Journal of Machine Learning Research 15 (2014) 1903-1928 Submitted 5/10; Revised 1/14; Published 6/14

Pattern Alternating Maximization Algorithm for Missing Data in High-Dimensional Problems

Nicolas Städler N.STADLER@NKI.NL
The Netherlands Cancer Institute
Plesmanlaan 121
1066 CX Amsterdam, The Netherlands

Daniel J. Stekhoven STEKHOVEN@QUANTIK.CH
Quantik AG
Bahnhofstrasse 57
8965 Berikon, Switzerland

Peter Bühlmann BUHLMANN@STAT.MATH.ETHZ.CH
Seminar for Statistics, ETH Zurich
Rämistrasse 101
8092 Zurich, Switzerland

Editor: Tommi Jaakkola

Abstract

We propose a novel and efficient algorithm for maximizing the observed log-likelihood of a multivariate normal data matrix with missing values. We show that our procedure, based on iteratively regressing the missing on the observed variables, generalizes the standard EM algorithm by alternating between different complete data spaces and performing the E-Step incrementally. In this non-standard setup we prove numerical convergence to a stationary point of the observed log-likelihood. For high-dimensional data, where the number of variables may greatly exceed sample size, we perform regularization using a Lasso-type penalty. This introduces sparsity in the regression coefficients used for imputation, permits fast computation and warrants competitive performance in terms of estimating the missing entries. We show on simulated and real data that the new method often improves upon other modern imputation techniques such as k-nearest neighbors imputation, nuclear norm minimization or a penalized likelihood approach with an ℓ_1-penalty on the concentration matrix.

Keywords: missing data, observed likelihood, (partial) E- and M-Step, Lasso, penalized variational free energy

1. Introduction and Motivation

Missing data imputation for large data sets is a significant challenge in many complex data applications. One well-known example are microarray data sets which contain expression profiles of p genes from a series of n experiments, where p is typically much larger than

n (Troyanskaya et al., 2001; Aittokallio, 2010). In this paper, we propose a novel and computationally efficient imputation algorithm based on missingness pattern alternating maximization in the high-dimensional multivariate normal model. The Gaussian assumption in our model is used for computation of the likelihood but empirical findings suggest that the method is applicable to a wide range of problems where continuous data is arranged in the form of a $n \times p$ matrix with $p \gg n$.

There is a growing literature on missing values in the high-dimensional context (Allen and Tibshirani, 2010; Josse et al., 2011; Loh and Wainwright, 2012; Rosenbaum and Tsybakov, 2010). In recent years, a special focus has been given to the so-called matrix completion problem, where the goal is to recover a low-rank matrix from an incomplete set of entries. It has been shown in a series of fascinating papers that one can recover the missing data entries by solving a convex optimization problem, namely, nuclear-norm minimization subject to data constraints (Candès and Recht, 2009; Candès and Tao, 2010; Keshavan et al., 2010). Efficient algorithms for the matrix completion problem were proposed by Cai et al. (2010) and Mazumder et al. (2010). However, many incomplete data problems do not arise from a near low rank matrix scenario. In these cases there is substantial room to improve upon the convex matrix completion algorithms. We will empirically demonstrate this point for some high-throughput biological data.

In this manuscript we assume a multivariate normal model (MVN) with p-dimensional covariance matrix Σ and address the missing data problem through a likelihood approach (Little and Rubin, 1987; Schafer, 1997). Recently, in the high-dimensional setup, Städler and Bühlmann (2012) proposed to maximize the penalized observed log-likelihood with an ℓ_1-penalty on the concentration matrix Σ^{-1}. They called their method *MissGLasso*, as an extension of the graphical Lasso (Friedman et al., 2008) for missing data. MissGLasso induces sparsity in the concentration matrix and uses an EM algorithm for optimization. Roughly, the algorithm can be summarized as follows: in the E-Step, for each sample, the regression coefficients of the missing against the observed variables are computed from the current estimate $\hat{\Sigma}^{-1}$; in the following M-Step, the missing values are imputed by linear regressions and $\hat{\Sigma}^{-1}$ is re-estimated by applying the graphical Lasso on completed data. There are two main drawbacks of this algorithm in a high-dimensional context. First, the E-Step is rather complex as it involves (for each sample) inversion and multiplication of large matrices in order to compute the regression coefficients. Secondly, a sparse concentration matrix does not imply sparse regression coefficients while we believe that in high-dimensions, sparse regression coefficients would enhance imputations. Our new algorithm, *MissPALasso* in this paper, generalizes the E-Step in order to resolve the disadvantages of MissGLasso. In particular, inversion of a matrix (in order to compute the regression coefficients) will be replaced by a simple soft-thresholding operator. In addition, the regression coefficients will be sparse, which leads to a new sparsity concept for missing data estimation.

MissPALasso emerges from the missingness pattern alternating maximization algorithm (*MissPA*) which we propose for optimizing the (unpenalized) observed log-likelihood. We show that this method generalizes the E- *and* M-Step of the EM algorithm by alternating between different complete data spaces *and* performing the E-Step incrementally (Dempster et al., 1977; Fessler and Hero, 1994; Neal and Hinton, 1998). Such a generalization does not

fit into any of the existing methodologies which extend the standard EM. We analyse our procedure using the variational free energy (Jordan et al., 1999) and prove convergence to a stationary point of the observed log-likelihood.

The further organization of the paper is as follows: Section 2 introduces the setup and the useful notion of missingness patterns. In Section 3 we present our new methodology based on (missingness) pattern alternating maximization and develop MissPALasso for imputation in the high-dimensional scenario. Section 4 compares performance of MissPALasso with other competitive methods and reports on computational efficiency. Finally, in Section 5, we present some theory to gain insights into the numerical properties of the procedure.

2. Setup

We assume $X = (X_1, \ldots, X_p) \sim \mathcal{N}(\mu, \Sigma)$ has a p-variate normal distribution with mean μ and covariance matrix Σ. In order to simplify the notation we set without loss of generality $\mu = 0$: for $\mu \neq 0$, some of the formulae involve the parameter μ and an intercept column of $(1, \ldots, 1)$ in the design matrices but conceptually, we can proceed as for the case with $\mu = 0$. We then write $\mathbf{X} = (\mathbf{X}_{\text{obs}}, \mathbf{X}_{\text{mis}})$, where $\mathbf{X}$ represents an i.i.d. random sample of size n, $\mathbf{X}_{\text{obs}}$ denotes the set of observed values, and $\mathbf{X}_{\text{mis}}$ the missing data.

2.1 Missingness Patterns and Different Parametrizations

For our purpose it will be convenient to group rows of the matrix $\mathbf{X}$ according to their missingness patterns (Schafer, 1997). We index the unique missingness patterns that actually appear in our data by $k = 1, \ldots, s$. Furthermore, with $o_k \subset \{1, \ldots, p\}$ and $m_k = \{1, \ldots, p\} \setminus o_k$ we denote the set of observed variables and the set of missing variables, respectively. $\mathcal{I}_k$ is the index set of the samples (row numbers) which belong to pattern k, whereas $\mathcal{I}_k^c = \{1, \ldots, n\} \setminus \mathcal{I}_k$ stands for the row numbers which do not belong to that pattern. By convention, samples with all variables observed do not belong to a missingness pattern.

Consider a partition $X = (X_{o_k}, X_{m_k})$ of a single Gaussian random vector. It is well known that $X_{m_k}|X_{o_k}$ follows a linear regression on X_{o_k} with regression coefficients $B_{m_k|o_k}$ and covariance $\Sigma_{m_k|o_k}$ given by

$$\begin{aligned} B_{m_k|o_k} &= \Sigma_{m_k,o_k}\Sigma_{o_k}^{-1}, \\ \Sigma_{m_k|o_k} &= \Sigma_{m_k} - \Sigma_{m_k,o_k}\Sigma_{o_k}^{-1}\Sigma_{o_k,m_k}. \end{aligned} \tag{1}$$

Consequently, we can write the density function $p(x; \Sigma)$ of X as

$$p(x; \Sigma) = p(x_{m_k}|x_{o_k}; B_{m_k|o_k}, \Sigma_{m_k|o_k})p(x_{o_k}; \Sigma_{o_k}),$$

i.e., the density can be characterized by either the parameter Σ or $(\Sigma_{o_k}, B_{m_k|o_k}, \Sigma_{m_k|o_k})$. The transformation (1) allows us to switch between both parametrizations.

2.2 Observed Log-Likelihood and Maximum Likelihood Estimation (MLE)

A systematic approach to estimate the parameter of interest Σ from $\mathbf{X}_{\text{obs}}$ maximizes the observed log-likelihood $\ell(\Sigma; \mathbf{X}_{\text{obs}})$ given by

$$\ell(\Sigma; \mathbf{X}_{\text{obs}}) = \sum_{i \notin \bigcup_k \mathcal{I}_k} \log p(x_i; \Sigma) + \sum_{k=1}^{s} \sum_{i \in \mathcal{I}_k} \log p(x_{i,o_k}; \Sigma_{o_k}). \tag{2}$$

Inference for Σ can be based on (2) if the underlying missing data mechanism is *ignorable*. The missing data mechanism is said to be *ignorable* if the probability that an observation is missing may depend on $\mathbf{X}_{\text{obs}}$ but not on $\mathbf{X}_{\text{mis}}$ (*Missing at Random*) and if the parameters of the data model and the parameters of the missingness mechanism are *distinct*. For a precise definition see Little and Rubin (1987).

Explicit maximization of $\ell(\Sigma; \mathbf{X}_{\text{obs}})$ is only possible for special missing data patterns. Most prominent are examples with a so-called monotone missing data pattern (Little and Rubin, 1987; Schafer, 1997), where X_1 is more observed than X_2, which is more observed than X_3, and so on. In this case, the observed log-likelihood factorizes and explicit maximization is achieved by performing several regressions. For a general pattern of missing data, the standard EM algorithm is often used for optimization of (2). See Schafer (1997) for a detailed description of the algorithm. In the next section we present an alternative method for maximizing the observed log-likelihood. We will argue that this new algorithm is computationally more efficient than the standard EM.

