produces an efficient, if not comparable, approximation to the fully-iterated solution (18; 9). Thus, for our one-step approximation we seek a good initial solution from which to start our CM and E steps. For this, we turn to the multivariate regularized covariance models. Recall that all marginals of the mean-restricted matrix-variate normal are multivariate normal, and hence, if one of the penalty parameters for the TRCM model is infinitely large, we get back the RCM solution (i.e. if $\rho_r = \infty$, we get the RCM solution with penalized covariance among the columns). We propose to use the estimates from the two marginal distributions with penalized row covariances and penalized column covariances to obtain our initial starting point. This is similar to the COSSO one-step algorithm which uses a marginal solution as a good initial starting point, (18).

Since the final goal of our approximation algorithm is missing value imputation, and not parameter estimation, we then tailor our one-step algorithm to favor imputation. First, instead of using the marginal RCM covariance estimates as starting values for the subsequent TRCMimpute E and CM steps, we use the marginal estimates to obtain two sets of imputed missing values through applying the RCMimpute method to the rows and then the columns. We then average the two sets of missing value estimates and fix these to find the maximum likelihood parameter estimates for the TRCM model, competing the maximization step. Our initial estimates are then given by applying an EM-type method to the marginal models. Biernacki *et al.* (3) similarly use other EM-type algorithms to find good initial starting values for their EM mixture model algorithm. The final step of our algorithm is the Expectation step where we take the conditional expectation of the missing values given the observed values and the TRCM estimates. Note that the E step of the MCECM algorithm includes both an imputation part and a covariance correction part (see Proposition 4. For our one-step algorithm, however, the covariance correction part is unnecessary since our final goal is missing value imputation. We give the one-step approximation, called TRCMAimpute, in Algorithm 3.

Before discussing the calculations necessary in the final step of the algorithm, we pause to note a major advantage of our one-step method. If the sets of missing values from the marginal models using RCMimpute are saved in the first step, then TRCMAimpute can give three sets of missing value estimates. Since it is often unknown whether a given dataset may have independent rows or columns, cross validation, for example, can be used to determine whether penalizing the covariances of the rows, columns, or both is best for missing value imputation. This is discussed in detail in Section 6.

5.2.2 Conditional Expectations

We now discuss the final conditional expectation step of our one-step approximation algorithm. Recall that the conditional expectation can be computed via (11), but this requires inverting $\mathbf{\Omega}$ and is therefore avoided. Instead, we exploit a property of the mean-restricted matrix-variate normal, namely that all marginals of our model are multivariate normal. This

Algorithm 3 One-step algorithm approximating TRCMimpute (TRCMAimpute)

1. Initial imputation:
   (a) Impute missing values with RCMimpute assuming $\mathbf{\Sigma} = \mathbf{I}$.
   (b) Impute missing values with RCMimpute assuming $\mathbf{\Delta} = \mathbf{I}$.
   (c) Average the two estimates.
2. Find the MLE's of the transposable regularized covariance model, $\hat{\nu}$, $\hat{\mu}$, $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Delta}}$ with the imputed missing values fixed.
3. Set the missing values to their conditional expectation given these parameters: $\hat{\mathbf{X}}_m = \mathrm{E}\left(\mathbf{X}_m \,|\, \mathbf{X}_o, \hat{\nu}, \hat{\mu}, \hat{\mathbf{\Sigma}}, \hat{\mathbf{\Delta}}\right)$.

allows us to find the conditional distributions in a two step process given by Theorem 2.

Theorem 2. *Let* $\mathbf{X} \sim N_{n,p}(\nu, \mu, \mathbf{\Sigma}, \mathbf{\Delta})$, $\mathbf{M} = \nu \mathbf{1}^T + \mu^T \mathbf{1}$ *and partition* $\mathbf{X}$, $\mathbf{M}$, $\mathbf{\Sigma}$, $\mathbf{\Delta}$ *as*

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_{i,m_i} & \mathbf{X}_{i,o_i} \\ \mathbf{X}_{k,m_i} & \mathbf{X}_{k,o_i} \end{pmatrix} = \begin{pmatrix} \mathbf{X}_{m_j,j} & \mathbf{X}_{m_j,l} \\ \mathbf{X}_{o_j,j} & \mathbf{X}_{o_j,l} \end{pmatrix}, \quad \mathbf{M} = \begin{pmatrix} \mathbf{M}_{i,\cdot} \\ \mathbf{M}_{k,\cdot} \end{pmatrix} = \begin{pmatrix} \mathbf{M}_{\cdot,j} & \mathbf{M}_{\cdot,l} \end{pmatrix},$$
$$\mathbf{\Sigma} = \begin{pmatrix} \mathbf{\Sigma}_{i,i} & \mathbf{\Sigma}_{i,k} \\ \mathbf{\Sigma}_{k,i} & \mathbf{\Sigma}_{k,k} \end{pmatrix}, \quad \& \quad \mathbf{\Delta} = \begin{pmatrix} \mathbf{\Delta}_{j,j} & \mathbf{\Delta}_{j,l} \\ \mathbf{\Delta}_{l,j} & \mathbf{\Delta}_{l,l} \end{pmatrix},$$

where i *and* j *denote indices of a row and column respectively,* k *and* l *are vectors of indices of length* $n-1$ *and* $p-1$ *respectively, and* m_i *and* o_i *denote vectors of indices within row* i *and* m_j *and* o_j *indices within column* j.
Define

$$\psi = \mathbf{M}_{i,\cdot} + \mathbf{\Sigma}_{i,k}\, \mathbf{\Sigma}_{k,k}^{-1} \left(\mathbf{X}_{k,\cdot} - \mathbf{M}_{k,\cdot}\right), \quad \eta = \mathbf{M}_{\cdot,j} + \left(\mathbf{X}_{\cdot,l} - \mathbf{M}_{\cdot,l}\right) \mathbf{\Delta}_{l,l}^{-1}\, \mathbf{\Delta}_{l,j},$$
$$\mathbf{\Gamma} = \left[\mathbf{\Sigma}_{i,i} - \mathbf{\Sigma}_{i,k}\, \mathbf{\Sigma}_{k,k}^{-1}\, \mathbf{\Sigma}_{k,i}\right] \otimes \mathbf{\Delta}, \quad \& \quad \mathbf{\Phi} = \mathbf{\Sigma} \otimes \left[\mathbf{\Delta}_{j,j} - \mathbf{\Delta}_{j,l}\, \mathbf{\Delta}^{-1}{}_{l,l}\, \mathbf{\Delta}_{l,j}\right].$$

Partition ψ, η, $\mathbf{\Gamma}$ *and* $\mathbf{\Phi}$ *as* $\psi = \begin{pmatrix} \psi_{m_i} \\ \psi_{o_i} \end{pmatrix}$, $\eta = \begin{pmatrix} \eta_{m_j} & \eta_{o_j} \end{pmatrix}$,

$$\mathbf{\Gamma} = \begin{pmatrix} \mathbf{\Gamma}_{m_i,m_i} & \mathbf{\Gamma}_{m_i,o_i} \\ \mathbf{\Gamma}_{o_i,m_i} & \mathbf{\Gamma}_{o_i,o_i} \end{pmatrix}, \quad \& \quad \mathbf{\Phi} = \begin{pmatrix} \mathbf{\Phi}_{m_j,m_j} & \mathbf{\Phi}_{m_j,o_j} \\ \mathbf{\Phi}_{o_j,m_j} & \mathbf{\Phi}_{o_j,o_j} \end{pmatrix}.$$

Then,

(a) $(\mathbf{X}_{i,m_i} \,|\, \mathbf{X}_{i,o_i}, \mathbf{X}_{k,\cdot})$
$\sim N\left(\psi_{m_i} + \mathbf{\Gamma}_{m_i,o_i} \mathbf{\Gamma}_{o_i,o_i}^{-1} \left(\mathbf{X}_{i,o_i} - \psi_{o_i}\right), \mathbf{\Gamma}_{m_i,m_i} - \mathbf{\Gamma}_{m_i,o_i} \mathbf{\Gamma}_{o_i,o_i}^{-1} \mathbf{\Gamma}_{o_i,m_i}\right)$.

