The Annals of Applied Statistics
2010, Vol. 4, No. 2, 764–790
DOI: 10.1214/09-AOAS314

TRANSPOSABLE REGULARIZED COVARIANCE MODELS WITH AN APPLICATION TO MISSING DATA IMPUTATION

BY GENEVERA I. ALLEN AND ROBERT TIBSHIRANI

Stanford University

Missing data estimation is an important challenge with high-dimensional data arranged in the form of a matrix. Typically this data matrix 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. By placing additive penalties on the inverse covariance matrices of the rows and columns, these so-called transposable regularized covariance models allow for maximum likelihood estimation of the mean and nonsingular covariance matrices. Using these models, we formulate EM-type algorithms for missing data imputation in both the multivariate and transposable frameworks. We present theoretical results exploiting the structure of our transposable models that allow these models and imputation methods to be applied to high-dimensional data. 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 data sets have become more common in biological and data mining applications, missing data imputation is a significant challenge. We motivate missing data estimation in matrix data with the example of the Netflix movie rating data [Bennett and Lanning (2007)]. This data set 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. The movies and customers, however, are very correlated and an imputation method should take advantage of these relationships. Customers who enjoy horror films, for example, are likely to rate movies similarly, in the same way that horror films are likely to have similar ratings from these customers. Modeling the ratings by the relationships between only the movies or only the customers, as with multivariate methods and k-nearest neighbor methods, seems shortsighted. Customer A's rating of Movie 1, for example, is related to Customer B's rating of Movie 2 by more than simply the connection between Customer A and B or Movie 1 and 2. In addition, modeling

Received April 2009; revised November 2009.

Key words and phrases. Matrix-variate normal, covariance estimation, imputation, EM algorithm, transposable data.

ratings as a linear combination of the ratings of movies or a combination of customer ratings as with singular value decomposition (SVD) methods fails to capture a more sophisticated connection between the movies and customers [Troyanskaya et al. (2001)]. Bell et al., in their discussion of imputation for the Netflix data, call all of these methods either "movie-centric" or "user-centric" [Bell, Koren and Volinsky (2007)].

We propose to directly model the correlations among and between both the customers (rows) and the movies (columns). Thus, our model is *transposable* in the sense that it treats both the rows and columns as features of interest. The model is based on the matrix-variate normal distribution brought to our attention by Efron (2009), which has separate covariance matrix parameters for both the rows and the columns. Thus, both the relationships between customers and between movies are incorporated in the model. 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. This means that the relationship between Customer A's rating of Movie 1 and Customer B's rating of Movie 2 can be modeled directly as the interaction between Customers A and B and Movies 1 and 2.

In practice, however, transposable models based on the matrix-variate normal distribution have largely been of theoretical interest and have rarely applied to real data sets because of the computational burden of high-dimensional parameters [Gupta and Nagar (1999)]. In this paper we introduce modifications of the matrix-variate normal distribution, specifically restrictions on the means and penalties on the inverse covariances, that allow us to fit this transposable model to a single matrix of data. The penalties we employ give us nonsingular covariance estimates that have connections to the singular value decomposition and graphical models. With this theoretical foundation, we present computationally efficient Expectation Maximization-type (EM) algorithms for missing data imputation. We also 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 numerous applications including user-ratings data.

We organize the paper beginning with a review of the multivariate regularized covariance models (RCM) and a new imputation method based on these models, Section 2. The RCMs form the foundation for the transposable regularized covariance models (TRCM) introduced in Section 3. We then present new EM-type imputation algorithms for transposable data, Section 4, along with a one-step approximation in Section 4.2. Simulations and results on microarray and the Netflix data are given in Section 5, and we conclude with a discussion of our methods in Section 6.

2. Regularized covariance models and imputation with multivariate data. Several recent papers have presented algorithms and discussed applications of regularized covariance models (RCM) for the multivariate normal distribution [Friedman, Hastie and Tibshirani (2007); Witten and Tibshirani (2009)]. 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 nonsingular, thus enabling covariance estimation when the number of features is greater than the number of observations. In this section we give a review of these models and briefly describe a new penalized EM algorithm for imputation of missing values using the regularized covariance model.

Let $X_i \sim N(0, \mathbf{\Delta})$ 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 of the regularized covariance model is then proportional to

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

where $\| \cdot \|^q = \sum_{i=1}^{p^2} | \cdot |^q$ and q is either 1 or 2, that is, 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 $\rho \operatorname{tr}(\mathbf{\Delta}^{-1}\mathbf{\Delta}^{-1}) = \rho \|\mathbf{\Delta}^{-1}\|_F^2$.

Maximizing $\ell(\mathbf{\Delta})$ gives the penalized-maximum likelihood estimate (MLE) of $\mathbf{\Delta}$. Friedman, Hastie and Tibshirani (2007) 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 $\hat{\mathbf{\Delta}}^{-1}$. A zero in the ijth component of $\mathbf{\Delta}^{-1}$ implies that variables i and j are conditionally independent given the other variables. Thus, these penalized-maximum likelihood models with L_1 penalties can be used to estimate sparse undirected graphs. With an L_2 penalty, the problem has an analytical solution [Witten and Tibshirani (2009)]. If we take the singular value decomposition (SVD) of $\mathbf{X}$, $\mathbf{X} = \mathbf{UDV}^T$, with $d = \operatorname{diag}(\mathbf{D})$, then

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

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 nonzero eigenvalues. Regularized covariance models provide an alternative method of estimating the covariance matrix with many desirable properties [Rothman et al. (2008)].

With this underlying model, we can form a new missing data imputation algorithm by maximizing the observed penalized log-likelihood of the regularized covariance model via the EM algorithm. Our method is the same as that of the EM algorithm for the multivariate normal described in Little and Rubin [Little and Rubin (2002)], except for an addition in the maximization step. In our M step, we find the MLE of the RCM covariance matrix instead of the multivariate normal MLE. Thus, our method fits into a class of penalized EM algorithms which give nonsingular covariance estimates [Green (1990)], thus enabling use of the EM framework when $p > n$. We give full details of the algorithm, which we call *RCMimpute*, in the Supplementary Materials [Allen and Tibshirani (2010)]. As we will discuss later, this imputation algorithm is a special case of our algorithm for transposable data and forms an integral part of our one-step approximation algorithm presented in Section 4.2.