3. Missingness Pattern Alternating Maximization

For each missingness pattern, indexed by $k = 1, \ldots, s$, we introduce some further notation:

$$\begin{aligned} \mathbf{X}^k &= (x_{i,j}) \quad \text{with} \quad i \in \mathcal{I}_k, \quad j = 1, \ldots, p \\ \mathbf{X}^{-k} &= (x_{i,j}) \quad \text{with} \quad i \in \mathcal{I}_k^c, \quad j = 1, \ldots, p. \end{aligned}$$

Thus, $\mathbf{X}^k$ is the $|\mathcal{I}_k| \times p$ submatrix of $\mathbf{X}$ with rows belonging to the kth pattern. Similarly, $\mathbf{X}^{-k}$ is the $|\mathcal{I}_k^c| \times p$ matrix with rows not belonging to the kth pattern. In the same way we define $\mathbf{X}^k_{o_k}, \mathbf{X}^k_{m_k}, \mathbf{X}^{-k}_{o_k}$ and $\mathbf{X}^{-k}_{m_k}$. For example, $\mathbf{X}^k_{o_k}$ is defined as the $|\mathcal{I}_k| \times |o_k|$ matrix with

$$\mathbf{X}^k_{o_k} = (x_{i,j}) \quad \text{with} \quad i \in \mathcal{I}_k, \quad j \in o_k.$$

3.1 MLE for Data with a Single Missingness Pattern

Assume that the data matrix $\mathbf{X}$ has only one single missingness pattern, denoted by s. This is the most simple example of a monotone pattern. The observed log-likelihood factorizes according to:

$$\ell(\Sigma; \mathbf{X}_{\text{obs}}) = \sum_{i \in \mathcal{I}_s} \log p(x_{i,o_s}; \Sigma_{o_s}) + \sum_{i \in \mathcal{I}_s^c} \log p(x_i; \Sigma)$$
$$= \sum_{i=1}^{n} \log p(x_{i,o_s}; \Sigma_{o_s}) + \sum_{i \in \mathcal{I}_s^c} \log p(x_{i,m_s} | x_{i,o_s}; B_{m_s|o_s}, \Sigma_{m_s|o_s}). \quad (3)$$

The left and right part in Equation (3) can be maximized separately. The first part is maximized by the sample covariance of the observed variables based on *all samples*, whereas the second part is maximized by a regression of the missing against observed variables based on only the *fully observed samples*. In formulae:

$$\hat{\Sigma}_{o_s} = {}^t\mathbf{X}_{o_s}\mathbf{X}_{o_s}/n, \quad (4)$$

and

$$\hat{B}_{m_s|o_s} = {}^t\mathbf{X}_{m_s}^{-s}\mathbf{X}_{o_s}^{-s}({}^t\mathbf{X}_{o_s}^{-s}\mathbf{X}_{o_s}^{-s})^{-1},$$
$$\hat{\Sigma}_{m_s|o_s} = {}^t(\mathbf{X}_{m_s}^{-s} - \mathbf{X}_{o_s}^{-s}\,{}^t\hat{B}_{m_s|o_s})(\mathbf{X}_{m_s}^{-s} - \mathbf{X}_{o_s}^{-s}\,{}^t\hat{B}_{m_s|o_s})/|\mathcal{I}_s^c|. \quad (5)$$

Having these estimates at hand, it is easy to impute the missing data:

$$\hat{x}_{i,m_s} = \hat{B}_{m_s|o_s}\,{}^t x_{i,o_s} \text{ for all } i \in \mathcal{I}_s, \text{ or, in matrix notation, } \hat{\mathbf{X}}_{m_s}^s = \mathbf{X}_{o_s}^s\,{}^t\hat{B}_{m_s|o_s}.$$

It is important to note, that, if interested in imputation, only the regression part of the MLE is needed and the estimate $\hat{\Sigma}_{o_s}$ in (4) is superfluous.

3.2 MLE for General Missing Data Pattern

We turn now to the general case, where we have more than one missingness pattern, indexed by $k = 1, \ldots, s$. The general idea of the new algorithm is as follows. Assume we have some initial imputations for all missing values. Our goal is to improve on these imputations. For this purpose, we iterate as follows:

- Keep all imputations except those of the 1st missingness pattern fixed and compute the single pattern MLE (for the first pattern) as explained in Section 3.1. In particular, compute the regression coefficients of the missing 1st pattern against all other variables (treated as "observed") based on all samples which do not belong to the 1st pattern.
- Use the resulting estimates (regression coefficients) to impute the missing values from only the 1st pattern.

Next, turn to the 2nd pattern and repeat the above steps. In this way we continue cycling through the different patterns until convergence.

We now describe the missingness pattern alternating maximization algorithm (MissPA) which makes the above idea precise. Let $T = {}^t\mathbf{X}\mathbf{X}$ be the sufficient statistic in the multivariate normal model. Furthermore, we let $T^k = {}^t(\mathbf{X}^k)\mathbf{X}^k$ and $T^{-k} = {}^t(\mathbf{X}^{-k})\mathbf{X}^{-k} = \sum_{l \neq k} T^l$.

Let $\mathcal{T}$ and $\mathcal{T}^k\,(k=1,\ldots,s)$ be some initial guess of T and $T^k\,(k=1,\ldots,s)$, for example, using zero imputation. Our algorithm proceeds as follows:

Algorithm 1: MissPA

(1) $\mathcal{T}$, $\mathcal{T}^k$: initial guess of T and T^k $(k=1,\ldots,s)$.

(2) For $k=1,\ldots,s$ do:

M-Step: Compute the MLE $\hat{B}_{m_k|o_k}$, and $\hat{\Sigma}_{m_k|o_k}$, based on $\mathcal{T}^{-k}=\mathcal{T}-\mathcal{T}^k$:

$$\hat{B}_{m_k|o_k}=\mathcal{T}^{-k}_{m_k,o_k}(\mathcal{T}^{-k}_{o_k,o_k})^{-1},$$
$$\hat{\Sigma}_{m_k|o_k}=\left(\mathcal{T}^{-k}_{m_k,m_k}-\mathcal{T}^{-k}_{m_k,o_k}(\mathcal{T}^{-k}_{o_k,o_k})^{-1}\mathcal{T}^{-k}_{o_k,m_k}\right)/|\mathcal{I}^c_k|.$$

Partial E-Step:

Set $\mathcal{T}^l=\mathcal{T}^l$ for all $l\neq k$ (this takes no time),
Set $\mathcal{T}^k=\mathbb{E}[T^k|\mathbf{X}^k_{o_k},\hat{B}_{m_k|o_k},\hat{\Sigma}_{m_k|o_k}]$,
Update $\mathcal{T}=\mathcal{T}^{-k}+\mathcal{T}^k$.

(3) Repeat step (2) until some convergence criterion is met.

(4) Compute the final maximum likelihood estimator $\hat{\Sigma}$ via:

$\hat{\Sigma}_{o_s}=\mathcal{T}_{o_s,o_s}/n$, $\hat{\Sigma}_{m_s,o_s}=\hat{B}_{m_s|o_s}\hat{\Sigma}_{o_s}$ and $\hat{\Sigma}_{m_s}=\hat{\Sigma}_{m_s|o_s}+\hat{B}_{m_s|o_s}\hat{\Sigma}_{o_s,m_s}$.

Note, that we refer to the maximization step as M-Step and to the imputation step as *partial* E-Step. The word partial refers to the fact that the expectation is only performed with respect to samples belonging to the current pattern. The partial E-Step takes the following simple form:

$$\mathcal{T}^k_{o_k,m_k}={}^t(\mathbf{X}^k_{o_k})\hat{\mathbf{X}}^k_{m_k},$$
$$\mathcal{T}^k_{m_k,m_k}={}^t(\hat{\mathbf{X}}^k_{m_k})\hat{\mathbf{X}}^k_{m_k}+|\mathcal{I}_k|\;\hat{\Sigma}_{m_k|o_k},$$

where $\hat{\mathbf{X}}^k_{m_k}=\mathbb{E}[\mathbf{X}^k_{m_k}|\mathbf{X}^k_{o_k},\hat{B}_{m_k|o_k},\hat{\Sigma}_{m_k|o_k}]=\mathbf{X}^k_{o_k}\,{}^t\hat{B}_{m_k|o_k}$.

Algorithm 1 does not require an evaluation of $\hat{\Sigma}_{o_k}$ in the M-Step, as it is not used in the following partial E-Step. But, if we are interested in the observed log-likelihood or the maximum likelihood estimator $\hat{\Sigma}$ at convergence, we compute $\hat{\Sigma}_{o_s}$ (at convergence), use it together with $\hat{B}_{m_s|o_s}$ and $\hat{\Sigma}_{m_s|o_s}$ to get $\hat{\Sigma}$ via the transformations (1) as explained in step (4).

MissPA is computationally more efficient than the standard EM for missing data: one cycle through all patterns $(k=1,\ldots,s)$ takes about the same time as one iteration of the standard EM. But our algorithm makes more progress since the information from the partial E-Step is employed immediately to perform the next M-Step. We will demonstrate empirically the gain of computational efficiency in Section 4.2. The new MissPA generalizes the standard EM in two ways. Firstly, MissPA alternates between different complete data spaces in the sense of Fessler and Hero (1994). Secondly, the E-Step is performed incrementally (Neal and Hinton, 1998). In Section 5 we will expand on these generalizations and we will provide an appropriate framework which allows analyzing the convergence properties of MissPA.

Finally, a small modification of MissPA, namely replacing in Algorithm 1 the M-Step by

M-Step2: Compute the MLE $\hat{B}_{m_k|o_k}$, and $\hat{\Sigma}_{m_k|o_k}$, based on $\mathcal{T}$:

$$\hat{B}_{m_k|o_k} = \mathcal{T}_{m_k,o_k}(\mathcal{T}_{o_k,o_k})^{-1}$$
$$\hat{\Sigma}_{m_k|o_k} = \left(\mathcal{T}_{m_k,m_k} - \mathcal{T}_{m_k,o_k}(\mathcal{T}_{o_k,o_k})^{-1}\mathcal{T}_{o_k,m_k}\right)/n,$$

results in an alternative algorithm. We show in Section 5 that Algorithm 1 with M-Step2 is equivalent to an incremental EM in the sense of Neal and Hinton (1998).

3.3 High-Dimensionality and Lasso Penalty

The M-Step of Algorithm 1 is basically a multivariate regression of the missing (X_{m_k}) against the observed variables (X_{o_k}). In a high-dimensional framework with $p \gg n$ the number of observed variables $|o_k|$ will be large and therefore some regularization is necessary. The main idea is, in order to regularize, to replace regressions with Lasso analogues (Tibshirani, 1996). We give now the details.

Estimation of $B_{m_k|o_k}$: The estimation of the multivariate regression coefficients in the M-Step2 can be expressed as $|m_k|$ separate minimization problems of the form

$$\hat{B}_{j|o_k} \quad = \quad \arg\min_{\beta} -\mathcal{T}_{j,o_k}\beta + {}^t\beta\mathcal{T}_{o_k,o_k}\beta/2,$$

where $j \in m_k$. Here, $\hat{B}_{j|o_k}$ denotes the jth row vector of the $(|m_k| \times |o_k|)$-matrix $\hat{B}_{m_k|o_k}$ and represents the regression of variable j against the variables from o_k.