(b) $\left(\mathbf{X}_{m_j,j} \,|\, \mathbf{X}_{o_j,j}, \mathbf{X}_{\cdot,l}\right)$
$\sim N\left(\eta_{m_j} + \mathbf{\Phi}_{m_j,o_j} \mathbf{\Phi}_{o_j,o_j}^{-1} \left(\mathbf{X}_{o_j,j} - \eta_{o_j}\right), \mathbf{\Phi}_{m_j,m_j} - \mathbf{\Phi}_{m_j,o_j} \mathbf{\Phi}_{o_j,o_j}^{-1} \mathbf{\Phi}_{o_j,m_j}\right)$.

Proof. See Appendix A.

Thus, from Theorem 2, the conditional distribution of values in a row or column given the rest of the matrix can be calculated in a two step process where each step takes at most the number of computations as required for calculating multivariate conditional distributions. The first step finds the distribution of an entire row or column conditional on the rest of

the matrix, and the second step finds the conditional distribution of the values of interest within the row or column. By splitting the calculations in this manner, we avoid inverting the $np \times np$ Kronecker product covariance. This alternative form for the conditional distributions of elements in a row or column leads to an iterative algorithm for calculating the conditional expectation of the missing values given the observed values. We call this the Alternating Conditional Expectations Algorithm, given in Algorithm 4.

Algorithm 4 Alternating Conditional Expectations Algorithm.

1. Initialize $\hat{\mathbf{X}}_{i,j}^{(0)} = \hat{\nu}_i + \hat{\mu}_j$ for $\mathbf{X}_{i,j} \in \mathbf{X}_m$.
2. For each row, i, with missing values:
   (a) Set $\hat{\mathbf{X}}_{i,m_i}^{(k+1)} = \mathrm{E}\left(\mathbf{X}_{i,m_i} \,|\, \mathbf{X}_{i,o_i}^{(k)}, \mathbf{X}_{\neq i,\cdot}^{(k)}\right)$.
3. For each column, j, with missing values:
   (a) Set $\hat{\mathbf{X}}_{m_j,j}^{(k+1)} = \mathrm{E}\left(\mathbf{X}_{m_j,j} \,|\, \mathbf{X}_{o_j,j}^{(k)}, \mathbf{X}_{\cdot,\neq j}^{(k)}\right)$.
4. Repeat Steps 1 and 2 until convergence.

Theorem 3. *Let* $\mathbf{X} \sim N_{n,p}(\nu, \mu, \mathbf{\Sigma}, \mathbf{\Delta})$ *and partition* $\mathrm{vec}(\mathbf{X}) = (\mathrm{vec}(\mathbf{X}_m) \;\; \mathrm{vec}(\mathbf{X}_o))$ *where* m *and* o *are indices partitioned by rows,* (m_i *and* o_i) *and columns,* (m_j *and* o_j), *so that a row,* $\mathbf{X}_{i,\cdot} = (\mathbf{X}_{i,m_i} \;\; \mathbf{X}_{i,o_i})$ *and a column* $\mathbf{X}_{\cdot,j} = \begin{pmatrix} \mathbf{X}_{m_j,j} \\ \mathbf{X}_{o_j,j} \end{pmatrix}$. *Then, the Alternating Conditional Expectations Algorithm, Algorithm 4, converges to* $\mathrm{E}(\mathbf{X}_m \,|\, \mathbf{X}_o)$.

Proof. See Appendix A.

Theorem 3 shows that the conditional expectations needed in step 3 of the one-step approximation algorithm can be calculated in an iterative manner from the conditional distributions of elements in a row and column, as in Algorithm 4. Thus, the two Theorems, 2 and 3, show that the conditional expectations can be calculated by separately inverting the row and column covariance matrices, instead of the overall Kronecker product covariance. This reduces the order of operations from around $O(n^3p^3)$ to $O(n^3 + p^3)$, a substantial savings. We also note that the structure of the Alternating Conditional Expectations Algorithm often leads to a faster rate of convergence as discussed in the proof of Theorem 3. In practice, we have found that the algorithm converges in a small number of iterations. For high-dimensional data, these two results mean that matrix-variate models can be used in any application where multivariate models are computationally feasible, thus opening the door to applications of transposable models!

5.2.3 Numerical Comparisons

We now investigate the accuracy of the one-step approximation algorithm in terms of observed log-likelihood and imputation accuracy with a numerical example. Here, we simulate fifty datasets, 25×25, from the matrix-variate normal model with autoregressive covariance matrices.

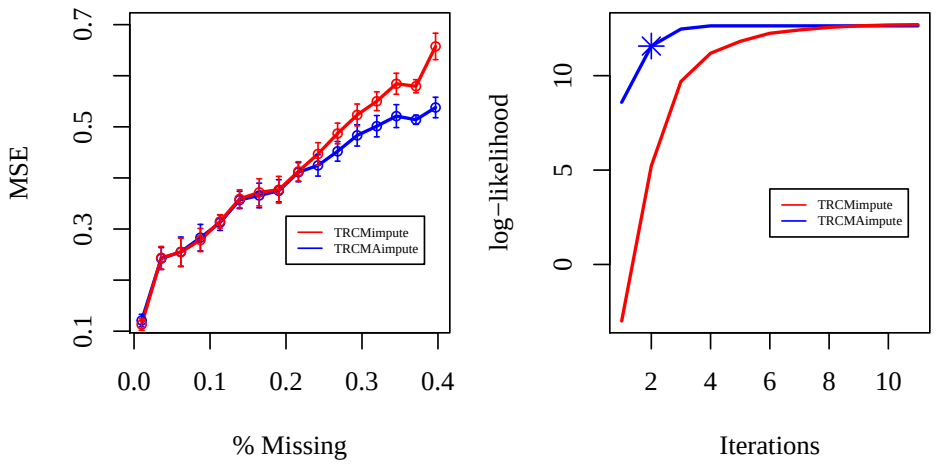

Figure 3: Comparison of the mean MSE with standard errors (left) of the MCECM imputation algorithm (TRCMimpute) and the one-step approximation (TRCMAimpute) for transposable data of dimension 25×25 with various percentages of missing data. Fifty datasets were simulated from the matrix-variate normal distribution with autoregressive covariances as given in Section 5.2. Observed log-likelihood (right) verses iterations for TRCMimpute and TRCMAimpute with 25% missing values. The one-step approximation begins at the TRCM parameter estimates using the imputed values from RCMimpute. The observed log-likelihood of one-step approximation is given by a star after the first step. All methods use L_2 penalties with $\rho_r = \rho_c = 1$ for comparison purposes.

- Autoregressive: $\mathbf{\Sigma}_{ij} = 0.8^{|i-j|}$ and $\mathbf{\Delta}_{ij} = 0.6^{|i-j|}$.

We delete values at random according to certain percentages and report the mean MSE for both the MCECM algorithm, TRCMimpute, and the one-step approximation, TRCMAimpute on the right in Figure 3. The one-step approximation performs comparably, or slightly better, in terms of imputation error to the MCECM algorithm for all percentages of missing values. We note that TRCMAimpute could give better missing value estimates if the MCECM algorithm converges to a sub-optimal stationary point of the observed log-likelihood. For a dataset with 25% missing values, we apply the MCECM algorithm and also apply our approximation extended beyond the first step, but denote the observed log-likelihood after the first step with a star on the right in the above figure. This shows that using marginals to provide a good starting value, does indeed begin the algorithm at a higher observed log-likelihood. Also, after the first step, the observed log-likelihood is very close to the fully-iterated maximum. Thus, the one-step approximation appears to be a comparable approximation to the TRCMimpute approximation which is feasible for use with high-dimensional datasets.