3. Transposable regularized covariance models.

As previously mentioned, we model the possible dependencies between and within the rows and columns using the matrix-variate normal distribution. In this section we first present a modification of this model, 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 parameter estimates and illustrate the connections between these estimates and those of multivariate models, the singular value decomposition and graphical models.

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 [Gupta and Nagar (1999)]. 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 $\mathbf{X}$ [Dutilleul (1999)], 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 interpretability.

We denote the mean-restricted matrix-variate normal distribution by $\mathbf{X} \sim N_{n,p}(\boldsymbol{\nu}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\Delta})$ 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 $\boldsymbol{\Sigma} \in \Re^{n\times n}$ and the column covariance $\boldsymbol{\Delta} \in \Re^{p\times p}$. If we place the matrix $\mathbf{X}$ into a vector of length np, we have $\mathrm{vec}(\mathbf{X}) \sim N(\mathrm{vec}(\mathbf{M}), \boldsymbol{\Omega})$, where $\mathbf{M} = \nu \mathbf{1}_{(p)}^{T} + \mathbf{1}_{(n)}\mu^{T}$, and $\boldsymbol{\Omega} = \boldsymbol{\Delta} \otimes \boldsymbol{\Sigma}$. 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. This covariance structure can be seen as a tensor product Gaussian process on the rows and columns, an approach explored in Bonilla, Chai and Williams (2008) and Yu et al. (2007).

This distribution implies that a single element, X_{ij}, has mean $\nu_i + \mu_j$ along with variance $\mathbf{\Sigma}_{ii}\mathbf{\Delta}_{jj}$, a mean and variance component from the row and column to which it belongs. As pointed out by a referee, this can be viewed as the following random effects model: $X_{ij} = \nu_i + \mu_j + \varepsilon_{ij}$, where $\varepsilon_{ij} \sim N(0, \mathbf{\Sigma}_{ii}\mathbf{\Delta}_{jj})$, which has two additive fixed effects depending on the row and column means and a random effect whose variance depends on the product of the corresponding row and column covariances. This model shares the same first and second moments as elements from the mean-restricted matrix-variate normal. It does not, however, capture the Kronecker covariance structure between the elements of $\mathbf{X}$ unless both the row and column covariances, $\mathbf{\Sigma}$ and $\mathbf{\Delta}$, are diagonal. This random effect model differs from the more common two-way random effects model with additive errors, which assumes that errors from the two sources are independent. Our model, however, assumes that the errors are related and models them as an interaction effect. A similar random effects approach was taken in Yu et al. (2009), also using a Kronecker product covariance matrix.

To further illustrate the model, we note that the rows and columns are both marginally multivariate normal. The ith row, denoted as X_{ir}, is distributed as $X_{ir} \sim N(\nu_i + \mu, \Sigma_{ii}\mathbf{\Delta})$ and the jth column, denoted by X_{cj}, is distributed as $X_{cj} \sim N(\nu + \mu_j, \Delta_{jj}\mathbf{\Sigma})$. 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 $\mathbf{X} \sim N(\nu, \mathbf{\Sigma})$. Also, two elements from different rows or columns are distributed as a bivariate normal, $(X_{ij}, X_{i'j'}) \sim N(\binom{\nu_i+\mu_j}{\nu_{i'}+\mu_{j'}}, (\begin{smallmatrix} \mathbf{\Sigma}_{ii}\mathbf{\Delta}_{jj} & \mathbf{\Sigma}_{ii'}\mathbf{\Delta}_{jj'} \\ \mathbf{\Sigma}_{i'i}\mathbf{\Delta}_{j'j} & \mathbf{\Sigma}_{i'i'}\mathbf{\Delta}_{j'j'} \end{smallmatrix}))$. Thus, our model is more general than the multivariate normal, with the flexibility to encompass many different marginal multivariate models.

For completeness, the density function of this distribution is

$$\begin{aligned} &p(\nu, \mu, \mathbf{\Sigma}, \mathbf{\Delta}) \\ &\quad = (2\pi)^{-np/2} |\mathbf{\Sigma}|^{-p/2} |\mathbf{\Delta}|^{-n/2} \\ &\qquad \times \operatorname{etr}\bigl(-\tfrac{1}{2}(\mathbf{X} - \nu \mathbf{1}_{(p)}^T - \mathbf{1}_{(n)}\mu^T)\mathbf{\Delta}^{-1}(\mathbf{X} - \nu \mathbf{1}_{(p)}^T - \mathbf{1}_{(n)}\mu^T)^T \mathbf{\Sigma}^{-1}\bigr), \end{aligned}$$

where $\operatorname{etr}(\cdot)$ 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*). 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 nonsingular 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

$$\begin{aligned}
&\ell(\nu, \mu, \mathbf{\Sigma}, \mathbf{\Delta}) \\
&\quad = \frac{p}{2} \log |\mathbf{\Sigma}^{-1}| + \frac{n}{2} \log |\mathbf{\Delta}^{-1}| \\
&\qquad - \frac{1}{2} \operatorname{tr}\big(\mathbf{\Sigma}^{-1}(\mathbf{X} - \nu \mathbf{1}_{(p)}^T - \mathbf{1}_{(n)}\mu^T)\mathbf{\Delta}^{-1}(\mathbf{X} - \nu \mathbf{1}_{(p)}^T - \mathbf{1}_{(n)}\mu^T)^T\big) \\
&\qquad - \rho_r \|\mathbf{\Sigma}^{-1}\|^{q_r} - \rho_c \|\mathbf{\Delta}^{-1}\|^{q_c},
\end{aligned} \tag{3}$$

where $\|\cdot\|^{q_r} = \sum_{i=1}^{m^2} |\cdot|^{q_r}$ and q_r and q_c are either 1 or 2, that is, 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.