Consider now the objective function

$$-\mathcal{T}_{j,o_k}\beta + {}^t\beta\mathcal{T}_{o_k,o_k}\beta/2 + \lambda\|\beta\|_1, \tag{6}$$

with an additional Lasso penalty. Instead of minimizing (6) with respect to β (for all $j \in m_k$), it is computationally much more efficient to perform coordinate-wise improvements from the old parameters (computed in the last cycle through all patterns). For that purpose, let $B^{(r)}_{m_k|o_k}$ be the regression coefficients for pattern k in cycle r and $B^{(r)}_{j|o_k}$ its jth row vector. In cycle $r+1$ we compute $B^{(r+1)}_{j|o_k}$ by minimizing (6) with respect to each of the components of β, holding the other components fixed at their current value. Closed-form updates have the form:

$$B^{(r+1)}_{j|l} \quad = \quad \frac{\mathrm{Soft}\left(\mathcal{T}_{l,l}B^{(r)}_{j|l} - S^{(r)}_l, \lambda\right)}{\mathcal{T}_{l,l}}, \qquad \text{for all } l \in o_k, \tag{7}$$

where

- $B^{(r+1)}_{j|l}$ is the lth component of $B^{(r+1)}_{j|o_k}$ equal to the element (j,l) of matrix $B^{(r+1)}_{m_k|o_k}$
- $S^{(r)}_l$, the gradient of $-\mathcal{T}_{j,o_k}\beta + {}^t\beta\mathcal{T}_{o_k,o_k}\beta/2$ with respect to β_l, which equals

$$S^{(r)}_l = -\mathcal{T}_{j,l} + \sum_{\substack{v<l \\ v\in o_k}} \mathcal{T}_{l,v}B^{(r+1)}_{j|v} + \mathcal{T}_{l,l}B^{(r)}_{j|l} + \sum_{\substack{v>l \\ v\in o_k}} \mathcal{T}_{l,v}B^{(r)}_{j|v} \tag{8}$$

- $\text{Soft}(z, \lambda) = \begin{cases} z - \lambda & \text{if } z > \lambda \\ z + \lambda & \text{if } z < -\lambda \\ 0 & \text{if } |z| \leq \lambda \end{cases}$, is the standard soft-thresholding operator.

In a sparse setup the soft-thresholding update (7) can be evaluated very quickly as l varies. Often coefficients which are zero remain zero after thresholding and therefore nothing has to be changed in (8). See also the *naive-* or *covariance update* of Friedman et al. (2010) for efficient computation of (7) and (8).

Estimation of $\Sigma_{m_k|o_k}$: We update the residual covariance matrix as:

$$\Sigma^{(r+1)}_{m_k|o_k} = \Big(\mathcal{T}_{m_k,m_k} - \mathcal{T}_{m_k,o_k}{}^t B^{(r+1)}_{m_k|o_k} - B^{(r+1)}_{m_k|o_k}\mathcal{T}_{o_k,m_k} + B^{(r+1)}_{m_k|o_k}\mathcal{T}_{o_k,o_k}{}^t B^{(r+1)}_{m_k|o_k}\Big)/n. \tag{9}$$

Formula (9) can be viewed as a generalized version of Equation (5), when multiplying out the matrix product in (5) and taking conditional expectations.

Our regularized algorithm, MissPALasso, is summarized in Algorithm 2. Note, that we update the sufficient statistic in the partial E-Step according to $\mathcal{T} = \gamma\mathcal{T} + \mathcal{T}^k$ where $\gamma = 1 - |\mathcal{I}_k|/n$. This update, motivated by Nowlan (1991), calculates $\mathcal{T}$ as an exponentially decaying average of recently-visited data points. It prevents MissPALasso from storing $\mathcal{T}^k$ for all $k = 1, \ldots, s$ which gets problematic for large p. As we are mainly interested in estimating the missing values, we will output the data matrix with missing values imputed by the regression coefficients $\hat{B}_{m_k|o_k}$ $(k = 1, \ldots, s)$ as indicated in step (4) of Algorithm 2. MissPALasso provides not only the imputed data matrix $\hat{\mathbf{X}}$ but also $\hat{\mathcal{T}}$, the completed version of the sufficient statistic ${}^t\mathbf{X}\mathbf{X}$. The latter can be very useful if MissPALasso is used as a pre-processing step followed by a learning method which is expressible in terms of the sufficient statistic. Examples include regularized regression (e.g., Lasso), discriminant analysis, or estimation of directed acyclic graphs with the PC-algorithm (Spirtes et al., 2000).

By construction, the regression estimates $\hat{B}_{m_k|o_k}$ are sparse due to the employed ℓ_1-penalty, and therefore imputation of missing values $\hat{\mathbf{X}}^k_{m_k} = \mathbf{X}^k_{o_k}{}^t\hat{B}_{m_k|o_k}$ is based on sparse regressions. This is in sharp contrast to the MissGLasso approach (see Section 4.1) which places sparsity on Σ^{-1}. But this does not imply that regressions of variables in m_k on variables in o_k are sparse since the inverse of sub-matrices of a sparse Σ^{-1} are not sparse in general. MissPALasso employs another type of sparsity and this seems to be the main reason for its better statistical performance than MissGLasso.

In practice, we propose to run MissPALasso for a decreasing sequence of values for λ, using each solution as a warm start for the next problem with smaller λ. This pathwise strategy is computationally very attractive and our algorithm converges (for each λ) after a few cycles.

Algorithm 2: MissPALasso

(1) Set $r = 0$ and start with initial guess for $\mathcal{T}$ and $B^{(0)}_{m_k|o_k}$ $(k = 1, \ldots, s)$.

(2) In cycle $r + 1$; for $k = 1, \ldots, s$ do:

Penalized M-Step2:

For all $j \in m_k$, compute $B^{(r+1)}_{j|o_k}$ by improving $-\mathcal{T}_{j,o_k}\beta + {}^t\beta\mathcal{T}_{o_k,o_k}\beta/2 + \lambda\|\beta\|_1$ in a coordinate-wise manner from $B^{(r)}_{j|o_k}$.

Set $\Sigma^{(r+1)}_{m_k|o_k} = \Big(\mathcal{T}_{m_k,m_k} - \mathcal{T}_{m_k,o_k}\,{}^tB^{(r+1)}_{m_k|o_k} - B^{(r+1)}_{m_k|o_k}\mathcal{T}_{o_k,m_k} + B^{(r+1)}_{m_k|o_k}\mathcal{T}_{o_k,o_k}\,{}^tB^{(r+1)}_{m_k|o_k}\Big)/n.$

Partial E-Step:

Set $\mathcal{T}^k = \mathbb{E}[T^k|\mathbf{X}^k_{o_k}, B^{(r+1)}_{m_k|o_k}, \Sigma^{(r+1)}_{m_k|o_k}]$,

Update $\mathcal{T} = \gamma\mathcal{T} + \mathcal{T}^k$ where $\gamma = 1 - |\mathcal{I}_k|/n$.

Increase: $r \leftarrow r + 1$.

(3) Repeat step (2) until some convergence criterion is met.

(4) Output the imputed data matrix $\hat{\mathbf{X}}$, with missing values estimated by: $\hat{\mathbf{X}}^k_{m_k} = \mathbf{X}^k_{o_k}\,{}^t\hat{B}_{m_k|o_k}$, $k = 1, \ldots, s$.

4. Numerical Experiments

In this section we explore the performance of MissPALasso in recovering missing entries and we report on computational efficiency of the algorithm.

4.1 Performance of MissPALasso

Our new approach is compared with the following imputation methods which are well-suited for the high-dimensional context:

- *KnnImpute.* Impute the missing values by the K-nearest neighbors imputation method introduced by Troyanskaya et al. (2001).
- *SoftImpute.* The soft imputation algorithm is proposed by Mazumder et al. (2010) in order to solve the matrix completion problem. They propose to approximate the incomplete data matrix $\mathbf{X}$ by a complete (low-rank) matrix $\mathbf{Z}$ minimizing

$$\frac{1}{2}\sum_{(i,j)\in\Omega}(z_{ij} - x_{ij})^2 + \lambda\|\mathbf{Z}\|_*.$$

Here, Ω denotes the indices of observed entries and $\|\cdot\|_*$ is the nuclear norm, or the sum of the singular values. The missing values of $\mathbf{X}$ are imputed by the corresponding values of $\mathbf{Z}$.

- *MissGLasso.* Compute $\hat{\Sigma}$ by minimizing $-\ell(\Sigma; \mathbf{X}_{\text{obs}}) + \lambda\|\Sigma^{-1}\|_1$, where $\|\cdot\|_1$ is the entrywise ℓ_1-norm. Then, use this estimate to impute the missing values by conditional mean imputation. MissGLasso is described in Städler and Bühlmann (2012).
- *MissPALasso.* This is the method introduced in Section 3.3.

To assess the performances of the methods we use the normalized root mean squared error (Oba et al., 2003) which is defined by

$$\text{NRMSE} \quad = \quad \sqrt{\frac{\textbf{mean}\left((\mathbf{X}^{\text{true}} - \hat{\mathbf{X}})^2\right)}{\textbf{var}\left(\mathbf{X}^{\text{true}}\right)}}.$$

Here, $\mathbf{X}^{\text{true}}$ is the original data matrix (before deleting values) and $\hat{\mathbf{X}}$ is the imputed matrix. With **mean** and **var** we abbreviate the empirical mean and variance, calculated over only the missing entries.

All methods involve one tuning parameter. In KnnImpute we have to choose the number K of nearest neighbors, while SoftImpute, MissGLasso and MissPALasso involve a regularization parameter which is always denoted by λ. In all of our experiments we select the tuning parameters to obtain optimal prediction of the missing entries in terms of NRMSE.

4.1.1 Simulation Study

We consider both high- and a low-dimensional MVN models with $\sim \mathcal{N}_p(0, \Sigma)$ where

- **Model 1:** $p = 50$ and 500;
  Σ: block diagonal with $p/2$ blocks of the form $\left(\begin{smallmatrix} 1 & 0.9 \\ 0.9 & 1 \end{smallmatrix}\right)$.
- **Model 2:** $p = 100$ and 1000;
  Σ: two blocks B_1, B_2 each of size $\frac{p}{2} \times \frac{p}{2}$ with $\mathrm{B}_1 = \mathrm{I}_{\frac{p}{2}}$ and $(\mathrm{B}_2)_{j,j'} = 0.9^{|j-j'|}$.
- **Model 3:** $p = 55$ and 496;
  Σ: block diagonal with $b = 1, \ldots, 10$ for $p = 55$ and $b = 1, \ldots, 31$ for $p = 496$ (increasing) blocks B_b of the size $b \times b$, with $(\mathrm{B}_b)_{j,j'} = 0.9$ $(j \neq j')$ and $(\mathrm{B}_b)_{j,j} = 1$.
- **Model 4:** $p = 100$ and 500;
  $\Sigma_{j,j'} = 0.9^{|j-j'|}$ for $j, j' = 1, \ldots, p$.

For all four settings we perform 50 independent simulation runs. In each run we generate $n = 50$ i.i.d. samples from the model. We then delete randomly 5%, 10% and 15% of the values in the data matrix, apply an imputation method and compute the NRMSE. The results of the different imputation methods are reported in Table 1 for the low-dimensional models and Table 2 for the high-dimensional models. MissPALasso is very competitive in all setups. SoftImpute works rather poorly, perhaps because the resulting data matrices are not well approximable by low-rank matrices. KnnImpute works very well in model 1 and model 4. Model 1, where each variable is highly correlated with its neighboring variable, represents an example which fits well into the KnnImpute framework. However, in model 2 and model 3, KnnImpute performs rather poorly. The reason is that with an inhomogeneous covariance

matrix, as in model 2 and 3, the optimal number of nearest neighbors is varying among the different blocks, and a single parameter K is too restrictive. For example in model 2, a variable from the first block is not correlated to any other variable, whereas a variable from the second block is correlated to other variables. Except for the low-dimensional model 3 MissGLasso is inferior to MissPALasso. Furthermore, MissPALasso strongly outperforms MissGLasso with respect to computation time (see Figure 4 in Section 4.2). Interestingly, all methods exhibit a quite large NRMSE in the high-dimensional model 3. They seem to have problems coping with the complex covariance structure in higher dimensions. If we look at the same model but with $p = 105$ the NRMSE for 5% missing values is: 0.85 for KnnImpute, 0.86 for SoftImpute, 0.77 for MissGLasso and 0.77 for MissPALasso. This indicates an increase in NRMSE according to the size of p. Arguably, we consider here only multivariate normal models which are ideal, from a distributional point of view, for MissGLasso and our MissPALasso. The more interesting case will be with real data (all from genomics) where model assumptions never hold exactly.