5.2.4 Bayesian One-Step Approximation

In discussing the computations involved in the Expectation step of the MCECM algorithm for imputation, we mentioned the possibility of extending the algorithm to a stochastic or stochastic approximation ECM-type algorithm, Section 5.1.3. Here, we also note that the one-step approximation can be formulated as a stochastic one-step algorithm using Gibbs

sampling. First, we present a blocked Gibbs sampler, Algorithm 5, for stochastically generating missing values from their posterior distribution, and then we discuss how this can be used in a Bayesian-type one-step approximation.

Algorithm 5 Blocked Gibbs Sampler for estimating E step calculations in TRCMimpute or the final step of the Bayesian one-step approximation.

1. For each row, i, with missing values:
   (a) Generate $\mathbf{X}_{i,m_i}^{(k+1)}$ from $g\left(\mathbf{X}_{i,m_i} \,|\, \mathbf{X}_{i,o_i}^{(k)}, \mathbf{X}_{\neq i,\cdot}^{(k)}\right)$, given in Theorem 2 (a).
2. For each column, j, with missing values:
   (a) Generate $\mathbf{X}_{m_j,j}^{(k+1)}$ from $g\left(\mathbf{X}_{m_j,j} \,|\, \mathbf{X}_{o_j,j}^{(k)}, \mathbf{X}_{\cdot,\neq j}^{(k)}\right)$, given in Theorem 2 (b).
3. Repeat Steps 1 and 2 until a stationary distribution is reached.

The blocked Gibbs sampler generates all missing values in a row or a column as a group from their conditional distributions given in Theorem 2. Thus, a deterministic overlapping blocking scheme is used to update the elements that can lead to faster convergence (24). This algorithm converges to the stationary distribution of the missing values given the observed values, and thus can be used in place of the Alternating Conditional Expectation Algorithm in the final step of the approximation. We call this the Bayesian one-step approximation. The conditional distribution of the missing values can also be thought of as the posterior distribution from which repeated draws can form a set of repeated imputations in the multiple imputation framework (26).

6 Results and Simulations

The following results indicate that imputation with transposable regularized covariance models is useful in a variety of situations and data types often giving much better error rates than existing methods. We first assess the performance of our one-step approximation, TRCMAimpute, under a variety of simulations with both full and sparse covariance matrices. Then, we apply our methods to microarray data and a subset of the Netflix movie rating data. We compare performances to three commonly used single imputation methods - SVD methods (SVDimpute), k-nearest neighbors (KNNimpute), and local least squares (LLSimpute) (29), (16).

With real data, the TRCM penalty parameters ρ_r and ρ_c must be estimated from the data. To this end, we use 5-fold cross validation in all simulations and examples. This is accomplished by randomly deleting 20% of the observed values in each fold, applying an imputation method, and then measuring the imputation error on the deleted values. When selecting penalty parameters, we always include the possibility of an infinite value giving a marginal, multivariate model obtained in the first step of our approximation method. Thus, for application of TRCMAimpute, a model penalizing only the rows, only the columns, or both may be chosen by the cross-validation procedure. We remind the reader that the four penalty types of the TRCM model are referred to as $L_{q_r} : L_{q_c}$, or the penalty-type placed on the rows and then the penalty-type on the columns.

6.1 Simulations

We test our imputation method for transposable data under a variety of simulated distributions, both multivariate and matrix-variate. All simulations use one of four covariance types given below. These are numbered as they appear in the simulation tables.

1. Autoregressive: $\mathbf{\Sigma}_{ij} = 0.8^{|i-j|}$ and $\mathbf{\Delta}_{ij} = 0.6^{|i-j|}$.
2. Equal off-diagonals: $\mathbf{\Sigma}_{ij} = 0.5$ and $\mathbf{\Delta}_{ij} = 0.5$ for $i \neq j$, and $\mathbf{\Sigma}_{ii} = 1$ and $\mathbf{\Delta}_{ii} = 1$.
3. Blocked diagonal: $\mathbf{\Sigma}_{ii} = 1$ and $\mathbf{\Delta}_{ii} = 1$ with off-diagonal elements of 5×5 blocks of $\mathbf{\Sigma}$ are 0.8 and of $\mathbf{\Delta}$, 0.6.
4. Banded off-diagonals: $\mathbf{\Sigma}_{ii} = 1$ and $\mathbf{\Delta}_{ii} = 1$ with $\mathbf{\Sigma}_{ij} = \begin{cases} 0.8 & \text{if } |i-j| \text{ divisible by 5.} \\ 0 & \text{otherwise.} \end{cases}$, $\mathbf{\Delta}_{ij} = \begin{cases} 0.6 & \text{if } |i-j| \text{ divisible by 5.} \\ 0 & \text{otherwise.} \end{cases}$.

The first simulation, with results in Table 1, compares performances with both multivariate distributions, only $\mathbf{\Sigma}$ given, and matrix-variate distributions, both $\mathbf{\Sigma}$ and $\mathbf{\Delta}$ given. In all simulations, the data is of dimension 50×50 with either 25% or 75% of the values missing at random. The second simulation, given in Table 2, gives results for matrix-variate distributions with one dimension much larger than the other, 100×10 and 10% of values missing at random. The final simulation, in Table 3, tests the performance of our method when the data has a transposable covariance structure, but is not normally distributed. Here, the data, of dimension 50×50, is either distributed Chi-square with three degrees of freedom or Poisson with mean three. The Chi-square and Poisson distributions introduce large outliers and the Poisson distribution is discrete. The data has 25% missing values deleted at random. All three sets of simulations are compared to SVD imputation and k-nearest neighbor imputation.

These three sets of simulations show that TRCMAimpute is competitive with the two of the most commonly used single imputation methods, SVD and k-nearest neighbor imputation. First, TRCM with L_2 penalties outperforms the other possible TRCM penalty types. This may be due to the fact that the covariance estimates with L_2 penalties has a globally unique solution, Theorem 1, while the estimation procedure for other penalty types only reaches a stationary point, Proposition 2. The one-step approximation permits the flexibility to choose either multivariate or transposable models. As seen in the first simulation, Table 1, with smaller percentages of missing values, cross-validation generally chooses the correct model. However, with 75% of the values missing, the transposable model is chosen even if the underlying distribution is multivariate. The additional structure of the TRCM covariances may allow for more information to be gleaned from the few observed values, perhaps explaining the better performance of the matrix-variate model. TRCMAimpute seems to perform best in comparison to SVD and k-nearest neighbor imputation for the full covariances with equal off-diagonal elements. Our TRCM-based imputation methods appear particularly robust to departures from normality and outliers compared to existing methods, as shown in Table 3. The flexibility of the method to adapt to various covariance structures could account for this. Overall, imputation methods based on transposable covariance models compare favorably in simulations.

6.2 Examples & Comparisons

In this section, we use two data types to test transposable regularized covariance model imputation methods and compare the results to existing methods. Authors have suggested