With transposable regularized covariance models, as with their multivariate counterpart, the penalties are placed on the inverse covariance matrix, or concentration matrix. Estimation of the concentration matrix has long been associated with graphical models, especially with an L_1 penalty which is useful to model sparse graphical models [Friedman, Hastie and Tibshirani (2007)]. Here, a nonzero entry of the concentration matrix, $\mathbf{\Sigma}_{ij} \neq 0$, means that the ith row conditional on all other rows is correlated with row j. Thus, a "link" is formed in the graph structure between nodes i and j. Conversely, zeros in the concentration matrix imply conditional independence. Hence, since we are estimating both a regularized row and column concentration matrix, our model can be interpreted as modeling both the rows and columns with a graphical model.

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*

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

where X_{cj} denotes the jth column and X_{ir} the ith row of $\mathbf{X} \in \Re^{n \times p}$.

PROOF. See Supplementary Materials. □

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 bi-concave 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 Supplementary Materials. □

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. We also note 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:

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

Maximization with L_1 penalties can be achieved by applying the graphical lasso algorithm to the second term with the coefficient of the third term 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 $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ with $d = \operatorname{diag}(\mathbf{D})$ and let r be the rank of $\mathbf{X}$, then*

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

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

$$\beta_i^* = \begin{cases} 2\sqrt{\dfrac{\rho_r}{p}}, & \textit{if } i \geq r, \\ \sqrt{\dfrac{-c_2^{(i)} - \sqrt{c_2^{(i)2} - 4c_1^{(i)}c_3^{(i)}}}{2c_1^{(i)}}}, & \textit{otherwise}, \end{cases}$$

$$\theta_i^* = \begin{cases} 2\sqrt{\dfrac{\rho_c}{n}}, & \textit{if } i \geq r, \\ \dfrac{d_i^2\beta_i^*}{p\beta_i^{*2} - 4\rho_r}, & \textit{otherwise}, \end{cases}$$

with coefficients

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

PROOF. See Supplementary Materials. □

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):

$$p\theta_i\beta_i^2 - d_i^2\beta_i - 4\rho_r\theta_i = 0 \quad \text{and} \quad n\beta_i\theta_i^2 - d_i^2\theta_i - 4\rho_c\beta_i = 0.$$

These are two quadratic functions in β and θ, so the quadratic formula gives us the eigenvalues in terms of each other. 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. 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. We give numerical results comparing our TRCM covariance estimates to other shrinkage covariance estimators in the Supplementary Materials.

Here, we also pause to compare our TRCM model with L_2 penalties to the singular value decomposition model commonly employed with matrix data. If we include both row and column intercepts, we can write the rank-reduced SVD model as $X_{ij} = \nu_i + \mu_j + \mathbf{u}_i^T \mathbf{D}_r \mathbf{v}_j + \varepsilon$, where $\mathbf{u}_i$ and $\mathbf{v}_j$ are the ith and jth right and left singular vectors, $\mathbf{D}_r$ is the rank-reduced diagonal matrix of singular values and $\varepsilon \sim N(0, \sigma^2)$. Thus, the model appears similar to L_2 TRCM, which can be written as $X_{ij} = \nu_i + \mu_j + \varepsilon_{ij}$ where $\varepsilon_{ij} \sim N(0, \mathbf{u}_i^T \operatorname{diag}(\beta)\mathbf{u}_i * \mathbf{v}_j^T \operatorname{diag}(\theta)\mathbf{v}_j)$. There are important differences between the models, however. First, the left and right singular vectors are incorporated directly into the SVD model, whereas they form the bases of the variance component of TRCM. Second, a rank-reduced SVD incorporates only the first r left and right singular vectors. Our model uses all the singular vectors as β and θ are of lengths n and p, respectively. Finally, the SVD allows the covariances of the rows to vary with $\mathbf{u}_i$, whereas with TRCM the rows share a common covariance matrix. Thus, while the SVD and TRCM share similarities, the models differ in structure and, hence, each offers a separate approach to matrix-data.

4. Imputation for transposable data.

Imputation methods for transposable data are the main focus of this paper. We formulate methods based on the transposable regularized covariance models introduced in Section 3. Because computational costs have limited use of the matrix-variate normal in applications, we let computational considerations motivate the formulation of our imputation methods.

We propose a Multi-Cycle Expectation Conditional Maximization (MCECM) algorithm, given by Meng and Rubin (1993), maximizing the observed penalized log-likelihood of the transposable regularized covariance models. The algorithm exploits the 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 algorithm mathematically, provide some rationale behind the structure of the algorithm via numerical examples, and then briefly discuss computational strategies and considerations.

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 4.2. 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. We also demonstrate the utility of this one-step procedure in numerical examples. A Bayesian variation of the one-step approximation using Gibbs sampling is given in the Supplementary Materials.

Prior to formulating the imputation algorithm for transposable models, we pause to address a logical question: Why do we not use the multivariate imputation method based on regularized covariance models, given that the mean-restricted matrix normal distribution can be written as a multivariate normal with $\mathrm{vec}(\mathbf{X}) \sim N(\mathrm{vec}(\mathbf{M}), \boldsymbol{\Omega})$? 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, $\boldsymbol{\Omega}$, 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 multivariate methods is computational. Recall that $\boldsymbol{\Omega}$ 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 $\boldsymbol{\Omega}$.

4.1. *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 row index and j the column index. 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.

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

$$\begin{aligned}\ell(\nu, \mu, \boldsymbol{\Sigma}, \boldsymbol{\Delta}) = {}& \frac{1}{2}\left[\sum_{j=1}^{p} \log|\boldsymbol{\Sigma}^{-1}_{o_j,o_j}| + \sum_{i=1}^{n} |\boldsymbol{\Delta}^{-1}_{o_i,o_i}|\right] \\ & - \frac{1}{2}\mathrm{tr}\left(\sum_{i=1}^{n}(x^*_{i,o_i} - \mu_{o_i})^T(x^*_{i,o_i} - \mu_{o_i})\boldsymbol{\Delta}^{-1}_{o_i,o_i}\right) \qquad (7) \\ & - \rho_r\|\boldsymbol{\Sigma}^{-1}\|^{q_r} - \rho_c\|\boldsymbol{\Delta}^{-1}\|^{q_c}.\end{aligned}$$

One can show that this is indeed the observed log-likelihood by starting with the multivariate observed log-likelihood and using $\mathrm{vec}(\mathbf{X})$ and the corresponding $\mathrm{vec}(\mathbf{M})$ and $\mathbf{\Omega}$. We maximize (7) via an EM-type 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. Letting $\theta = \{\nu, \mu, \mathbf{\Sigma}, \mathbf{\Delta}\}$, the parameters of the mean-restricted matrix-variate normal, and letting o be the indices of the observed values, the E step, denoted by $Q(\theta|\theta', X_o)$, has the following form. Here, we assume that $\mathbf{X}$ is centered:

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

Thus, we have two equivalent forms of the conditional expectation which we give below.