		KnnImpute	SoftImpute	MissGLasso	MissPALasso
Model 1	5%	0.4874 (0.0068)	0.7139 (0.0051)	0.5391 (0.0079)	0.5014 (0.0070)
p=50	10%	0.5227 (0.0051)	0.7447 (0.0038)	0.5866 (0.0057)	0.5392 (0.0055)
	15%	0.5577 (0.0052)	0.7813 (0.0037)	0.6316 (0.0048)	0.5761 (0.0047)
Model 2	5%	0.8395 (0.0101)	0.8301 (0.0076)	0.7960 (0.0082)	0.7786 (0.0075)
p=100	10%	0.8572 (0.0070)	0.8424 (0.0063)	0.8022 (0.0071)	0.7828 (0.0066)
	15%	0.8708 (0.0062)	0.8514 (0.0053)	0.8082 (0.0058)	0.7900 (0.0054)
Model 3	5%	0.4391 (0.0061)	0.4724 (0.0050)	0.3976 (0.0056)	0.4112 (0.0058)
p=55	10%	0.4543 (0.0057)	0.4856 (0.0042)	0.4069 (0.0047)	0.4155 (0.0047)
	15%	0.4624 (0.0054)	0.4986 (0.0036)	0.4131 (0.0043)	0.4182 (0.0044)
Model 4	5%	0.3505 (0.0037)	0.5515 (0.0039)	0.3829 (0.0035)	0.3666 (0.0031)
p=100	10%	0.3717 (0.0033)	0.5623 (0.0033)	0.3936 (0.0027)	0.3724 (0.0026)
	15%	0.3935 (0.0032)	0.5800 (0.0031)	0.4075 (0.0026)	0.3827 (0.0026)

Table 1: Average (SE) NRMSE of KnnImpute, SoftImpute, MissGLasso and MissPALasso with different degrees of missingness in the low-dimensional models.

4.1.2 Real Data Examples

We consider the following four publicly available data sets:

- **Isoprenoid gene network in Arabidopsis thaliana:** The number of genes in the network is $p = 39$. The number of observations (gene expression profiles), corresponding to different experimental conditions, is $n = 118$. More details about the data can be found in Wille et al. (2004).

- **Colon cancer:** In this data set, expression levels of 40 tumor and 22 normal colon tissues ($n = 62$) for $p = 2000$ human genes are measured. For more information see Alon et al. (1999).

		KnnImpute	SoftImpute	MissGLasso	MissPALasso
Model 1	5%	0.4913 (0.0027)	0.9838 (0.0006)	0.6705 (0.0036)	0.5301 (0.0024)
p=500	10%	0.5335 (0.0020)	0.9851 (0.0005)	0.7613 (0.0031)	0.5779 (0.0019)
	15%	0.5681 (0.0016)	0.9870 (0.0004)	0.7781 (0.0013)	0.6200 (0.0015)
Model 2	5%	0.8356 (0.0020)	0.9518 (0.0009)	0.8018 (0.0012)	0.7958 (0.0017)
p=1000	10%	0.8376 (0.0016)	0.9537 (0.0007)	0.8061 (0.0002)	0.7990 (0.0013)
	15%	0.8405 (0.0014)	0.9562 (0.0006)	0.8494 (0.0080)	0.8035 (0.0011)
Model 3	5%	1.0018 (0.0009)	0.9943 (0.0005)	0.9722 (0.0013)	0.9663 (0.0010)
p=496	10%	1.0028 (0.0007)	0.9948 (0.0004)	0.9776 (0.0010)	0.9680 (0.0007)
	15%	1.0036 (0.0006)	0.9948 (0.0003)	0.9834 (0.0010)	0.9691 (0.0007)
Model 4	5%	0.3487 (0.0016)	0.7839 (0.0020)	0.4075 (0.0016)	0.4011 (0.0016)
p=500	10%	0.3721 (0.0014)	0.7929 (0.0015)	0.4211 (0.0012)	0.4139 (0.0013)
	15%	0.3960 (0.0011)	0.8045 (0.0014)	0.4369 (0.0012)	0.4292 (0.0014)

Table 2: Average (SE) NRMSE of KnnImpute, SoftImpute, MissGLasso and MissPALasso with different degrees of missingness in the high-dimensional models.

- **Lymphoma:** This data set, presented in Alizadeh et al. (2000), contains gene expression levels of 42 samples of diffuse large B-cell lymphoma, 9 observations of follicular lymphoma, and 11 cases of chronic lymphocytic leukemia. The total sample size is $n = 62$ and $p = 1332$ complete measured expression profiles are documented.

- **Yeast cell-cycle:** The data set, described in Spellman et al. (1998), monitors expressions of 6178 genes. The data consists of four parts, which are relevant to alpha factor (18 samples), elutriation (14 samples), *cdc15* (24 samples), and *cdc28* (17 samples). The total sample size is $n = 73$. We use the $p = 573$ complete profiles in our study.

For all data sets we standardize the columns (genes) to zero mean and variance one. In order to compare the performance of the different imputation methods we randomly delete values to obtain an overall missing rate of 5%, 10% and 15%. Table 3 shows the results for 50 simulation runs, where in each run another random set of values is deleted.

MissPALasso exhibits in all setups the lowest averaged NRMSE. MissGLasso performs nearly as well as MissPALasso on the Arabidopsis data. However, its R implementation cannot cope with large values of p. If we were to restrict our analysis to the 100 variables exhibiting the most variance we would see that MissGLasso performs slightly less well than MissPALasso (results not included). Compared to KnnImpute, SoftImpute works well for all data sets. Interestingly, for all data sets, KnnImpute performance was very inferior compared to MissPALasso. In light of the simulation results of Section 4.1.1, a reason for the poor performance could be that KnnImpute has difficulties with the inhomogeneous correlation structure between different genes which plausibly could be present in real data sets.

To investigate the effect of already missing values on the imputation performance of the compared methods we use the original lymphoma and yeast cell-cycle data sets which already have "real" missing values. We only consider the 100 most variable genes in these

		KnnImpute	SoftImpute	MissGLasso	MissPALasso
Arabidopsis	5%	0.7732 (0.0086)	0.7076 (0.0065)	0.7107 (0.0076)	0.7029 (0.0077)
n=118	10%	0.7723 (0.0073)	0.7222 (0.0052)	0.7237 (0.0064)	0.7158 (0.0060)
p=39	15%	0.7918 (0.0050)	0.7369 (0.0041)	0.7415 (0.0053)	0.7337 (0.0050)
Colon cancer	5%	0.4884 (0.0011)	0.4921 (0.0011)	-	0.4490 (0.0011)
n=62	10%	0.4948 (0.0008)	0.4973 (0.0006)	-	0.4510 (0.0006)
p=2000	15%	0.5015 (0.0007)	0.5067 (0.0006)	-	0.4562 (0.0007)
Lymphoma	5%	0.7357 (0.0014)	0.6969 (0.0008)	-	0.6247 (0.0012)
n=62	10%	0.7418 (0.0009)	0.7100 (0.0006)	-	0.6384 (0.0009)
p=1332	15%	0.7480 (0.0007)	0.7192 (0.0005)	-	0.6525 (0.0008)
Yeast cell-cycle	5%	0.8083 (0.0018)	0.6969 (0.0012)	-	0.6582 (0.0016)
n=73	10%	0.8156 (0.0011)	0.7265 (0.0010)	-	0.7057 (0.0013)
p=573	15%	0.8240 (0.0009)	0.7488 (0.0007)	-	0.7499 (0.0011)

Table 3: Average (SE) NRMSE of KnnImpute, SoftImpute, MissGLasso and MissPALasso for different real data sets from genomics. The R implementation of MissGLasso is not able to handle real data sets of such high dimensionality.

data sets to be able to compare all four methods with each other. From the left panel of Figures 1 and 2 we can read off how many values are missing for each of the 100 variables. In the right panel of Figures 1 and 2 we show how well the different methods are able to estimate $2\%, 4\%, 6\% \ldots, 16\%$ of additionally deleted entries.

4.2 Computational Efficiency

We first compare the computational efficiency of MissPA (Algorithm 1) with the standard EM for missing values described for example in Schafer (1997). A key attribute of MissPA is that the computational cost of one cycle through all patterns is the same as the cost of a single standard EM-iteration. The reason why our algorithm takes less time to converge is that the latent distribution is updated much more frequently. We emphasize the big contrast of MissPA to the incremental EM, mostly applied to finite mixtures (Thiesson et al., 2001; Ng and McLachlan, 2003), where there is a trade-off between the additional computation time per cycle, or "scan" in the language of Ng and McLachlan (2003), and the fewer number of "scans" required because of the more frequent updating after each partial E-Step. The speed of convergence of the standard EM and MissPA for three data sets are shown in Figure 3, in which the log-likelihood is plotted as a function of the number of iterations (cycles). The left panel corresponds to the subset of the lymphoma data set when only the ten genes with highest missing rate are used. This results in a 62×10 data matrix with 22.85% missing values. For the middle panel we draw a random sample of size 62×10 from $\mathcal{N}_{10}(0, \Sigma)$, $\Sigma_{j,j'} = 0.9^{|j-j'|}$, and delete the same entries which are missing in the reduced lymphoma data. For the right panel we draw from the multivariate t-model with one degree of freedom and again with the same values deleted. As can be seen, MissPA converges after fewer cycles. A very extreme example is obtained with the multivariate t-model where the standard EM reaches the log-likelihood level of MissPA about 400 iterations later. We note

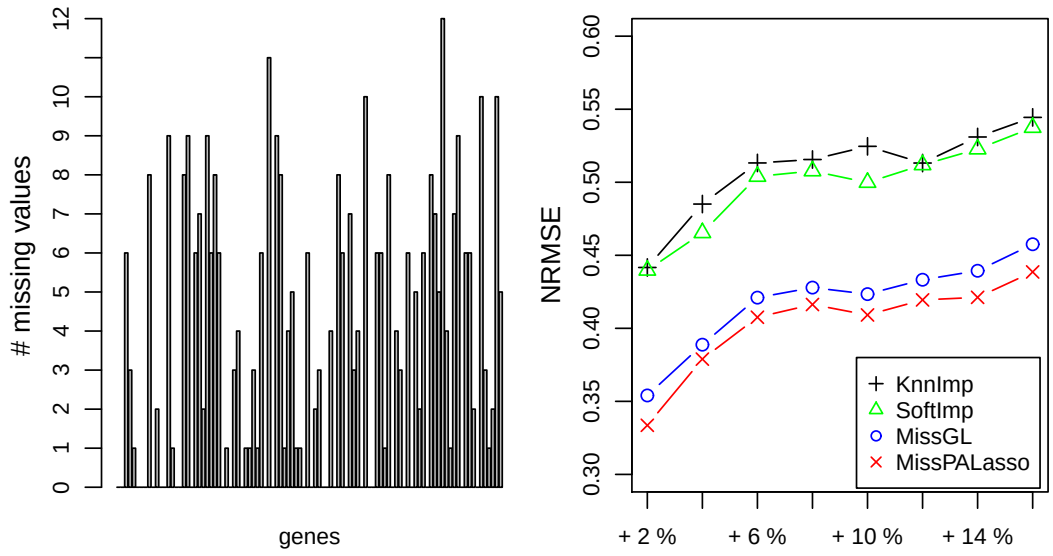

Figure 1: Lymphoma data set. Left panel: Barplots which count the number of missing values for each of the 100 genes. Right panel: NRMSE for KnnImpute, SoftImpute, MissGLasso and MissPALasso if we introduce additional $2\%, 4\%, 6\%, \ldots, 16\%$ missing values.