	TRCMAimpute			Others	
	$L_1 : L_1$	$L_1 : L_2$	$L_2 : L_2$	SVD	KNN
$\mathbf{\Sigma}_1$	0.8936 (0.01) 45/50	0.725 (0.0069) 0/50	**0.5919** (0.0056) 50/50	0.634 (0.0081)	**0.448** (0.005)
$\mathbf{\Sigma}_1$, $\mathbf{\Delta}_1$	0.8255 (0.012) 0/50	0.6315 (0.0078) 0/50	**0.5402** (0.0067) 0/50	**0.4603** (0.0083)	0.8034 (0.016)
$\mathbf{\Sigma}_2$	0.895 (0.016) 43/50	**0.7829** (0.013) 0/50	**0.6392** (0.008) 48/50	0.993 (0.019)	0.9498 (0.017)
$\mathbf{\Sigma}_2$, $\mathbf{\Delta}_2$	0.749 (0.044) 0/50	0.6867 (0.034) 0/50	**0.4556** (0.0098) 48/50	**0.6821** (0.051)	0.8273 (0.055)
$\mathbf{\Sigma}_3$	1.04 (0.017) 37/50	1.02 (0.017) 9/50	0.9348 (0.016) 49/50	**0.7384** (0.012)	**0.9115** (0.014)
$\mathbf{\Sigma}_3$, $\mathbf{\Delta}_3$	1.012 (0.02) 5/50	0.9477 (0.019) 0/50	**0.8585** (0.017) 37/50	**0.7271** (0.016)	0.9886 (0.019)
$\mathbf{\Sigma}_4$	0.9986 (0.018) 37/50	0.9407 (0.017) 3/50	**0.8067** (0.014) 48/50	**0.4903** (0.0076)	0.9057 (0.014)
$\mathbf{\Sigma}_4$, $\mathbf{\Delta}_4$	0.9726 (0.033) 6/50	0.855 (0.028) 0/50	**0.6999** (0.022) 39/50	**0.5282** (0.024)	0.9366 (0.031)
$\mathbf{\Sigma}_1$	0.9134 (0.0096) 21/50	**0.9083** (0.0092) 18/50	**0.8948** (0.009) 21/50	1.173 (0.013)	0.9349 (0.0092)
$\mathbf{\Sigma}_1$, $\mathbf{\Delta}_1$	0.867 (0.011) 0/50	**0.8569** (0.01) 0/50	**0.845** (0.0096) 0/50	0.9535 (0.01)	0.9736 (0.013)
$\mathbf{\Sigma}_3$	1.053 (0.01) 7/50	1.052 (0.01) 7/50	**1.048** (0.01) 7/50	1.22 (0.013)	**1.03** (0.01)
$\mathbf{\Sigma}_3$, $\mathbf{\Delta}_3$	1.001 (0.014) 0/50	**0.9968** (0.014) 0/50	**0.9945** (0.014) 1/50	1.11 (0.016)	1.006 (0.014)

Table 1: *Mean MSE with standard error computed over 50 datasets of dimension* 50×50 *simulated under the matrix-variate normal distribution with covariances given in Section 6.1. In the upper portion of the table, 25% of values are missing and in the lower, 75% missing. The TRCM one-step approximation with* $L_1 : L_1$, $L_1 : L_2$ *and* $L_2 : L_2$ *penalties was used as well as the SVD and k-nearest neighbor imputation. Below the errors for TRCMAimpute, we give the number of simulations out of 50 in which a marginal, multivariate method (RCMimpute) was chosen over the matrix-variate method. Parameters were chosen for all methods via 5-fold cross validation. Best performing methods are given in bold.*

	TRCMAimpute		Others	
	$L_2 : L_1$	$L_2 : L_2$	SVD	KNN
$\mathbf{\Sigma}_1$, $\mathbf{\Delta}_1$	0.8227 (0.019)	0.7072 (0.016)	1.075 (0.024)	**0.6971** (0.018)
$\mathbf{\Sigma}_2$, $\mathbf{\Delta}_2$	1.019 (0.15)	**0.9441** (0.13)	1.306 (0.23)	1.057 (0.17)
$\mathbf{\Sigma}_3$, $\mathbf{\Delta}_3$	0.9372 (0.047)	**0.841** (0.042)	1.121 (0.05)	0.9241 (0.042)
$\mathbf{\Sigma}_4$, $\mathbf{\Delta}_4$	0.7044 (0.059)	**0.6148** (0.049)	0.9751 (0.074)	1.118 (0.089)

Table 2: *Mean MSE with standard errors over 50 datasets of dimension* 100×10 *with 10% missing values simulated under the matrix-variate normal with covariances given in Section 6.1. The TRCM one-step approximation with* $L_2 : L_1$ *and* $L_2 : L_2$ *penalties was used as well as the SVD and k-nearest neighbor imputation. Parameters were chosen for all methods via 5-fold cross validation. Best performing methods are given in bold.*

that microarrays and user-ratings data, such as the Netflix movie-rating data are transposable or matrix-distributed (8), (1), hence, we assess the performance of our methods on these types of datasets.

		TRCMAimpute		Others	
		$L_2 : L_1$	$L_2 : L_2$	KNN	SVD
Chi-square	$\mathbf{\Sigma}_1, \mathbf{\Delta}_1$	3.824 (0.065)	**2.611** (0.044)	6.85 (0.34)	7.684 (0.15)
	$\mathbf{\Sigma}_3, \mathbf{\Delta}_3$	5.525 (0.14)	**5.068** (0.15)	29.41 (0.83)	50.16 (0.74)
Poisson	$\mathbf{\Sigma}_1, \mathbf{\Delta}_1$	2.442 (0.05)	**1.571** (0.021)	8.04 (0.34)	5.824 (0.11)
	$\mathbf{\Sigma}_3, \mathbf{\Delta}_3$	3.045 (0.075)	**2.813** (0.081)	29.13 (0.95)	49.2 (0.68)

Table 3: *Mean MSE with standard error computed over 50 datasets of dimension* 50×50 *with 25% missing values simulated under the Chi-square distribution with 3 degrees of freedom or the Poisson distribution with mean 3 with Kronecker product covariance structure given by the covariances in Section 6.1. The TRCM one-step approximation with* $L_2 : L_1$ *and* $L_2 : L_2$ *penalties was used as well as the SVD and k-nearest neighbor imputation. Parameters were chosen for all methods via 5-fold cross validation. Best performing methods are given in bold.*

6.2.1 Microarray Data

For our analysis, we use a microarray dataset of kidney cancer tumor samples (31). The dataset contains 14,814 genes and 178 samples. About 10% of the data is missing. For the following figures, all of the genes with no missing values were taken, totaling 1,031 genes. Missing values were then placed at random. Errors were assessed by comparing the imputed values to the true observed values.

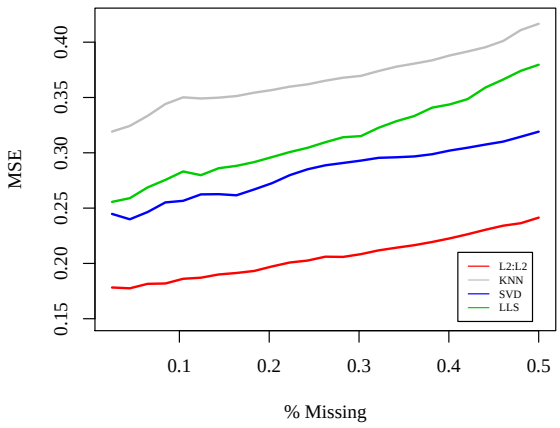

Figure 4: Comparison of MSE for imputation methods on kidney cancer microarray data with different percentages of missing values. Genes in which all samples are observed are taken with values deleted at random. TRCMAimpute, $L_2 : L_2$ and common imputation methods KNNimpute, SVDimpute and LLSimpute are compared with all parameters chosen by 5-fold cross-validation. Cross-validation chose to penalize only the arrays for the one-step approximation algorithm, TRCMAimpute.

We assess the performance of TRCM imputation methods on this microarray data and compare them to existing methods for various percentages of missing values, deleted at random, in Figure 4. Here, we use L_2 penalties since these are computationally less expensive for high-dimensional data. TRCMAimpute outperforms competing methods in terms of im-

putation error for all percentages of missing values. We note that cross-validation exclusively choose the marginal, multivariate model from the one-step approximation. This indicates that the arrays in this microarray dataset may indeed by independent.

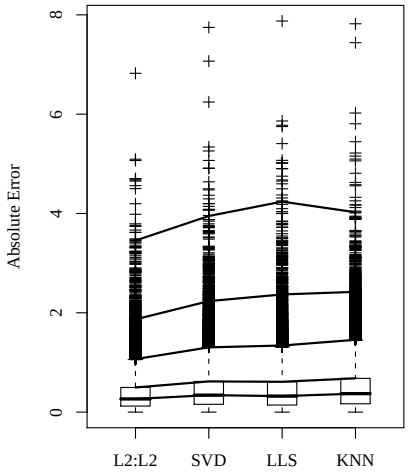

Figure 5: Boxplots of individual absolute errors for various imputation methods on kidney cancer microarray data. Genes in which all samples are observed were taken and deleted in the same pattern as a random gene in the original dataset. Lines are drawn at the 50%, 75%, 95%, 99%, and 99.9% quantiles. TRCMAimpute, $L_2 : L_2$ has a mean absolute error of 0.37 and has lower errors at every quantile than its closest competitor, SVDimpute which has a mean absolute error of 0.46.