PROPOSITION 3. *The E step is proportional to the following form*:

$$\begin{aligned} \mathrm{E}[\mathrm{tr}(\mathbf{X}^T\mathbf{\Sigma}^{-1}\mathbf{X}\mathbf{\Delta}^{-1})|X_o, \theta'] &= \mathrm{tr}\big[(\hat{\mathbf{X}}^T\mathbf{\Sigma}^{-1}\hat{\mathbf{X}} + \mathbf{G}(\mathbf{\Sigma}^{-1}))\mathbf{\Delta}^{-1}\big] \\ &= \mathrm{tr}\big[(\hat{\mathbf{X}}\mathbf{\Delta}^{-1}\hat{\mathbf{X}}^T + \mathbf{F}(\mathbf{\Delta}^{-1}))\mathbf{\Sigma}^{-1}\big], \end{aligned} \tag{8}$$

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

$$\begin{aligned} \mathbf{G}(\mathbf{\Sigma}^{-1}) &= \begin{pmatrix} \mathrm{tr}(\mathbf{C}^{(11)}\mathbf{\Sigma}^{-1}) & \cdots & \mathrm{tr}(\mathbf{C}^{(1p)}\mathbf{\Sigma}^{-1}) \\ \vdots & \ddots & \vdots \\ \mathrm{tr}(\mathbf{C}^{(p1)}\mathbf{\Sigma}^{-1}) & \cdots & \mathrm{tr}(\mathbf{C}^{(pp)}\mathbf{\Sigma}^{-1}) \end{pmatrix}, \\ \mathbf{C}^{(jj')} &= \mathrm{Cov}(X_{cj}, X_{cj'}|X_o, \theta'), \\ \mathbf{F}(\mathbf{\Delta}^{-1}) &= \begin{pmatrix} \mathrm{tr}(\mathbf{D}^{(11)}\mathbf{\Delta}^{-1}) & \cdots & \mathrm{tr}(\mathbf{D}^{(1n)}\mathbf{\Delta}^{-1}) \\ \vdots & \ddots & \vdots \\ \mathrm{tr}(\mathbf{D}^{(n1)}\mathbf{\Delta}^{-1}) & \cdots & \mathrm{tr}(\mathbf{D}^{(nn)}\mathbf{\Delta}^{-1}) \end{pmatrix}, \\ \mathbf{D}^{(ii')} &= \mathrm{Cov}(X_{ir}, X_{i'r}|X_o, \theta'). \end{aligned}$$

PROOF. See Supplementary Materials. □

The E step in the matrix-variate normal framework has a similar structure to that of the multivariate normal (see Supplementary Materials) 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 nonzero 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 E step, which will be inserted between the two Conditional Maximization (CM) steps to form the MCECM algorithm.

The CM steps which maximize the conditional expectation functions, in Proposition 3, along with 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:

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

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

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.

We now put these steps together and present the Multi-Cycle ECM algorithm for imputation with transposable data, TRCMimpute, in Algorithm 1. 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,

Algorithm 1 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 nonsingular 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.
```

ignoring the missing values by summing over the observed values, until convergence. Second, the initial estimates of $\hat{\mathbf{\Sigma}}^{-1}$ and $\hat{\mathbf{\Delta}}^{-1}$ must be nonsingular in order to preform the needed computations in the E step. While any nonsingular 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. Some properties and numerical comparisons of the MCECM algorithm are given in the Supplementary Materials.

4.1.2. *Computational considerations.* 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 $\hat{\mathbf{X}} = \mathrm{E}(\mathbf{X}|X_o, \theta')$, and the covariance terms, $\mathbf{C}^{(jj')} = \mathrm{Cov}(X_{cj}, X_{cj'}|X_o, \theta')$ and $\mathbf{D}^{(ii')} = \mathrm{Cov}(X_{ir}, X_{i'r}|X_o, \theta')$. 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 $\mathbf{\Omega}$, that is, if we let m be the indices of the missing values of $\mathrm{vec}(\mathbf{X})$ and o be the observed,

$$\text{(9)} \quad \mathrm{vec}(\hat{\mathbf{X}})_k = \begin{cases} \mathrm{vec}(\mathbf{M})_k + \mathbf{\Omega}_{ko}\mathbf{\Omega}_{oo}^{-1}(\mathrm{vec}(\mathbf{X})_o - \mathrm{vec}(\mathbf{M})_o), & \text{if } k \in m, \\ \mathrm{vec}(X)_k, & \text{if } k \in o, \end{cases}$$

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

$$\text{(10)} \qquad \mathrm{Cov}(\mathrm{vec}(\mathbf{X})_m, \mathrm{vec}(\mathbf{X})_m|X_o, \theta') = \mathbf{\Omega}_{mm} - \mathbf{\Omega}_{mo}\mathbf{\Omega}_{oo}^{-1}\mathbf{\Omega}_{om}.$$

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 [Celeux, Chauveau and Diebolt (1996)] is one computational approach (we present Gibbs sampling as part of our Bayesian one-step approximation in the Supplementary Materials). A stochastic approach, however, is still computationally expensive and, thus, an approximation to our MCECM algorithm is needed.

4.2. *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 which has computational costs comparable to multivariate imputation methods.

4.2.1. *One-step algorithm.* 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 inverse 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 [Lin and Zhang (2006); Fan and Li (2001)]. 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 obtain 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 [Lin and Zhang (2006)].

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, completing the maximization step. In summary, our initial estimates are obtained by applying an EM-type method to the marginal models. Biernacki, Celeux and Govaert (2003) 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 3). 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 2.

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,

Algorithm 2 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:
   $\hat{\mathbf{X}}_m = \mathrm{E}(\mathbf{X}_m | \mathbf{X}_o, \hat{\nu}, \hat{\mu}, \hat{\mathbf{\Sigma}}, \hat{\mathbf{\Delta}})$.
```

then TRCMAimpute can give three sets of missing value estimates. Since it is often unknown whether a given data set 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 the Supplementary Materials.