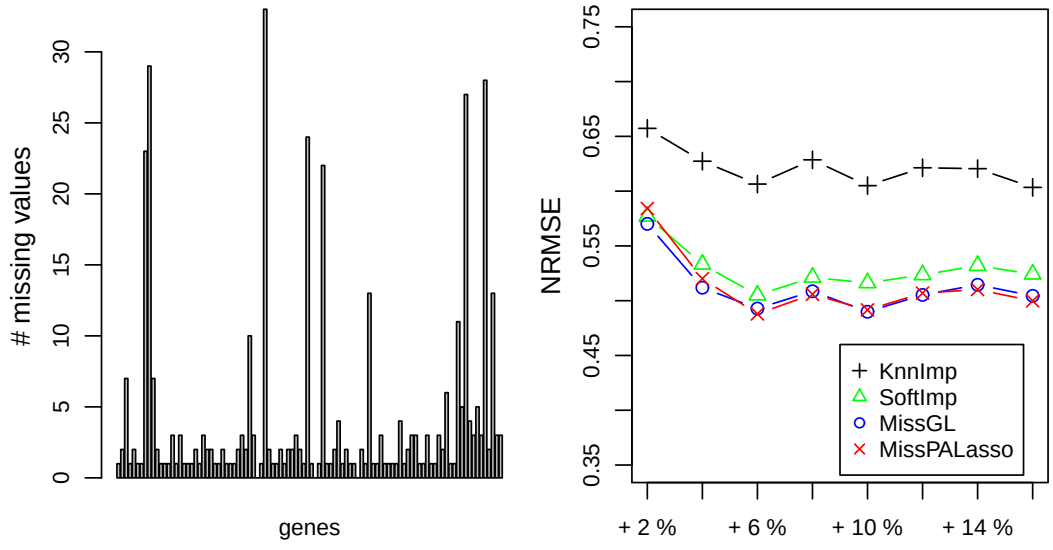

Figure 2: Yeast cell-cycle data set. Left panel: Barplots which count the number of missing values for each of the 100 genes. Right panel: NRMSE for KnnImpute, SoftImpute, MissGLasso and MissPALasso if we introduce additional $2\%, 4\%, 6\%, \ldots, 16\%$ missing values.

here, that the results shown in the middle and right panels highly depend on the realized random sample. With other realizations, we get less and more extreme results than the one shown in Figure 3.

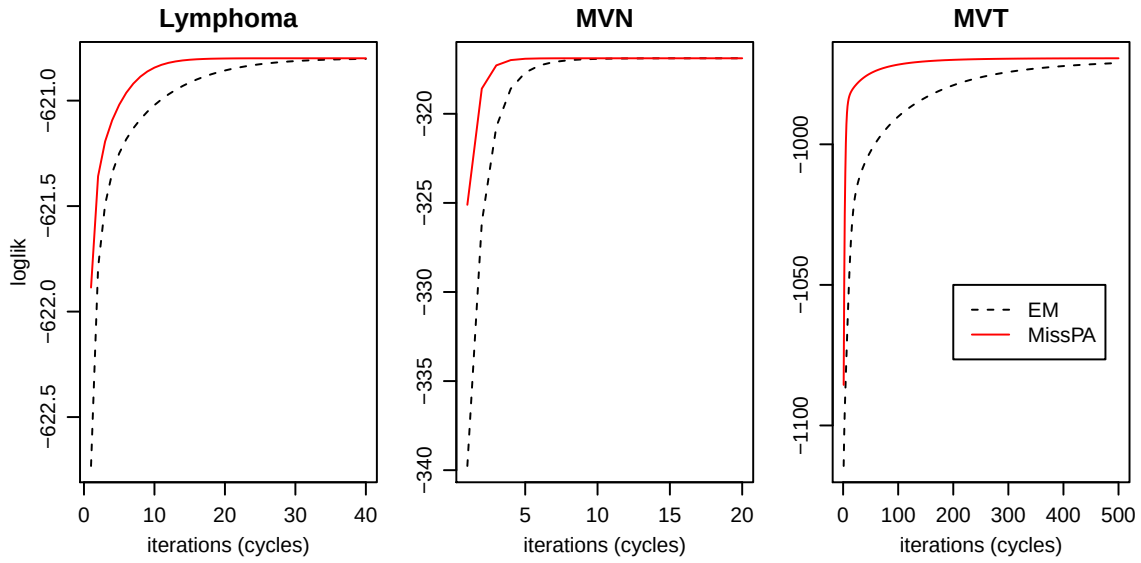

Figure 3: Log-likelihood as a function of the number of iterations (cycles) for standard EM and MissPA. Left panel: subset of the lymphoma data ($n = 62$, $p = 10$ and 22.85% missing values). Middle panel: random sample of size 62×10 from the multivariate normal model with the same missing entries as in the reduced lymphoma data. Right panel: random sample of the size 62×10 from the multivariate t-model again with the same missing values.

We end this section by illustrating the computational timings of MissPALasso and MissGLasso implemented with the statistical computing language R. We consider two settings. Firstly, model 4 of Section 4.1.1 with $n = 50$ and a growing number of variables p ranging from 10 to 500. Secondly, the colon cancer data set from Section 4.1.2 with $n = 62$ and also a growing number of variables where we sorted the variables according to the empirical variance. For each p we delete 10% of the data, run MissPALasso and MissGLasso ten times on a decreasing grid (on the log-scale) of λ values with thirty grid points. For a fixed λ we stop the algorithm if the relative change in imputation satisfies,

$$\frac{\|\hat{\mathbf{X}}^{(r+1)} - \hat{\mathbf{X}}^{(r)}\|^2}{\|\hat{\mathbf{X}}^{(r+1)}\|^2} \leq 10^{-5}.$$

In Figure 4 the CPU times in seconds are plotted for various values of p in the two settings. As shown, with MissPALasso we are typically able to solve a problem of size $p = 100$ in about 9 seconds and a problem of size $p = 500$ in about 400 seconds. For MissGLasso these times are highly increased to 27 and 4300 seconds respectively. Furthermore, we can see that MissPALasso has much smaller variability in runtimes. The computational complexity of MissGLasso is $O\big(p^3 + \sum_{k=1}^{s}(\max\{|m_k|, |o_k|\}|m_k|^2) + np^2\big)$: the

graphical Lasso algorithm costs $O(p^3)$, calculating the coefficients needed in the E-Step involves $O(\sum_{k=1}^{s} \max\{|m_k|, |o_k|\}|m_k|^2)$ operations and updating the sufficient statistic costs $O(np^2)$. In contrast, in a sparse setting, the complexity of MissPALasso is considerably smaller: MissPALasso costs $O\big(\sum_{k=1}^{s} \big(\max\{|m_k|, |o_k|\} \sum_{j \in m_k} q_j\big) + np^2\big)$ operations where q_j denotes the average number of nonzero elements in $B^{(r)}_{j|o_k}$.

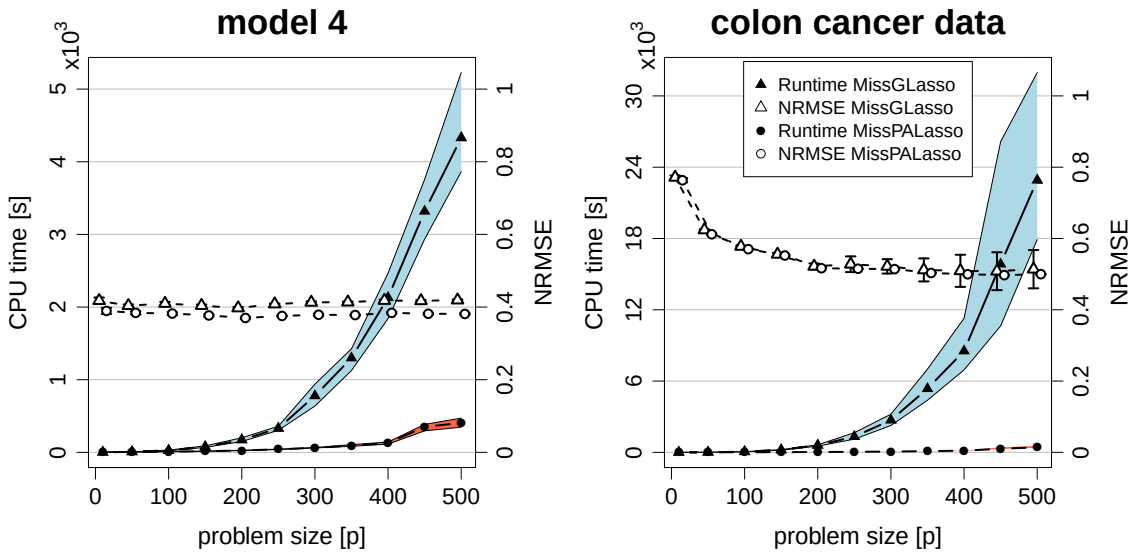

Figure 4: CPU times (filled points, left axis) and NRMSE (hollow points, right axis) vs. problem size p of MissPALasso (circles) and MissGLasso (triangles) in simulation model 4 (left panel) and the colon cancer data (right panel). MissPALasso and MissGLasso are applied on a grid of thirty λ values. The shaded area shows the full range of CPU times over 10 simulation runs. Measurements of NRMSE include standard error bars which are due to their small size ($\sim 10^{-3}$) mostly not visible except for MissGLasso in the real data example.