In Figure 4, we deleted values at random, but missing values in microarray datasets are often not missing randomly. Also, one is often interested in not only the error in terms of MSE, but the individual errors made as well. To investigate these issues, we assess individual absolute errors of data that is missing in the same pattern as the original data. For each complete gene, values were set to missing in the same samples as a randomly sampled gene from the original dataset. Figure 5 displays the boxplots of the absolute imputation errors. Lines are drawn at quantiles to assess the relative performances of each method. Here, TRCMAimpute has lower absolute errors at each quantile. Also, the set of imputed values has far fewer outliers than competing methods. The mean absolute error for TRCMAimpute is 0.37, far below the next two methods, LLSimpute and SVDimpute which have a mean absolute error of 0.46. Altogether, our results illustrate the utility and flexibility of using TRCMs for missing value imputation in microarray data.

6.2.2 Netflix Data

We compare transposable regularized covariance models and existing methods on the Netflix movie rating data (2). This dataset is extremely high-dimensional, with over 480,000 customers and over 17,000 movies, and is very sparse, with over 98% of the ratings missing. Hence, assessing the utility of our methods from this data as a whole is not currently feasible. Instead, we rank both the movies and the customers by the number of ratings and take as a subset, the top 250 customer's ratings of the top 250 movies. This subset has around 12% of

the ratings missing. We then delete more data at random to evaluate the performance of the methods. Figure 6 compares the performances of the TRCM methods to existing methods.

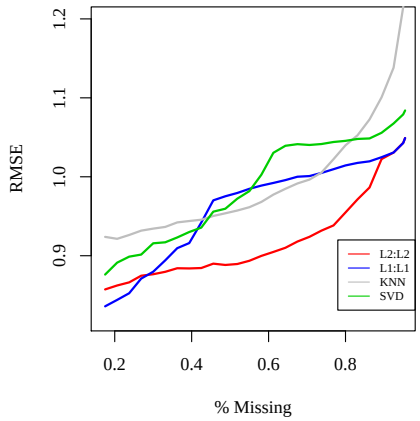

Figure 6: Comparison of the root MSE (RMSE) for a subset of the Netflix data for TRCMAimpute, $L_2 : L_2$ and $L_1 : L_1$, to KNNimpute and SVDimpute. A dense subset was obtained by ranking the movies and customers in terms of number of ratings and taking the top 250 movies and 250 customers. This subset has around 12% missing and additional values were deleted at random, up to 95%. With 95% missing, the RMSE of TRCMAimpute is 1.049 compared to 1.084 of the SVD and 1.354 using the movie averages.

Before discussing these results, we first make a note about the comparability of our errors rates to those for the Netflix Prize. Because we chose the subset of data based on the rank of the number of observed ratings, we can expect the RMSE to be higher here than applying these methods to the full dataset. This method of obtaining a subset leaves out potentially thousands of highly correlated customers or movies that would greatly increase a method's predictive ability. In fact, the RMSE of the SVD method on the entire Netflix data is 0.91 (27), much less than the observed RMSE of 1.084 on our subset. Thus, we can conjecture, that all of the methods we present would do better in terms of RMSE using the entire dataset than the small subset on which we present results.

The results indicate that TRCM imputation methods, particularly with L_2 penalties, are competitive with existing methods. At higher percentages of missing values, our methods perform notably well. With 95% missing values in our subset, TRCMAimpute has an RMSE of 1.049 compared to the SVD at 1.084 and using simply the movie averages with an error of 1.354. This is of potentially great interest for missing data imputation on a larger scale where the percentage of missing data is greater than 98%. We note that at smaller percentages of missing values, marginal models penalizing the movies were often chosen by cross-validation, indicating that the movies may have more predictive power. Since leading imputation methods for the Netflix Prize, are ensembles of many different methods (1), we do not believe that TRCM methods alone would outperform ensemble methods. If, however, our methods outperform other individual methods, they could prove to be beneficial additions to imputation ensembles.

7 Discussion

We have extended regularized covariance models to the transposable or matrix-variate framework and have applied these models to estimation of missing values in high-dimensional data. In the process, we have proposed a variation on the matrix-variate normal, the mean-restricted matrix-variate normal distribution to give separable parameters for the means of the rows and the columns. Also, the separate penalties we have placed on the row and column covariance matrices allow for non-singular maximum likelihood covariance estimation. This along with the efficient computational methods for determining conditional distributions in the matrix-variate framework permit the use of transposable methods in applications such as imputation. We have developed penalized EM-type algorithms maximizing the observed penalized log-likelihood for use in missing data imputation in both the multivariate and matrix-variate frameworks. We also have proposed a one-step approximation algorithm that is computationally feasible in high-dimensional data. We have given results on both simulated and real data, displaying the utility of our imputation methods, which often outpreform existing methods.

Our imputation methods are restricted to datasets where for each pair of rows, there is at least one column in which both entries are observed and vice versa for each pair of columns. A major drawback of TRCM imputation methods is computational cost. First, RCMimpute using the columns as features costs $O(2kp^3)$, where k is the total number of iterations. This is roughly on the order of other common imputation methods such as the SVDimpute which costs $O(knp^2)$. Our one-step approximation, TRCMAimpute, with RCMimpute as a subroutine, costs $O((2k + mn + 1)p^3 + mpn^3)$, where m is the total number of iterations of the Alternating Conditional Expectations Algorithm.

The main application of this paper has been to missing value imputation. We note that this is separate from the matrix-completion methods via convex optimization of Candes and Recht (4), which focuses on matrix-reconstruction instead of imputation. Also, we have presented a single imputation procedure, but our techniques can easily be extended to incorporate multiple imputation approaches such as repeated imputations via samples from the posterior distribution in Bayesian settings (26), as presented in the Bayesian one-step approximation. The blocked Gibbs sampler we present can also be used in place of Expectation step in our MCECM algorithm to form a stochastic or stochastic approximation in the Bayesian framework. In addition, we have not discussed ultimate use or analysis of the imputed data, which will often dictate the imputation approach. Our imputation methods form a foundation and can be extended to further address these issues.

Transposable regularized covariance models may be of potential mathematical and practical interest in numerous fields. TRCMs allow for non-singular estimation of the covariances of the rows and columns, which is essential for any application. Adding restrictions to the mean of the TRCM, allows one to estimate all parameters from a single observed data matrix. Also, introduction of efficient methods of calculating conditional distributions and expectations make this model computationally feasible for many applications. Hence, transposable regularized covariance models have many potential future uses in areas such as hypothesis testing, classification and prediction, and data mining.

A Proofs

Proof of Proposition 1. Expanding the trace term of $\ell(\mathbf{M}, \mathbf{\Sigma}, \mathbf{\Delta})$ in terms of μ and ν and then taking partial derivatives, we get

$$\begin{aligned}\frac{\partial \ell}{\partial \nu} &= 2\,\mathbf{\Sigma}^{-1}\,\nu \mathbf{1}^T\,\mathbf{\Delta}^{-1} - 2\,\mathbf{\Sigma}^{-1}(\mathbf{X} - \mathbf{1}\mu^T)\,\mathbf{\Delta}^{-1} = 0 \\ &\Rightarrow \hat{\nu}\mathbf{1}^T = \mathbf{X} - \mathbf{1}\mu^T \\ &\Rightarrow \hat{\nu} = \frac{\mathbf{1}^T(\mathbf{X} - \mathbf{1}\mu^T)}{p} = \sum_{j=1}^{p} \frac{X_{\cdot j} - \mu_j}{p}\end{aligned}$$

A similar argument gives $\hat{\mu}$. □

Proof of Propoisition 2. Notice that the first three terms of $\ell(\mathbf{\Sigma}, \mathbf{\Delta})$ are differentiable and $\ell(\mathbf{\Sigma}, \mathbf{\Delta})$ is continuous. Then, since $\ell(\mathbf{\Sigma}, \mathbf{\Delta})$ is strictly concave in $\mathbf{\Sigma}^{-1}$ with $\mathbf{\Delta}^{-1}$ fixed and in $\mathbf{\Delta}^{-1}$ with $\mathbf{\Sigma}^{-1}$ fixed, maximization with respect to each gradient gives a unique coordinate-wise maximum. Thus, we have satisfied the conditions of Tseng, Theorem 4.1 (c) (28), and block coordinate-wise maximization converges to a stationary point of $\ell(\mathbf{\Sigma}, \mathbf{\Delta})$. □

Proof of Theorem 1. Our proof that the solution eq6 maximizes the penalized log-likelihood with L_2 penalties begins with the coordinate-wise gradients, (5), and shows that there is only one solution to these equations and it is thus the globally optimal solution. Throughout this proof we assume $\mathbf{X}$ is centered and let $\mathbf{X} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^{\mathrm{T}}$ be the SVD of $\mathbf{X}$ and $d = \mathrm{diag}(\mathbf{D})$.