4.2.2. *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 (9), 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 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*

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_{i,m_i} & \mathbf{X}_{i,o_i} \\ \mathbf{X}_{k,m_i} & \mathbf{X}_{k,o_i} \end{pmatrix} = \begin{pmatrix} \mathbf{X}_{m_j,j} & \mathbf{X}_{m_j,l} \\ \mathbf{X}_{o_j,j} & \mathbf{X}_{o_j,l} \end{pmatrix}, \qquad \mathbf{M} = \begin{pmatrix} \mathbf{M}_{i,r} \\ \mathbf{M}_{k,r} \end{pmatrix} = \begin{pmatrix} \mathbf{M}_{c,j} & \mathbf{M}_{c,l} \end{pmatrix},$$

$$\mathbf{\Sigma} = \begin{pmatrix} \mathbf{\Sigma}_{i,i} & \mathbf{\Sigma}_{i,k} \\ \mathbf{\Sigma}_{k,i} & \mathbf{\Sigma}_{k,k} \end{pmatrix}, \quad \text{and} \quad \mathbf{\Delta} = \begin{pmatrix} \mathbf{\Delta}_{j,j} & \mathbf{\Delta}_{j,l} \\ \mathbf{\Delta}_{l,j} & \mathbf{\Delta}_{l,l} \end{pmatrix},$$

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

$$\psi = \mathbf{M}_{i,r} + \mathbf{\Sigma}_{i,k} \mathbf{\Sigma}_{k,k}^{-1} (\mathbf{X}_{k,r} - \mathbf{M}_{k,r}), \qquad \eta = \mathbf{M}_{c,j} + (\mathbf{X}_{c,l} - \mathbf{M}_{c,l}) \mathbf{\Delta}_{l,l}^{-1} \mathbf{\Delta}_{l,j},$$

$$\mathbf{\Gamma} = [\mathbf{\Sigma}_{i,i} - \mathbf{\Sigma}_{i,k} \mathbf{\Sigma}_{k,k}^{-1} \mathbf{\Sigma}_{k,i}] \otimes \mathbf{\Delta}, \quad \text{and} \quad \mathbf{\Phi} = \mathbf{\Sigma} \otimes [\mathbf{\Delta}_{j,j} - \mathbf{\Delta}_{j,l} \mathbf{\Delta}_{l,l}^{-1} \mathbf{\Delta}_{l,j}].$$

Partition ψ, η, $\mathbf{\Gamma}$ *and* $\mathbf{\Phi}$ *as* $\psi = \binom{\psi_{m_i}}{\psi_{o_i}}$, $\eta = (\eta_{m_j} \ \eta_{o_j})$,

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

Then,

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

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

PROOF. See Supplementary Materials. □

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 3.

THEOREM 3. *Let* $\mathbf{X} \sim N_{n,p}(\nu,\mu,\boldsymbol{\Sigma},\boldsymbol{\Delta})$ *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* $\mathbf{X}_{i,r} = (\mathbf{X}_{i,m_i}\ \mathbf{X}_{i,o_i})$ *and a column* $\mathbf{X}_{c,j} = \left(\begin{smallmatrix}\mathbf{X}_{m_j,j}\\ \mathbf{X}_{o_j,j}\end{smallmatrix}\right)$. *Then*, *the Alternating Conditional Expectations Algorithm*, *Algorithm* 3, *converges to* $\mathrm{E}(\mathbf{X}_m|\mathbf{X}_o)$.

Algorithm 3 Alternating Conditional Expectations Algorithm

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

PROOF. See Supplementary Materials. □

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 3. Thus, Theorems 2 and 3 mean 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 $O(n^3 p^3)$ to $O(n^3 + p^3)$, a substantial savings. In addition, if both the covariance estimates and their inverses are known, then one can use the properties of the Schur complement to further speed computation. For extremely sparse matrices or data with few missing elements, the order of operations is nearly linear in n and p (See Supplementary Materials). 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. 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!

4.2.3. *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 data sets, 25 × 25, from the matrix-variate normal model with autoregressive covariance matrices:

- 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 1. 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 suboptimal stationary point of the observed log-likelihood. For a data set 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 Figure 1. This shows that using marginals to provide a good starting value does indeed start 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 data sets.

5. 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

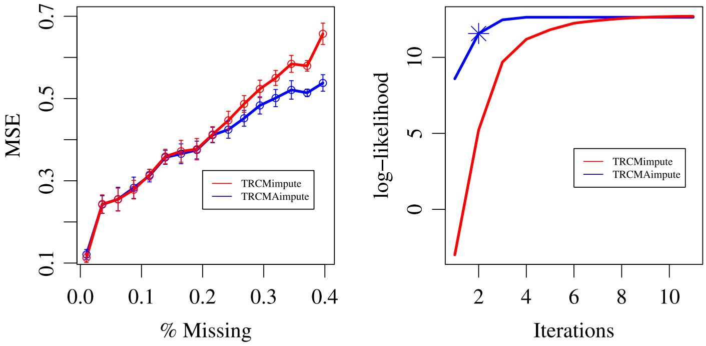

FIG. 1. *Comparison of the mean MSE with standard errors* (*left*) *of the MCECM imputation algorithm* (*TRCMimpute*) *and the one-step approximation* (*TRCMAimpute*) *for transposable data of dimension* 25×25 *with various percentages of missing data. Fifty data sets were simulated from the matrix-variate normal distribution with autoregressive covariances as given in Section* 4.2.3. *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.*

first assess the performance of our one-step approximation, TRCMAimpute, under a variety of simulations with both full and sparse covariance matrices. Authors have suggested that microarrays and user-ratings data, such as the Netflix movie-rating data, are transposable or matrix-distributed [Efron (2009); Bell, Koren and Volinsky (2007)], hence, we also assess the performance of our methods on these types of data sets. We compare performances to three commonly used single imputation methods—SVD methods (SVDimpute), k-nearest neighbors (KNNimpute) and local least squares (LLSimpute) [Troyanskaya et al. (2001); Kim, Golub and Park (2005)]. For the SVD method, we use a reduced rank model with a column mean effect. The rank of the SVD is determined by cross-validation; regularization is not used on the singular vectors so that only one parameter is needed for selection by cross-validation. For k-nearest neighbors and local least squares also, a column mean effect is used and the number of neighbors, k, is selected via cross-validation. If the number of observed elements is limited, the pairwise-complete correlation matrix is used to determine the closest neighbors.