5. Theory

A key characteristic of pattern alternating maximization (MissPA, Algorithm 1 in Section 3.2) is that the E-Step is only performed on those samples belonging to a single pattern. We already mentioned the close connection to the incremental EM introduced by Neal and Hinton (1998). In fact, if the density of $\mathbf{X}^k$, $k \in \{1, \ldots, s\}$, is denoted by $\mathrm{P}_\Sigma(\mathbf{X}^k) = \prod_{i \in \mathcal{I}_k} p(x_i; \Sigma)$, then the negative variational free energy (Neal and Hinton, 1998; Jordan et al., 1999) equals

$$\mathcal{F}[\Sigma \| \Psi_1, \ldots, \Psi_s] \quad = \quad \sum_{k=1}^{s} \big(\mathbb{E}_{\Psi_k}[\log \mathrm{P}_\Sigma(\mathbf{X}^k) | \mathbf{X}^k_{o_k}] + \mathcal{H}_k[\Psi_k]\big). \tag{10}$$

Here, $\Psi_k = \left(B_{k,m_k|o_k}, \Sigma_{k,m_k|o_k}\right)$ denotes the regression parameter of the latent distribution

$$\mathrm{P}_{\Psi_k}(\mathbf{X}^k_{m_k}|\mathbf{X}^k_{o_k}) \;=\; \prod_{i\in\mathcal{I}_k} p(x_{i,m_k}|x_{i,o_k}; B_{k,m_k|o_k}, \Sigma_{k,m_k|o_k})$$

and $\mathcal{H}_k[\Psi_k] = -\mathbb{E}_{\Psi_k}[\log \mathrm{P}_{\Psi_k}(\mathbf{X}^k_{m_k}|\mathbf{X}^k_{o_k})|\mathbf{X}^k_{o_k}]$ is the entropy. An iterative procedure alternating between maximization of $\mathcal{F}$ with respect to Σ,

$$\begin{aligned}\hat{\Sigma} &= \arg\max_{\Sigma} \mathcal{F}[\Sigma\|\Psi_1,\ldots,\Psi_s] \\ &= \frac{1}{n}\sum_{k=1}^{s} \mathbb{E}_{\Psi_k}[{}^t\mathbf{X}^k\mathbf{X}^k|\mathbf{X}^k_{o_k}] =: \frac{1}{n}\mathcal{T},\end{aligned}$$

and maximizing $\mathcal{F}$ with respect to Ψ_k,

$$\begin{aligned}(\hat{B}_{k,m_k|o_k}, \hat{\Sigma}_{k,m_k|o_k}) &= \arg\max_{\Psi_k} \mathcal{F}[\hat{\Sigma}\|\Psi_1,\ldots,\Psi_s] \\ &= \arg\max_{\Psi_k} \mathbb{E}_{\Psi_k}[\log \mathrm{P}_{\hat{\Sigma}}(\mathbf{X}^k)|\mathbf{X}^k_{o_k}] + \mathcal{H}_k[\Psi_k] \\ &= \left(\mathcal{T}_{m_k,o_k}\mathcal{T}^{-1}_{o_k,o_k}, \frac{1}{n}(\mathcal{T}_{m_k,m_k} - \mathcal{T}_{m_k,o_k}\mathcal{T}^{-1}_{o_k,o_k}\mathcal{T}_{o_k,m_k})\right),\end{aligned}$$

is equivalent to Algorithm 1 with $\mathcal{T}^{-k}$ replaced by $\mathcal{T}$ (see M-Step2 in Section 3.2). Alternating maximization of (10) is a GAM procedure in the sense of Gunawardana and Byrne (2005) for which convergence to a stationary point of the observed log-likelihood can be established easily.

Unfortunately, MissPA does not quite fit into the GAM formulation as it extends the standard EM in an additional manner, namely by using for each pattern a different complete data space (for each pattern k only those samples are augmented which do not belong to pattern k). From this point of view MissPA is related to the SAGE procedure (Fessler and Hero, 1994). To see this, consider Σ in the parameterization $\theta = \left(\Sigma_{o_k}, B_{m_k|o_k}, \Sigma_{m_k|o_k}\right)$ introduced in Section 2. From

$$\mathrm{P}_\theta(\mathbf{X}_{\mathrm{obs}}, \mathbf{X}^{-k}) = \mathrm{P}_\theta(\mathbf{X}_{\mathrm{obs}}|\mathbf{X}^{-k})\mathrm{P}_\theta(\mathbf{X}^{-k})$$

and observing that $\mathrm{P}_\theta(\mathbf{X}_{\mathrm{obs}}|\mathbf{X}^{-k}) = \mathrm{P}_{\Sigma_{o_k}}(\mathbf{X}_{o_k})$ we conclude that $\mathbf{X}^{-k}$ is an admissible hidden-data space with respect to $(B_{m_k|o_k}, \Sigma_{m_k|o_k})$ in the sense of Fessler and Hero (1994). The M-Step of MissPA then maximizes a conditional expectation of the log-likelihood $\log \mathrm{P}_\theta(\mathbf{X}^{-k})$ with respect to the parameters $(B_{m_k|o_k}, \Sigma_{m_k|o_k})$. Different from SAGE is the conditional distribution involved in the expectation: after each M-Step, our algorithm updates only the conditional distribution for a single pattern, consequently we do not need to compute estimates for Σ_{o_k}.

In summary, MissPA has similarities with GAM and SAGE. However, neither of these frameworks fit our purpose. In the next section we provide theory which justifies alternating between complete data spaces *and* incrementally performing the E-Step. In particular, we prove convergence to a stationary point of the observed log-likelihood.

5.1 Convergence Analysis of Missingness Pattern Alternating Maximization

In this section we study the numerical properties of MissPA.

5.1.1 Pattern-Depending Lower Bounds

Denote the density of $\mathbf{X}^k$, $k \in \{1,\ldots,s\}$, by $\mathrm{P}_\Sigma(\mathbf{X}^k) = \prod_{i\in\mathcal{I}_k} p(x_i;\Sigma)$ and define for $k,l \in \{1,\ldots,s\}$

$$\begin{aligned}\mathrm{P}_\Sigma(\mathbf{X}^l_{o_k}) &= \prod_{i\in\mathcal{I}_l} p(x_{i,o_k};\Sigma_{o_k}) \quad \text{and}\\ \mathrm{P}_\Sigma(\mathbf{X}^l_{m_k}|\mathbf{X}^l_{o_k}) &= \prod_{i\in\mathcal{I}_l} p(x_{i,m_k}|x_{i,o_k};B_{m_k|o_k},\Sigma_{m_k|o_k}).\end{aligned}$$

Set $\{\Sigma_l\}_{l\neq k} = (\Sigma_1,\ldots,\Sigma_{k-1},\Sigma_{k+1},\ldots,\Sigma_s)$ and consider for $k = 1,\ldots,s$

$$\mathcal{F}_k[\Sigma_k||\{\Sigma_l\}_{l\neq k}] = \log \mathrm{P}_{\Sigma_k}(\mathbf{X}^k_{o_k}) + \sum_{l\neq k}\big(\mathbb{E}_{\Sigma_l}[\log \mathrm{P}_{\Sigma_k}(\mathbf{X}^l)|\mathbf{X}^l_{o_l}] + \mathcal{H}_l[\Sigma_l]\big).$$

Here $\mathcal{H}_l[\tilde\Sigma] = -\mathbb{E}_{\tilde\Sigma}[\log \mathrm{P}_{\tilde\Sigma}(\mathbf{X}^l_{m_l}|\mathbf{X}^l_{o_l})|\mathbf{X}^l_{o_l}]$ denotes the entropy. Note that $\mathcal{F}_k$ is defined for fixed observed data $\mathbf{X}_{\mathrm{obs}}$. The subscript k highlights the dependence on the pattern k. Furthermore, for fixed $\mathbf{X}_{\mathrm{obs}}$ and fixed k, $\mathcal{F}_k$ is a function in the parameters $(\Sigma_1,\ldots,\Sigma_s)$. As a further tool we write the Kullback-Leibler divergence in the following form:

$$\mathcal{D}_l[\tilde\Sigma||\Sigma] = \mathbb{E}_{\tilde\Sigma}[-\log\big(\mathrm{P}_\Sigma(\mathbf{X}^l_{m_l}|\mathbf{X}^l_{o_l})/\mathrm{P}_{\tilde\Sigma}(\mathbf{X}^l_{m_l}|\mathbf{X}^l_{o_l})\big)|\mathbf{X}^l_{o_l}]. \tag{11}$$

An important property of the Kullback-Leibler divergence is its non-negativity:

$$\begin{aligned}&\mathcal{D}_l[\tilde\Sigma||\Sigma] \geq 0, \quad \text{with equality if and only if}\\ &\mathrm{P}_{\tilde\Sigma}(\mathbf{X}^l_{m_l}|\mathbf{X}^l_{o_l}) = \mathrm{P}_\Sigma(\mathbf{X}^l_{m_l}|\mathbf{X}^l_{o_l}).\end{aligned}$$

A simple calculation shows that

$$\mathbb{E}_{\tilde\Sigma}[\log \mathrm{P}_\Sigma(\mathbf{X}^l)|\mathbf{X}^l_{o_l}] + \mathcal{H}_l[\tilde\Sigma] = -\mathcal{D}_l[\tilde\Sigma||\Sigma] + \log \mathrm{P}_\Sigma(\mathbf{X}^l_{o_l}) \tag{12}$$

and that $\mathcal{F}_k[\Sigma_k||\{\Sigma_l\}_{l\neq k}]$ can be written as

$$\mathcal{F}_k[\Sigma_k||\{\Sigma_l\}_{l\neq k}] = \ell(\Sigma_k;\mathbf{X}_{\mathrm{obs}}) - \sum_{l\neq k}\mathcal{D}_l[\Sigma_l||\Sigma_k]. \tag{13}$$

In particular, for fixed values of $\{\Sigma_l\}_{l\neq k}$, $\mathcal{F}_k[\,\cdot\,||\{\Sigma_l\}_{l\neq k}]$ lower bounds the observed log-likelihood $\ell(\,\cdot\,;\mathbf{X}_{\mathrm{obs}})$ due to the non-negativity of the Kullback-Leibler divergence.

5.1.2 Optimization Transfer to Pattern-Depending Lower Bounds

We give now an alternative description of the MissPA algorithm. In cycle $r+1$ through all patterns, generate $(\Sigma_1^{(r+1)},\ldots,\Sigma_s^{(r+1)})$ given $(\Sigma_1^{(r)},\ldots,\Sigma_s^{(r)})$ according to

$$\Sigma_k^{(r+1)} = \arg\max_\Sigma \mathcal{F}_k[\Sigma||\mathrm{Z}_k^{(r+1)}], \quad k = 1,\ldots,s, \tag{14}$$

with $\mathrm{Z}_k^{(r+1)} = (\Sigma_1^{(r+1)}, \ldots, \Sigma_{k-1}^{(r+1)}, \Sigma_{k+1}^{(r)}, \ldots, \Sigma_s^{(r)})$.