First, we can take the eigenvectors of $\mathbf{\Sigma}^*$ and $\mathbf{\Delta}^*$ to be the left and right singular vectors of $\mathbf{X}$ respectively, because rearranging the gradients, (5), gives

$$\begin{aligned}p\,\mathbf{\Sigma} - 4\rho_r\,\mathbf{\Sigma}^{-1} &= \mathbf{X}\,\mathbf{\Delta}^{-1}\,\mathbf{X}^{\mathrm{T}} \\ n\,\mathbf{\Delta} - 4\rho_c\,\mathbf{\Delta}^{-1} &= \mathbf{X}^{\mathrm{T}}\,\mathbf{\Sigma}^{-1}\,\mathbf{X}\,.\end{aligned}$$

Thus, the eigenvectors of $\mathbf{\Sigma}$ and $\mathbf{\Delta}$ must be equal to their respective quadratic forms. This gives only one solution for the eigenvectors which is the left and right singular vectors of $\mathbf{X}$. Note that if $\mathrm{rank}(\mathbf{X}) = r$, then the last $n - r$ eigenvectors of $\mathbf{U}$ and the last $p - r$ of $\mathbf{V}$ are not unique.

Now, given the eigenvectors of $\mathbf{\Sigma}^*$ and $\mathbf{\Delta}^*$, we can write our penalized log-likelihood function in terms of the eigenvalues β and θ and the singular values d.

$$-\ell(\beta, \theta) = \frac{p}{2}\sum_{i=1}^{n}\log\beta_i + \frac{n}{2}\sum_{j=1}^{p}\log\theta_j + \frac{1}{2}\sum_{j=1}^{p}\frac{d_j^2}{\beta_j\theta_j} + \rho_r\sum_{i=1}^{n}\frac{1}{\beta_i^2} + \rho_c\sum_{j=1}^{p}\frac{1}{\theta_j^2}. \tag{13}$$

Note that this is a biconvex function of $\frac{1}{\beta}$ and $\frac{1}{\theta}$. Hence, we can use sequential convex programming to minimize (13), but this may not converge to the global minimum, which we seek. Instead, we use the coordinate-wise gradients of (13), adding $n - p$ zeros to θ and d so they are of length n. They are

$$\begin{aligned}p\theta_i\beta_i^2 - d_i^2\beta_i - 4\rho_r\theta_i = 0 \qquad &\text{for } i = 1\ldots n. \\ n\beta_i\theta_i^2 - d_i^2\theta_i - 4\rho_c\beta_i = 0 \qquad &\text{for } i = 1\ldots n.\end{aligned} \tag{14}$$

Note that the global minimum must satisfy (14). The last $n - r$ values of β^* are immediately known and equal to $2\sqrt{\frac{\rho_r}{p}}$. Also, if $p > r$, then the last $p - r$ values of θ^* are $2\sqrt{\frac{\rho_c}{n}}$. So, we concentrate on the first r values of θ and β.

Since the gradients, (14), are quadratic equations, we can write it as one equation in terms of β_i by letting $\theta_i = \frac{d_i^2\beta_i}{p\beta_i^2 - 4\rho_r}$. This gives us the following fifth degree polynomial.

$$\beta_i\left(c_1\beta_i^4 + c_2\beta_i^2 + c_3\right) = 0 \tag{15}$$

for each eigenvalue indexed by i with coefficients

$$c_1^{(i)} = -4\rho_c p^2, \quad c_2^{(i)} = 32\rho_r\rho_c p + d_i^4(n - p), \quad \& \quad c_3^{(i)} = 4\rho_r(d_i^4 - 16\rho_r\rho_c).$$

The five solutions to (15) are 0 and $\pm\sqrt{\frac{-c_2^{(i)} \pm \sqrt{{c_2^{(i)}}^2 - 4c_1^{(i)}c_3^{(i)}}}{2c_1^{(i)}}}$. But, we are looking for solutions that are real and positive in terms of both β_i and θ_i. Thus, we can immediately dismiss the zero solution and the two negative solutions. Now, notice that if $d_i^4 \geq 16\rho_r\rho_c$, then $c_3^{(i)} \geq 0$, and thus by Descartes' sign rule, there

exists only one positive real root. This root is given by $\beta_i = \sqrt{\frac{-c_2^{(i)} - \sqrt{{c_2^{(i)}}^2 - 4c_1^{(i)}c_3^{(i)}}}{2c_1^{(i)}}}$. We now consider the case when $d_i^4 < 16\rho_r\rho_c$ and we have two positive real roots. The above root is obviously still feasible in this case. We check the other positive root given by $\beta_i = \sqrt{\frac{-c_2^{(i)} + \sqrt{{c_2^{(i)}}^2 - 4c_1^{(i)}c_3^{(i)}}}{2c_1^{(i)}}}$ to see if the corresponding $\theta_i > 0$. The numerator of θ_i is always strictly positive, so for θ_i to be feasible, $p\beta_i^2 > 4\rho_r$. Substituting in the possible root, β_i, we see that this inequality does not hold. Hence, this root is infeasible, leaving us with only one feasible root.

Therefore, we have only one feasible solution (β_i, θ_i) for each i to the gradient equations (14). Since the global solution must satisfy these gradient equations, we conclude that the root is the unique global solution (β^*, θ^*). We comment here that using iterative coordinate descent to solve the quadratic equations (14) by taking the positive roots, converges to this globally optimal solution. This is true because any solution to the coordinate procedure must satisfy the coordinate-wise gradient conditions, which we have just proven to have only one solution. Thus, this is a rare instance when the coordinate descent solution to a biconvex problem converges to the global minimum. □

Proof of Proposition 3.