5.1. *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 table:

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 these simulations, the data is of dimension 50×50 with either 25% or 75% of the values missing at random. The 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 with 25% of values missing at random, 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. All three sets of simulations are compared to SVD imputation and k-nearest neighbor imputation.

These simulations show that TRCMAimpute is competitive with 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 with smaller percentages of missing values, cross-validation generally chooses the correct model for the $L_1 : L_1$ penalty-type, but seems to prefer the marginal multivariate models for the $L_2 : L_2$ penalties. However, with 75% of the values missing, the transposable model is often 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 perform well even in the presence of large outliers, as shown in Table 3. Overall, imputation methods based on transposable covariance models compare favorably in these simulations.

TABLE 1

Mean MSE with standard error computed over 50 *data sets of dimension* 50×50 *simulated under the matrix-variate normal distribution with covariances given in Section* 5.1. *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**	
	$L_1 : L_1$	$L_1 : L_2$	$L_2 : L_2$	**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)

5.2. *Microarray data.* Microarrays are high-dimensional matrix-data that often contain missing values. Usually, one assumes that the genes are correlated while the arrays are independent. Efron questions this assumption, however, and suggests using a matrix-variate normal model [Efron (2009)]. Indeed, the matrix-variate framework, and, more specifically, the TRCM model seem appropriate

TABLE 2

Mean MSE with standard errors over 50 *data sets of dimension* 100 × 10 *with* 10% *missing values simulated under the matrix-variate normal with covariances given in Section* 5.1. *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*

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

models for microarray data for several reasons. First, one usually centers both the genes and the arrays before analysis, a structure which is built in to our model. Second, TRCMs have the ability to span many models which include a marginal model where the rows are distributed as a multivariate normal and the arrays are independent. Hence, if a microarray is truly multivariate, our model can accommodate this. But, if there are true correlations within the arrays, TRCM can appropriately measure this correlation and account for it when imputing missing values. Last, the graphical nature of our model can estimate the gene network and then use this information to more accurately estimate missing data.

For our analysis, we use a microarray data set of kidney cancer tumor samples [Zhao, Tibshirani and Brooks (2005)]. The data set 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 1031 genes. Missing values were

TABLE 3

Mean MSE with standard error computed over 50 *data sets 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* 5.1. *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*

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

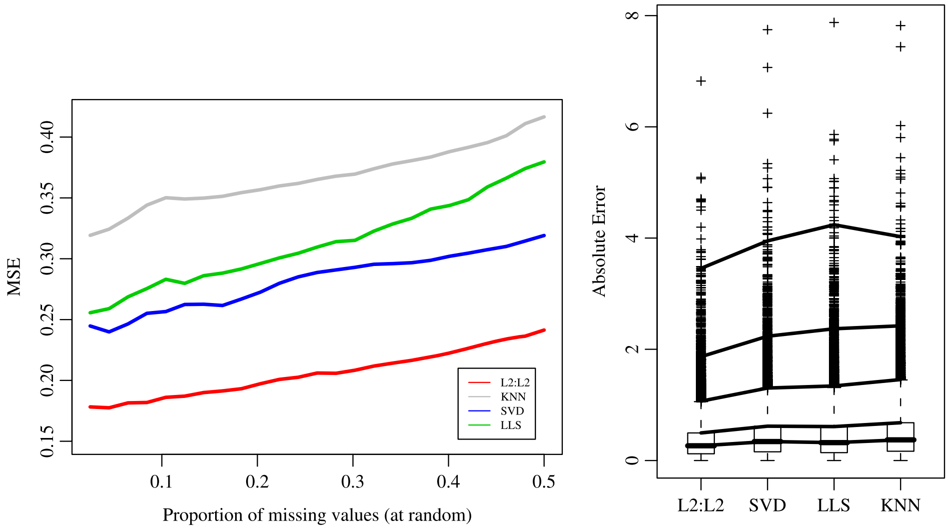

FIG. 2. *Left*: *Comparison of MSE for imputation methods on kidney cancer microarray data with different proportions of missing values. Genes in which all samples are observed are taken with values deleted at random. TRCMAimpute*, $L_2 : L_2$ *and common imputation methods KNNimpute, SVDimpute and LLSimpute 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*. *Right*: *Boxplots of individual absolute errors for various imputation methods. Genes in which all samples are observed were taken and deleted in the same pattern as a random gene in the original data set. 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.

then placed at random. Errors were assessed by comparing the imputed values to the true observed values.

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, on the right in Figure 2. Here, we use L_2 penalties since these are computationally less expensive for high-dimensional data. TRCMAimpute outperforms competing methods in terms of imputation error for all percentages of missing values. We note that cross-validation exclusively chose the marginal, multivariate model from the one-step approximation. This indicates that the arrays in this microarray data set may indeed by independent.

Often, microarray data sets are not missing randomly. Also, researchers are 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 arrays as a randomly sampled gene from the original data set. The right panel of Figure 2 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.