We have

$$\begin{aligned}\mathcal{F}_k[\Sigma||\mathrm{Z}_k^{(r+1)}] &= \log \mathrm{P}_\Sigma(\mathbf{X}_{o_k}^k) + \sum_{l<k}\left(\mathbb{E}_{\Sigma_l^{(r+1)}}[\log \mathrm{P}_\Sigma(\mathbf{X}^l)|\mathbf{X}_{o_l}^l] + \mathcal{H}_l[\Sigma_l^{(r+1)}]\right) \\ &\quad + \sum_{l>k}\left(\mathbb{E}_{\Sigma_l^{(r)}}[\log \mathrm{P}_\Sigma(\mathbf{X}^l)|\mathbf{X}_{o_l}^l] + \mathcal{H}_l[\Sigma_l^{(r)}]\right).\end{aligned}$$

The entropy terms do not depend on the optimization parameter Σ, therefore,

$$\begin{aligned}\mathcal{F}_k[\Sigma||\mathrm{Z}_k^{(r+1)}] &= \text{const} + \log \mathrm{P}_\Sigma(\mathbf{X}_{o_k}^k) + \sum_{l<k}\mathbb{E}_{\Sigma_l^{(r+1)}}[\log \mathrm{P}_\Sigma(\mathbf{X}^l)|\mathbf{X}_{o_l}^l] \\ &\quad + \sum_{l>k}\mathbb{E}_{\Sigma_l^{(r)}}[\log \mathrm{P}_\Sigma(\mathbf{X}^l)|\mathbf{X}_{o_l}^l].\end{aligned}$$

Using the factorization $\log \mathrm{P}_\Sigma(\mathbf{X}^l) = \log \mathrm{P}(\mathbf{X}_{o_k}^l; \Sigma_{o_k}) + \log \mathrm{P}(\mathbf{X}_{m_k}^l|\mathbf{X}_{o_k}^l; B_{m_k|o_k}, \Sigma_{m_k|o_k})$ (for all $l \neq k$), and separate maximization with respect to Σ_{o_k} and $(B_{m_k|o_k}, \Sigma_{m_k|o_k})$ we end up with the expressions from the M-Step of MissPA. Summarizing the above, we have recovered the M-Step as a maximization of $\mathcal{F}_k[\Sigma||\mathrm{Z}_k^{(r+1)}]$ which is a lower bound of the observed log-likelihood. Or in the language of Lange et al. (2000), optimization of $\ell(\,\cdot\,; \mathbf{X}_{\text{obs}})$ is transferred to the surrogate objective $\mathcal{F}_k[\,\cdot\,||\mathrm{Z}_k^{(r+1)}]$.

There is still an important piece missing: In M-Step k of cycle $r+1$ we are maximizing $\mathcal{F}_k[\,\cdot\,||\mathrm{Z}_k^{(r+1)}]$ whereas in the following M-Step $(k+1)$ we optimize $\mathcal{F}_{k+1}[\,\cdot\,||\mathrm{Z}_{k+1}^{(r+1)}]$. In order for the algorithm to make progress, it is essential that $\mathcal{F}_{k+1}[\,\cdot\,||\mathrm{Z}_{k+1}^{(r+1)}]$ attains higher values than its predecessor $\mathcal{F}_k[\,\cdot\,||\mathrm{Z}_k^{(r+1)}]$. In this sense the following proposition is crucial.

Proposition 1 *For $r = 0, 1, 2, \ldots$ we have that*

$$\begin{aligned}&\mathcal{F}_s[\Sigma_s^{(r)}||\mathrm{Z}_s^{(r)}] \leq \mathcal{F}_1[\Sigma_s^{(r)}||\mathrm{Z}_1^{(r+1)}], \quad \textit{and} \\ &\mathcal{F}_k[\Sigma_k^{(r+1)}||\mathrm{Z}_k^{(r+1)}] \leq \mathcal{F}_{k+1}[\Sigma_k^{(r+1)}||\mathrm{Z}_{k+1}^{(r+1)}] \quad \textit{for } k = 1, \ldots, s-1.\end{aligned}$$

Proof. We have,

$$\mathcal{F}_k[\Sigma_k^{(r+1)}||\mathrm{Z}_k^{(r+1)}] = \log \mathrm{P}_{\Sigma_k^{(r+1)}}(\mathbf{X}_{o_k}^k) + \mathbb{E}_{\Sigma_{k+1}^{(r)}}[\log \mathrm{P}_{\Sigma_k^{(r+1)}}(\mathbf{X}^{k+1})|\mathbf{X}_{o_{k+1}}^{k+1}] + \mathcal{H}_{k+1}[\Sigma_{k+1}^{(r)}] + \mathrm{A}$$

and

$$\mathcal{F}_{k+1}[\Sigma_k^{(r+1)}||\mathrm{Z}_{k+1}^{(r+1)}] = \log \mathrm{P}_{\Sigma_k^{(r+1)}}(\mathbf{X}_{o_{k+1}}^{k+1}) + \mathbb{E}_{\Sigma_k^{(r+1)}}[\log \mathrm{P}_{\Sigma_k^{(r+1)}}(\mathbf{X}^k)|\mathbf{X}_{o_k}^k] + \mathcal{H}_k[\Sigma_k^{(r+1)}] + \mathrm{A}$$

where

$$\mathrm{A} = \sum_{l<k}\mathbb{E}_{\Sigma_l^{(r+1)}}[\log \mathrm{P}_{\Sigma_k^{(r+1)}}(\mathbf{X}^l)|\mathbf{X}_{o_l}^l] + \mathcal{H}_l[\Sigma_l^{(r+1)}] + \sum_{l>k+1}\mathbb{E}_{\Sigma_l^{(r)}}[\log \mathrm{P}_{\Sigma_k^{(r+1)}}(\mathbf{X}^l)|\mathbf{X}_{o_l}^l] + \mathcal{H}_l[\Sigma_l^{(r)}].$$

Furthermore, using (12) and noting that $\mathcal{D}_k[\Sigma_k^{(r+1)}||\Sigma_k^{(r+1)}] = 0$, we obtain

$$\mathcal{F}_k[\Sigma_k^{(r+1)}||\mathrm{Z}_k^{(r+1)}] - \mathcal{F}_{k+1}[\Sigma_k^{(r+1)}||\mathrm{Z}_{k+1}^{(r+1)}] = \mathcal{D}_k[\Sigma_k^{(r+1)}||\Sigma_k^{(r+1)}] - \mathcal{D}_{k+1}[\Sigma_{k+1}^{(r)}||\Sigma_k^{(r+1)}]$$
$$= -\mathcal{D}_{k+1}[\Sigma_{k+1}^{(r)}||\Sigma_k^{(r+1)}] \leq 0.$$

Note that equality holds if and only if $\mathrm{P}_{\Sigma_k^{(r+1)}}(\mathbf{X}_{m_{k+1}}^{k+1}|\mathbf{X}_{o_{k+1}}^{k+1}) = \mathrm{P}_{\Sigma_{k+1}^{(r)}}(\mathbf{X}_{m_{k+1}}^{k+1}|\mathbf{X}_{o_{k+1}}^{k+1})$. ■

In light of Proposition 1 it is clear that (14) generates a monotonically increasing sequence of the form:

$$\mathcal{F}_s[\Sigma_s^{(0)}||\mathrm{Z}_s^{(0)}] \leq \mathcal{F}_1[\Sigma_s^{(0)}||\mathrm{Z}_1^{(1)}] \leq \mathcal{F}_1[\Sigma_1^{(1)}||\mathrm{Z}_1^{(1)}] \leq \mathcal{F}_2[\Sigma_1^{(1)}||\mathrm{Z}_2^{(1)}] \leq \mathcal{F}_2[\Sigma_2^{(1)}||\mathrm{Z}_2^{(1)}] \leq \cdots$$
$$\cdots \leq \mathcal{F}_k[\Sigma_k^{(r+1)}||\mathrm{Z}_k^{(r+1)}] \leq \mathcal{F}_{k+1}[\Sigma_k^{(r+1)}||\mathrm{Z}_{k+1}^{(r+1)}] \leq \mathcal{F}_{k+1}[\Sigma_{k+1}^{(r+1)}||\mathrm{Z}_{k+1}^{(r+1)}] \leq \cdots$$

For example, we can deduce that $\{\mathcal{F}_s[\Sigma_s^{(r)}||\mathrm{Z}_s^{(r)}]\}_{r=0,1,2,\ldots}$ is a monotone increasing sequence in r.

5.1.3 Convergence to Stationary Points

Ideally we would like to show that a limit point of the sequence generated by MissPA is a global maximum of $\ell(\cdot;\mathbf{X}_{\mathrm{obs}})$. Unfortunately, this is too ambitious because for general missing data patterns the observed log-likelihood is a non-concave function with several local maxima. Thus, the most we can expect is that our algorithm converges to a stationary point. This is ensured by the following theorem which we prove in the Appendix.

Theorem 2 *Assume that* $\mathcal{K} = \{(\Sigma_1, \ldots, \Sigma_s) : \mathcal{F}_s[\Sigma_s||\Sigma_1, \ldots, \Sigma_{s-1}] \geq \mathcal{F}_s[\Sigma_s^{(0)}||\mathrm{Z}_s^{(0)}]\}$ *is compact. Then every limit point* $\bar{\Sigma}_s$ *of* $\{\Sigma_s^{(r)}\}_{r=0,1,2,\ldots}$ *is a stationary point of* $\ell(\,\cdot\,;\mathbf{X}_{\mathrm{obs}})$.

6. Discussion and Extensions

We presented a novel methodology for maximizing the observed log-likelihood for a multivariate normal data matrix with missing values. Simplified, our algorithm iteratively cycles through the different missingness patterns, performs multivariate regressions of the missing on the observed variables and uses the regression coefficients for partial imputation of the missing values. We argued theoretically and gave numerical examples showing that our procedure is computationally more efficient than the standard EM algorithm. Furthermore, we analyzed the numerical properties using non-standard arguments and proved that solutions of our algorithm converge to stationary points of the observed log-likelihood.

In a high-dimensional setup regularization is achieved by replacing least squares regressions with Lasso analogues. Our proposed algorithm, MissPALasso, is built upon coordinate descent approximation of the corresponding Lasso problem in order to gain speed. On simulated and four real data sets (all from genomics) we demonstrated that MissPALasso outperforms other imputation techniques such as k-nearest neighbors imputation, nuclear norm minimization or a penalized likelihood approach with an ℓ_1-penalty on the inverse covariance matrix.

MissPALasso is a "heuristic" motivated by the aim of having sparse regression coefficients for imputation. It is unclear which objective function is optimized by MissPALasso. The comments of two referees on this point made us think of another way of imposing sparsity in the regression coefficients: Consider the penalized variational free energy

$$-\mathcal{F}[\Sigma\|\Psi_1,\ldots,\Psi_s] + \mathrm{Pen}(\Sigma,\Psi_1,\ldots,\Psi_s), \tag{15}$$

with $\mathcal{F}[\Sigma\|\Psi_1,\ldots,\Psi_s]$ defined in equation (10) and $\mathrm{Pen}(\Sigma,\Psi_1,\ldots,\Psi_s)$ some penalty function. If we take

$$\mathrm{Pen}(\Sigma,\Psi_1,\ldots,\Psi_s) = \lambda\sum_{k=1}^{s}\|B_{k,m_k|o_k}\|_1,$$

then, alternating minimization of (15) with respect to Σ and Ψ_k leads to an algorithm with sparse regression coefficients. This algorithm is different from MissPALasso, in fact, minimizing (15) with respect to $\Sigma_{k,m_k|o_k}$ and $B_{k,m_k|o_k}$ gives $\Sigma_{k,m_k|o_k} = \hat{\Sigma}_{m_k|o_k}$ and $\hat{B}_{k,m_k|o_k}$ satisfies the subgradient equation

$$0 = \left(\Omega_{m_k,m_k}\hat{B}_{k,m_k|o_k} - \Omega_{m_k,o_k}\right){}^t\mathbf{X}^k_{o_k}\mathbf{X}^k_{o_k} + \lambda\Gamma(\hat{B}_{k,m_k|o_k}),$$

where $\Omega = \Sigma^{-1}$ and $\Gamma(x)$ is the subgradient of $|x|$, applied componentwise to the elements of a matrix. We do not currently have knowledge of the theoretical or empirical properties of such an algorithm.