$$\begin{aligned}
\mathrm{E}_{(\mathbf{\Sigma},\mathbf{\Delta})}\left[\ell(\mathbf{\Sigma},\mathbf{\Delta}) - \ell(\hat{\mathbf{\Sigma}},\hat{\mathbf{\Delta}})\right] &= \frac{p}{2}\left(\log|\,\mathbf{\Sigma}^{-1}\,| - \log|\hat{\mathbf{\Sigma}^{-1}}|\right) + \frac{n}{2}\left(\log|\,\mathbf{\Delta}^{-1}\,| - \log|\hat{\mathbf{\Delta}^{-1}}|\right) \\
&\quad - \frac{1}{2}\mathrm{E}_{(\mathbf{\Sigma},\mathbf{\Delta})}\left[\mathrm{tr}\left(\mathbf{\Sigma}^{-1}\,\mathbf{X}\,\mathbf{\Delta}^{-1}\,\mathbf{X}^{\mathrm{T}}\right) - \mathrm{tr}\left(\hat{\mathbf{\Sigma}^{-1}}\,\mathbf{X}\,\hat{\mathbf{\Delta}}^{-1}\,\mathbf{X}^{\mathrm{T}}\right)\right] \\
&= \frac{p}{2}\log|\hat{\mathbf{\Sigma}}\,\mathbf{\Sigma}^{-1}\,| + \frac{n}{2}\log|\hat{\mathbf{\Delta}}\,\mathbf{\Delta}^{-1}\,| \\
&\quad - \frac{1}{2}\left[\mathrm{tr}\left(\mathbf{\Sigma}^{-1}\,\mathrm{E}_{(\mathbf{\Sigma},\mathbf{\Delta})}\left[\mathbf{X}\,\mathbf{\Delta}^{-1}\,\mathbf{X}^{\mathrm{T}}\right]\right) - \mathrm{tr}\left(\hat{\mathbf{\Sigma}^{-1}}\mathrm{E}_{(\mathbf{\Sigma},\mathbf{\Delta})}\left[\mathbf{X}\,\hat{\mathbf{\Delta}}^{-1}\,\mathbf{X}^{\mathrm{T}}\right]\right)\right] \\
&= \ldots - \frac{1}{2}\left[\mathrm{tr}\left(\mathbf{\Sigma}^{-1}\left[\mathrm{tr}(\mathbf{\Delta}^{-1}\,\mathbf{\Delta})\,\mathbf{\Sigma}\right]\right) - \mathrm{tr}\left(\hat{\mathbf{\Sigma}}^{-1}\left[\mathrm{tr}(\mathbf{\Delta}\,\hat{\mathbf{\Delta}}^{-1})\,\mathbf{\Sigma}\right]\right)\right] \\
&= \ldots - \frac{1}{2}\left[\mathrm{tr}(\mathbf{\Sigma}^{-1}\,\mathbf{\Sigma})\mathrm{tr}(\mathbf{\Delta}^{-1}\,\mathbf{\Delta}) - \mathrm{tr}(\hat{\mathbf{\Sigma}}^{-1}\,\mathbf{\Sigma})\mathrm{tr}(\hat{\mathbf{\Delta}}^{-1}\,\mathbf{\Delta})\right] \\
&= \frac{p}{2}\log|\hat{\mathbf{\Sigma}}\,\mathbf{\Sigma}^{-1}\,| + \frac{n}{2}\log|\hat{\mathbf{\Delta}}\,\mathbf{\Delta}^{-1}\,| - \frac{np}{2} + \frac{1}{2}\mathrm{tr}\left(\hat{\mathbf{\Sigma}^{-1}}\,\mathbf{\Sigma}\right)\mathrm{tr}\left(\hat{\mathbf{\Delta}^{-1}}\,\mathbf{\Delta}\right).
\end{aligned}$$

Note that $\mathrm{E}_{(\mathbf{\Sigma},\mathbf{\Delta})}\left[\mathbf{X}\,\mathbf{A}\,\mathbf{X}^{\mathrm{T}}\right] = \mathrm{tr}(\mathbf{\Delta}\,\mathbf{A}^T)\,\mathbf{\Sigma}$ where $\mathbf{A}$ is $p \times p$ (15). □

Proof of Proposition 4. We first show that $\mathrm{E}\left[\mathrm{tr}\left(\mathbf{X}^{\mathrm{T}}\,\mathbf{\Sigma}^{-1}\,\mathbf{X}\,\mathbf{\Delta}^{-1}\right)|X_o, \theta'\right]$
$= \mathrm{tr}\left[\left(\hat{\mathbf{X}}^T\,\mathbf{\Sigma}^{-1}\,\hat{\mathbf{X}} + \mathbf{G}(\mathbf{\Sigma}^{-1})\right)\mathbf{\Delta}^{-1}\right]$.
Let $\mathbf{A} = \mathbf{X}^{\mathrm{T}}\,\mathbf{\Sigma}^{-1}\,\mathbf{X}$, then,

$$\mathrm{E}\left[\mathrm{tr}\left(\mathbf{X}^{\mathrm{T}}\,\mathbf{\Sigma}^{-1}\,\mathbf{X}\,\mathbf{\Delta}^{-1}\right)|X_o, \theta'\right] = \mathrm{tr}\left[\mathrm{E}\left(\mathbf{A}\,|X_o, \theta'\right)\mathbf{\Delta}^{-1}\right]$$

$$\begin{aligned}
\text{And, } \mathrm{E}(\mathbf{A}_{jj'}\,|X_o\theta') &= \mathrm{E}\left(X_{\cdot j}^T\,\mathbf{\Sigma}^{-1}\,X_{\cdot j'}|X_o, \theta'\right) \\
&= \mathrm{E}\left[\sum_{k=1}^{n}\sum_{t=1}^{n} x_{tj}x_{kj'}\sigma_{tk}^{-1}|X_o, \theta'\right] \\
&= \sum_{k=1}^{n}\sum_{t=1}^{n}\hat{x}_{tj}\hat{x}_{kj'}\sigma_{tk}^{-1} + \sum_{k=1}^{n}\sum_{t=1}^{n}C_{tk}^{(jj')}\sigma_{tk}^{-1} \\
&= \hat{X}_{\cdot j}^T\,\mathbf{\Sigma}^{-1}\,\hat{X}_{\cdot j'} + \mathrm{tr}\left(\mathbf{C}^{(jj')}\,\mathbf{\Sigma}^{-1}\right)
\end{aligned}$$

Thus, $\mathrm{E}\left(\mathbf{A}\,|X_o, \theta'\right) = \hat{\mathbf{X}}^T\,\mathbf{\Sigma}^{-1}\,\hat{\mathbf{X}} + \mathbf{G}(\mathbf{\Sigma}^{-1})$.
The calculation showing $\mathrm{E}\left[\mathrm{tr}\left(\mathbf{X}^{\mathrm{T}}\,\mathbf{\Sigma}^{-1}\,\mathbf{X}\,\mathbf{\Delta}^{-1}\right)|X_o, \theta'\right] = \mathrm{tr}\left[\left(\hat{\mathbf{X}}\,\mathbf{\Delta}^{-1}\,\hat{\mathbf{X}}^T + \mathbf{F}(\mathbf{\Delta}^{-1})\right)\mathbf{\Sigma}^{-1}\right]$ is the analogous to the above calculation with $\mathbf{B} = \mathbf{X}\,\mathbf{\Delta}^{-1}\,\mathbf{X}^{\mathrm{T}}$ and

$$\mathrm{E}\left(\mathbf{B}_{ii'}\,|X_o, \theta'\right) = \hat{X}_{i\cdot}\,\mathbf{\Delta}^{-1}\,\hat{X}_{i'\cdot}^T + \mathrm{tr}\left(\mathbf{D}^{(ii')}\,\mathbf{\Delta}^{-1}\right).$$

□

Proof of Theorem 2. Notice that Theorem 2 gives the conditional distributions in a two step process. The first step involves finding the distribution of a row or column conditional on the rest of the matrix. Using the notation given, this means $(\mathbf{X}_{i,\cdot} \,|\, \mathbf{X}_{k,\cdot}) = (\mathbf{X}_{i,m_i}, \mathbf{X}_{i,o_i} \,|X_{k,\cdot}) \sim N(\psi, \mathbf{\Gamma})$ and $(\mathbf{X}_{\cdot,j} \,|\, \mathbf{X}_{\cdot,l}) = (\mathbf{X}_{m_j,j}, \mathbf{X}_{o_j,j} \,|X_{\cdot,l}) \sim N(\eta, \mathbf{\Phi})$. These are given in Gupta and Nagar (15). Given the conditional distributions of a row or column, the second step gives the distribution of a set of elements, m_i or m_j within a row or column respectively. Notice that these conditional distributions come directly from the conditional distribution formulas for partitioned multivariate normal matrices, given the partitions of ψ, η, $\mathbf{\Gamma}$ and $\mathbf{\Phi}$. Hence, the connection between the two steps is all that is needed. We show this for part (a) and the proof for part (b) is analogous.