5.3. *Netflix data.* We compare transposable regularized covariance models and existing methods on the Netflix movie rating data [Bennett and Lanning (2007)]. The TRCM framework seems well-suited to model this user-ratings data. As discussed in the Introduction, our model allows for not only correlations among both the customers and movies, but also between them as well. In addition, TRCM models the graph structure of the customers and the movies. Thus, we can fill in a customer's rating of a particular movie based on the customer's links with other customers and the movie's links with other movies. Also, many have noted that the unrated movies in the Netflix data are not simply missing at random and may contain meaningful information. A customer, for example, may not have rated a movie because the movie was not of interest and, thus, they never saw it. While it may appear that our method requires a missing at random assumption, this is not necessarily the case. When two customers have similar sets of unrated movies, after removing the means, our algorithm begins with the unrated movies set to zero. Thus, these two customers would exhibit high correlation simply due to the pattern of missing values. This correlation could yield an estimated "link" between the customers in the inverse covariance matrix. This would then be used to estimate the missing ratings. Hence, our method can find relationships between sets of missing values and use these to impute the missing values.

The Netflix data set 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. In addition, for each customer in this subset ratings were deleted for movies corresponding to the unrated movies of a randomly selected customer with at least one rating out of the 250 movies. This leaves 74% of ratings missing. Figure 3 compares the performances of the TRCM methods to existing methods for both subsets with both missing at random and missing in the pattern of the original values.

Before discussing these results, we first make a note about the comparability of our errors rates to those for the Netflix Prize [Bennett and Lanning (2007)]. Because we chose the subset of data based on the number of observed ratings, we can expect the RMSE to be higher here than applying these methods to the full data set. This method of obtaining a subset leaves out potentially thousands of highly

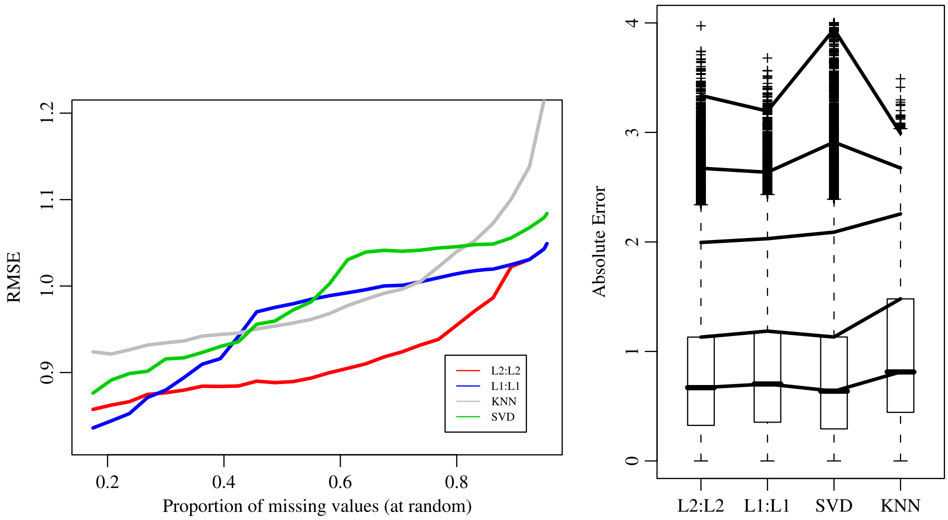

FIG. 3. *Left*: *Comparison of the root MSE* (*RMSE*) *for a subset of the Netflix data for TRCMAimpute*, $L_2 : L_2$ *and* $L_1 : L_1$, *to KNNimpute and SVDimpute. A dense subset was obtained by ranking the movies and customers in terms of number of ratings and taking the top* 250 *movies and* 250 *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. Right*: *Boxplots of absolute errors for the dense subset with missing entries in the pattern of the original data. Customers with at least one ranking out of the* 250 *movies were selected at random and entries were deleted according to these customers leaving* 74% *missing. Quantiles of the absolute errors are shown at* 50%, 75%, 95%, 99% *and* 99.9%. *The RMSE of the methods are as follows* $L_2 : L_2$: 1.005, $L_1 : L_1$: 1.029, *SVD*: 1.032, *KNN*: 1.184.

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 [Salakhutdinov, Mnih and Hinton (2008)], much less than the observed RMSE of 1.084 for the SVD on our subset with 95% missing. Thus, we can conjecture that all of the methods we present would do better in terms of RMSE using the entire data set 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 on the missing at random data. At higher percentages of missing values, our methods perform notably well. With 95% missing values in our subset, TRCMAimpute has a RMSE of 1.049 compared to the SVD at 1.084 and 1.354 using the movie averages. 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, whereas, at larger percentages of missing values, cross-validation chose to penalize both the rows and the columns, indicating that possibly more information can be gleaned from few observed values using transposable methods.

Our methods also preform well when the data is missing in the same pattern as the original. The $L_2 : L_2$ method had the best results with a RMSE of 1.005 followed by $L_1 : L_1$ with 1.029, SVD with 1.032 and k-nearest neighbors with 1.184. From the boxplots of absolute values Figure 3 (right), we see that the SVD has many large outliers in absolute value, while the L_1 penalties led to the fewest number of large errors. Since leading imputation methods for the Netflix Prize are ensembles of many different methods [Bell, Koren and Volinsky (2007)], 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.

6. Discussion.

We have formulated a parametric model for matrix-data along with computational advances that allow this model to be applied to missing value estimation in high-dimensional data sets with possibly complex correlations between and among the rows and columns.

Our MCECM and one-step approximation imputation approaches are restricted to data sets 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 $O(p^3)$. This is roughly on the order of other common imputation methods such as the SVDimpute which costs $O(np^2)$. Our one-step approximation, TRCMAimpute, using the computations for the Alternating Conditional Expectations algorithm given in the Supplementary Materials, costs $O(\sum_{i=1}^{n} \min\{|m_i|, |o_i|\}^3 + \sum_{j=1}^{p} \min\{|m_j|, |o_j|\}^3 + n^3 + p^3)$, where $|m_i|$ and $|o_i|$ are the number of missing and observed elements of row i, respectively.

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 [Candes and Recht (2009)], 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. We present a repeated imputations approach by taking samples from the posterior distribution [Rubin (1996)] with the Bayesian one-step approximation in the Supplementary Materials. 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 that can be extended to further address these issues.

We also pause to address the appropriateness of the Kronecker product covariance matrix to model the covariances observed in real data. While we do not assume that this particular structure is suitable for all data, we feel comfortable using the model because of its flexibility. Recall that all marginal distributions of the mean-restricted matrix-variate normal are multivariate normal. This includes the distribution of elements within a row or column, or the distribution of elements from different rows or columns. All of the marginals of a set of elements are given by the mean and covariance parameters of the elements' rows and columns. Thus, our model says that the location of elements within a matrix determine their distribution, often a reasonable assumption. Also, if either the covariance matrix of the rows or the columns is the identity matrix, then we are back to the familiar multivariate normal model. This flexibility to fit numerous multivariate models and to adapt to structure within a matrix is an important advantage of our matrix-variate model.

Transposable regularized covariance models may be of potential mathematical and practical interest in numerous fields. TRCMs allow for nonsingular 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.

Acknowledgments. Thanks to Steven Boyd for a discussion about the minimization of biconvex functions. We also thank two referees and the Editor for their helpful comments that led to several improvements in this paper.

SUPPLEMENTARY MATERIAL

Additional methods and proofs (DOI: 10.1214/09-AOAS314SUPP; .pdf). This includes sections on the multivariate imputation method *RCMimpute*, numerical results on TRCM covariance estimation, a discussion of properties of the MCECM algorithm for imputation, computations for the Alternating Conditional Expectations Algorithm, a Bayesian one-step approximation to *TRCMimpute* along with a Gibbs sampling algorithm, discussion of cross-validation for estimating penalty parameters, and proofs of theorems and propositions.

REFERENCES

ALLEN, G. I. and TIBSHIRANI, R. (2010). Supplement to "Transposable regularized covariance models with an application to missing data imputation." DOI: 10.1214/09-AOAS314SUPP.

BELL, R. M., KOREN, Y. and VOLINSKY, C. (2007). Modeling relationships at multiple scales to imporve accuracy of large recommender systems. In *Proceedings of KDD Cup and Workshop* 95–104. San Jose.

BENNETT, J. and LANNING, S. (2007). The Netlflix prize. In *Proceedings of KDD Cup and Workshop*. San Jose.

BIERNACKI, C., CELEUX, G. and GOVAERT, G. (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Comput. Statist. Data Anal.* **41** 561–575. MR1968069

BONILLA, E., CHAI, K. M. and WILLIAMS, C. (2008). Multi-task Gaussian process prediction. *Advances in Neural Information Processing Systems* **20** 153–160.

CANDES, E. J. and RECHT, B. (2009). Exact matrix completion via convex optimization. *Found. Comput. Math.* **9** 717–772.

CELEUX, G., CHAUVEAU, D. and DIEBOLT, J. (1996). Stochastic versions of the EM algorithm: An experimental study in the mixture case. *J. Stat. Comput. Simul.* **55** 287–314.

DUTILLEUL, P. (1999). The MLE algorithm for the matrix normal distribution. *J. Stat. Comput. Simul.* **64** 105–123.

EFRON, B. (2009). Are a set of microarrays independent of each other? *Ann. Appl. Statist.* **3** 922–942.

EFRON, B. (2009). Correlated z-values and the accuracy of large-scale statistical estimates. Working paper, Stanford Univ.

FAN, J. and LI, R. (2001). Variable selection via penalized likelihood and its oracle properties. *J. Amer. Stat. Assoc.* **96** 1348–1360. MR1946581

FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2007). Sparse inverse covariance estimation with the lasso. *Biostatistics* **9** 432–441.

GREEN, P. J. (1990). On use of the EM algorithm for penalized likelihood estimation. *J. Roy. Statist. Soc. Ser. B* **52** 443–452. MR1086796

GUPTA, A. K. and NAGAR, D. K. (1999). *Matrix Variate Distributions*. CRC Press, Boca Raton, FL. MR1738933

KIM, H., GOLUB, G. and PARK, H. (2005). Missing value estimation for DNA microarray gene expression data: Local least squares imputation. *Bioinformatics* **21** 187–198.

LIN, Y. and ZHANG, H. H. (2006). Component selection and smoothing in multivariate nonparametric regression. *Ann. Statist.* **34** 2272–2297. MR2291500

LITTLE, R. J. A. and RUBIN, D. B. (2002). *Statistical Analysis with Missing Data*. Wiley, Hoboken, NJ. MR1925014

MENG, X.-L. and RUBIN, D. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika* **80** 267–278. MR1243503

ROTHMAN, A. J., BICKEL, P. J., LEVINA, E. and ZHU, J. (2008). Sparse permutation invariant covariance estimation. *Electron. J. Stat.* **2** 494–515. MR2417391

RUBIN, D. B. (1996). Multiple imputation after 18+ years. *J. Amer. Statist. Assoc.* **91** 473–489.

SALAKHUTDINOV, R., MNIH, A. and HINTON, G. (2008). Restricted Boltzmann machines for collaborative filtering. Technical report, Univ. Toronto.

TROYANSKAYA, O., CANTOR, M., SHERLOCK, G., BROWN, P., HASTIE, T., TIBSHIRANI, R., BOTSTEIN, D. and ALTMAN, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics* **17** 520–525.

WITTEN, D. M. and TIBSHIRANI, R. (2009). Covariance-regularized regression and classification for high-dimensional problems. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **71** 615–636.

YU, K., CHU, W., YU, S., TRESP, V. and XU, Z. (2007). Stochastic relational models for discriminative link prediction. *Advances in Neural Information Processing Systems* **19** 1553–1560.

YU, K., LAFFERTY, J. D., ZHU, S. and GONG, Y. (2009). Large-scale collaborative prediction using a nonparametric random effects model. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14–18, 2009* (A. P. Danyluk, L. Bottou, M. L. Littman, eds.). *ACM International Conference Proceeding Series* **382**. ACM Press, New York.

ZHAO, H., TIBSHIRANI, R. and BROOKS, J. (2005). Gene expression profiling predicts survival in conventional renal cell carcinoma. *PLOS Medicine* **3** 511–533.

DEPARTMENT OF STATISTICS
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305
USA
E-MAIL: giallen@stanford.edu
tibs@stanford.edu