If we let $W = \mathbf{X}_{i,m_i}$, $Y = \mathbf{X}_{i,o_i}$ and $Z = \mathbf{X}_{\neq i,\cdot}$, and assume that they are centered so that $\text{vec}(W \;\; Y \;\; Z) \sim N(\mathbf{0}, \mathbf{\Omega})$, then notice that the first step gives $p\left((W,Y)|Z\right)$ and the second step gives $p\left((W|Z)|(Y|Z)\right)$. We show that this gives the desired conditional distribution.

$$\begin{aligned} p\left((W|Z)|(Y|Z)\right) &= \frac{p\left((W|Z),(Y|Z)\right)}{p\left(Y|Z\right)} \\ &= \frac{p\left(W,Y|Z\right)}{p(Y|Z)} \\ &= p\left(W|Y,Z\right). \end{aligned}$$

Since matrix normal random variables can be written as multivariate normal random variables, we have established that the two steps give the desired conditional distribution. □

Proof of Theorem 3. We will show that the iterates in Steps 2 and 3 of the Alternating Conditional Expectations Algorithm, Algorithm 4, are the block Gauss-Seidel iterates solving the linear system giving $\mathrm{E}\left(\mathbf{X}_m \,|\, \mathbf{X}_o\right)$. For simplicity, we consider $\mathbf{X}_{2\times 2}$ with $\text{vec}(\mathbf{X}) = \left(y_{(2)} \;\; z_{(2)}\right)^T \sim N(\mathbf{0}, \mathbf{\Omega})$ where y is missing and z is observed. *Note the subscript in parentheses denotes vector length.* If we partition $\mathbf{\Omega}$ accordingly, the $k+1$ iteration of Step 2 or 3 is then

$$\begin{aligned} y_1^{(k+1)} &= C \begin{pmatrix} y_2^{(k)} & z \end{pmatrix}^T \\ &\qquad \text{where } C = \begin{pmatrix} C^1_{(1)} & C^2_{(2)} \end{pmatrix} = \mathbf{\Omega}_{y_1,\neq y_1} \, \mathbf{\Omega}^{-1}_{\neq y_1,\neq y_1} \\ y_2^{(k+1)} &= D \begin{pmatrix} y_1^{(k)} & z \end{pmatrix}^T \\ &\qquad \text{where } D = \begin{pmatrix} D^1_{(1)} & D^2_{(2)} \end{pmatrix} = \mathbf{\Omega}_{y_2,\neq y_2} \, \mathbf{\Omega}^{-1}_{\neq y_2,\neq y_2} \end{aligned} \tag{16}$$

Define $\mathbf{A} = \begin{pmatrix} 1 & -C^1 \\ -D^1 & 1 \end{pmatrix}$ and $b = \begin{pmatrix} (C^2)^T z \\ (D^2)^T z \end{pmatrix}$. Then, solving the linear system $\mathbf{A}y = b$ gives the conditional expectation, $\mathbf{A}^{-1}b = \mathrm{E}\left(y|z\right) = \mathbf{\Omega}_{y,z} \, \mathbf{\Omega}^{-1}_{zz}(z)$. In addition, the Gauss-Seidel iterates,

$$\begin{aligned} y_1^{(k+1)} &= \left(b_1 - a_{12} y_2^{(k)}\right) / a_{11} \\ y_2^{(k+1)} &= \left(b_2 - a_{21} y_1^{(k)}\right) / a_{22} \end{aligned}$$

give back equations (16). Thus, finding the conditional expectation of each individual missing value in an iterative fashion converges to the conditional expectation of all missing values given the observed data. Steps 2 and 3 of the Alternating Conditional Expectation Algorithm, however, group the missing values by row or by column. Thus, these steps form the block Gauss-Seidel iterates which also converge, often with a faster rate of convergence (17). In addition, note that the groups of missing values by row and by column form overlapping blocks, or multi-splittings. These updating schemes have been shown to often converge faster than non-overlapping schemes in linear iterative methods (13). □

Acknowledgments

Thanks to Steven Boyd for a discussion about the minimization of biconvex functions. Also, thank you to the editor, associate editor and referees for the helpful comments and suggestions that led to numerous improvements in this paper.

References

[1] Robert M. Bell, Yehuda Koren, and Chris Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. *Proceedings of KDD Cup and Workshop*, 2007.

[2] James Bennett and Stan Lanning. The Netflix prize. *Proceedings of KDD Cup and Workshop*, 2007.

[3] Biernacki,, Christophe and Celeux,, Gilles and Govaert,, Gérard. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Comput. Stat. Data Anal.*, 2003.

[4] Candes, E. J. and Recht, B. Exact matrix completion via convex optimization. To Appear,*Found. of Comput. Math.* , 2008.

[5] Celeux, G. Chauveau, D. Diebolt, J. Stochastic versions of the EM algorithm: an experimental study in the mixture case. *J. Statist. Computn. Simuln.*, 1996.

[6] Michael J. Daniels and Robert E. Kass. Shrinkage estimators for covariance matrices. *Biometrics*, 2001.

[7] Pierre Dutilleul. The mle algorithm for the matrix normal distribution. *J. Statist. Comput. Simul.*, 1999.

[8] Bradley Efron. Correlated z-values and the accuracy of large-scale statistical estimates. Working paper, Stanford University, 2009.

[9] Jianqing Fan and Runze Li Variable selection via penalized likelihood. *J. Amer. Stat. Assoc.*, 2001.

[10] Jeffrey A. Fessler and Alfred O. Hero III. Space-alternating generalized expectation-maximization algorithm. *IEEE Trans. Signal Process.*, 1994.

[11] Jeffrey A. Fessler and Alfred O. Hero III. Penalized maximum-likelihood image reconstruction using space-alternating generalized em algorithms. *IEEE Trans. on Image Process.*, 1995.

[12] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the lasso. *Biostatistics*, 2008.

[13] Andreas Frommer and Bert Phol. A comparison result for multi-splittings and waveform relaxation methods. *Numer. Linear Algebra with Appl.*, 1995.

[14] Peter J. Green. On use of the em for penalized likelihood estimation. *J. R. Statisit. Soc. B.*, 1990.

[15] Arjun K. Gupta and D. K. Nagar. *Matrix variate distributions*. CRC Press, 1999.

[16] H. Kim, G. Golub, and H. Park. Missing value estimation for dna microarray gene expression data: local least squares imputation. *Bioinformatics*, 2005.

[17] Paul J. Lanzkron, Donald J. Rose, and Daniel B. Szyld. Convergence of nested classical iterative methods for linear systems. *Numerical Mathematics*, 1991.

[18] Lin,, Y. and Zhang,, H. H. Component selection and smoothing in multivariate nonparametric regression. *Ann. Statist.*, 2007.

[19] Roderick J. A. Little and Donald B. Rubin. *Statistical analysis with missing data*. Wiley-Interscience, 2002.

[20] Xiao-Li Meng. On the rate of convergence of the ecm algorithm. *Ann. of Statist.*, 1994.

[21] Xiao-Li Meng and Donald Rubin. Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika*, 1993.

[22] Terence Orchard and Max A. Woodbury. A missing information principle: theory and applications. *Proc. Sixth Berkeley Symp. on Math. Statist. and Prob.*, 1972.

[23] Alvaro R. De Pierro. Modified expectation maximization algorithm for penalized likelihood estimation in emission tomography. *IEEE Trans. on Medical Imaging*, 1995.

[24] G. O. Roberts and S. K. Sahu. Updating schemes, correlation structure, blocking and parameterization for the gibbs sampler. *J. R. Statist. Soc. B.*, 1997.

[25] Rothman, A. J. and Bickel, P. J. and Levina, E. and Zhu, J. Sparse permutation invariant covariance estimation. *Electron. J. Stat.*, 2008.

[26] Rubin, Donald B. Multiple imputation after 18+ years. *J. Amer. Statist. Assoc.*, 1996.

[27] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted botlzmann machines for collaborative filtering. Technical report, University of Tronto, 2008.

[28] Tseng, P. Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.*, 2001.

[29] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 2001.

[30] Daniela M. Witten and Robert Tibshirani. Covariance-regularized regression and classification for high-dimensional problems. *J. R. Statist. Soc. B.*, 2008.

[31] Hongjuan Zhao, Robert Tibshirani, and James Brooks. Gene expression profiling predicts survival in conventional renal cell carcinoma. *PLOS Medicine*, pages 511–533, 2005.