In this manuscript we only considered applications to microarray data sets. Our approach is not specifically designed for microarrays and is potentially very useful for many other high-dimensional applications: examples include mass spectrometry-based proteomics, climate field reconstructions and image analysis in cosmology (Karpievitch et al., 2009; Schneider, 2001; Starck and Bobin, 2010). We note that different imputation methods can be beneficial depending on the application context. For example estimating missing entries in gene expression data is a separate problem from dealing with missing values in recommender systems: the Netflix data set (Bennett and Lanning, 2007) involves "large n *and* large p" (480'000 customers, 17'000 movies) with about 98% of the movie ratings missing, in contrast, microarrays have the typical "large p, small n" form and have a much smaller fraction of the values missing. We think that the formulation (15) of our pattern alternating maximization framework is very compelling and can motivate new and efficient algorithms for missing data imputation with application-specific regularization strategies.

Acknowledgments

N.S. acknowledges financial support from Novartis International AG, Basel, Switzerland.

Appendix A.

In this appendix we prove Theorem 2. First, note that the sequence $\{(\Sigma_1^{(r)},\ldots,\Sigma_s^{(r)})\}_{r=0,1,2,\ldots}$ lies in the compact set $\mathcal{K}$. Now, let $\Sigma_s^{(r_j)}$ be a subsequence converging to $\bar{\Sigma}_s$ as $j\to\infty$. By

invoking compactness, we can assume w.l.o.g (by restricting to a subsequence) that

$$(\Sigma_1^{(r_j)}, \ldots, \Sigma_s^{(r_j)}) \to (\bar{\Sigma}_1, \ldots, \bar{\Sigma}_s).$$

As a direct consequence of the monotonicity of the sequence $\{\mathcal{F}_s[\Sigma_s^{(r)}||\mathrm{Z}_s^{(r)}]\}_{r=0,1,2,\ldots}$ we obtain

$$\lim_r \mathcal{F}_s[\Sigma_s^{(r)}||\mathrm{Z}_s^{(r)}] = \mathcal{F}_s[\bar{\Sigma}_s||\bar{\Sigma}_1, \ldots, \bar{\Sigma}_{s-1}] =: \bar{\mathcal{F}}.$$

From (14) and Proposition 1, for $k = 1, \ldots, s-1$ and $r = 0, 1, 2, \ldots$, the following "sandwich"-formulae hold:

$$\mathcal{F}_s[\Sigma_s^{(r)}||\mathrm{Z}_s^{(r)}] \leq \mathcal{F}_1[\Sigma_s^{(r)}||\mathrm{Z}_1^{(r+1)}] \leq \mathcal{F}_1[\Sigma_1^{(r+1)}||\mathrm{Z}_1^{(r+1)}] \leq \mathcal{F}_s[\Sigma_s^{(r+1)}||\mathrm{Z}_s^{(r+1)}],$$
$$\mathcal{F}_s[\Sigma_s^{(r)}||\mathrm{Z}_s^{(r)}] \leq \mathcal{F}_{k+1}[\Sigma_k^{(r+1)}||\mathrm{Z}_{k+1}^{(r+1)}] \leq \mathcal{F}_{k+1}[\Sigma_{k+1}^{(r+1)}||\mathrm{Z}_{k+1}^{(r+1)}] \leq \mathcal{F}_s[\Sigma_s^{(r+1)}||\mathrm{Z}_s^{(r+1)}].$$

As a consequence we have for $k = 1, \ldots, s-1$

$$\lim_r \mathcal{F}_1[\Sigma_s^{(r)}||\mathrm{Z}_1^{(r+1)}] = \lim_r \mathcal{F}_1[\Sigma_1^{(r+1)}||\mathrm{Z}_1^{(r+1)}] = \bar{\mathcal{F}} \tag{16}$$

and

$$\lim_r \mathcal{F}_{k+1}[\Sigma_k^{(r+1)}||\mathrm{Z}_{k+1}^{(r+1)}] = \lim_r \mathcal{F}_{k+1}[\Sigma_{k+1}^{(r+1)}||\mathrm{Z}_{k+1}^{(r+1)}] = \bar{\mathcal{F}}. \tag{17}$$

Now consider the sequence $(\Sigma_1^{(r_j+1)}, \ldots, \Sigma_s^{(r_j+1)})$. By compactness of $\mathcal{K}$ this sequence converges w.l.o.g to some $(\Sigma_1^*, \ldots, \Sigma_s^*)$. We now show by induction that

$$\bar{\Sigma}_s = \Sigma_1^* = \ldots = \Sigma_s^*.$$

From the 1st M-Step of cycle $r_j + 1$ we have

$$\mathcal{F}_1[\Sigma_1^{(r_j+1)}||\mathrm{Z}_1^{(r_j+1)}] \geq \mathcal{F}_1[\Sigma||\mathrm{Z}_1^{(r_j+1)}] \quad \text{for all} \quad \Sigma.$$

Taking the limit $j \to \infty$ we get:

$$\mathcal{F}_1[\Sigma_1^*||\{\bar{\Sigma}_l\}_{l>1}] \geq \mathcal{F}_1[\Sigma||\{\bar{\Sigma}_l\}_{l>1}] \quad \text{for all} \quad \Sigma.$$

In particular, Σ_1^* is the (unique) maximizer of $\mathcal{F}_1[\,\cdot\,||\{\bar{\Sigma}_l\}_{l>1}]$. Assuming $\Sigma_1^* \neq \bar{\Sigma}_s$ would imply

$$\mathcal{F}_1[\Sigma_1^*||\{\bar{\Sigma}_l\}_{l>1}] > \mathcal{F}_1[\bar{\Sigma}_s||\{\bar{\Sigma}_l\}_{l>1}].$$

But this contradicts $\mathcal{F}_1[\Sigma_1^*||\{\bar{\Sigma}_l\}_{l>1}] = \mathcal{F}_1[\bar{\Sigma}_s||\{\bar{\Sigma}_l\}_{l>1}] = \bar{\mathcal{F}}$, which holds by (16). Therefore we obtain $\Sigma_1^* = \bar{\Sigma}_s$.

Assume that we have proven $\Sigma_1^* = \ldots = \Sigma_k^* = \bar{\Sigma}_s$. We will show that $\Sigma_{k+1}^* = \bar{\Sigma}_s$. From the k+1st M-Step in cycle $r_j + 1$:

$$\mathcal{F}_{k+1}[\Sigma_{k+1}^{(r_j+1)}||\mathrm{Z}_{k+1}^{(r_j+1)}] \geq \mathcal{F}_{k+1}[\Sigma||\mathrm{Z}_{k+1}^{(r_j+1)}] \quad \text{for all} \quad \Sigma.$$

Taking the limit for $j \to \infty$, we conclude that Σ_{k+1}^* is the (unique) maximizer of

$$\mathcal{F}_{k+1}[\,\cdot\,||\{\Sigma_l^*\}_{l<k+1}, \{\bar{\Sigma}_l\}_{l>k+1}].$$

From (17),

$$\mathcal{F}_{k+1}[\Sigma_{k+1}^*||\{\Sigma_l^*\}_{l<k+1}, \{\bar{\Sigma}_l\}_{l>k+1}] = \mathcal{F}_{k+1}[\Sigma_k^*||\{\Sigma_l^*\}_{l<k+1}, \{\bar{\Sigma}_l\}_{l>k+1}] = \bar{\mathcal{F}},$$

and therefore Σ_{k+1}^* must be equal to Σ_k^*. By induction we have $\Sigma_k^* = \bar{\Sigma}_s$ and we have proven that $\Sigma_{k+1}^* = \bar{\Sigma}_s$ holds.

Finally, we show stationarity of $\bar{\Sigma}_s$. Invoking (13) we can write

$$\mathcal{F}_s[\Sigma||\bar{\Sigma}_s, \ldots, \bar{\Sigma}_s] = \ell(\Sigma; \mathbf{X}_{\text{obs}}) - \sum_{l=1}^{s-1} \mathcal{D}_l[\bar{\Sigma}_s||\Sigma].$$

Note that

$$\frac{\partial}{\partial \Sigma} \mathcal{D}_l[\bar{\Sigma}_s||\Sigma]\Big|_{\bar{\Sigma}_s} = 0.$$

Furthermore, as $\Sigma_s^{(r_j+1)}$ maximizes $\mathcal{F}_s[\Sigma||\Sigma_1^{(r_j+1)}, \ldots, \Sigma_{s-1}^{(r_j+1)}]$, we get in the limit as $j \to \infty$

$$\frac{\partial}{\partial \Sigma} \mathcal{F}_s[\Sigma||\bar{\Sigma}_s, \ldots, \bar{\Sigma}_s]\Big|_{\bar{\Sigma}_s} = \frac{\partial}{\partial \Sigma} \mathcal{F}_s[\Sigma||\Sigma_1^*, \ldots, \Sigma_{s-1}^*]\Big|_{\Sigma_s^*} = 0.$$

Therefore, we conclude that $\frac{\partial}{\partial \Sigma} \ell(\Sigma; \mathbf{X}_{\text{obs}})\Big|_{\bar{\Sigma}_s} = 0.$ ■

References

T. Aittokallio. Dealing with missing values in large-scale studies: microarray data imputation and beyond. *Briefings in Bioinformatics*, 11(2):253–264, 2010.

A. Alizadeh, M. Eisen, R. Davis, C. Ma, I. Lossos, A. Rosenwald, J. Boldrick, H. Sabet, T. Tran, X. Yu, J. Powell, L. Yang, G. Marti, T. Moore, J. Hudson, L. Lu, D. Lewis, R. Tibshirani, G. Sherlock, W. Chan, T. Greiner, D. Weisenburger, J. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. Byrd, D. Botstein, P. Brown, and L. Staudt. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

G. Allen and R. Tibshirani. Transposable regularized covariance models with an application to missing data imputation. *Annals of Applied Statistics*, 4(2):764–790, 2010.

U. Alon, N. Barkai, D. Notterman, K. Gishdagger, S. Ybarradagger, D. Mackdagger, and A. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, 96(12):6745–6750, 1999.

J. Bennett and S. Lanning. The Netflix prize. In *Proceedings of KDD Cup and Workshop*, San Jose, 2007.

J.-F. Cai, E. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

E. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

E. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5), 2010.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

J. Fessler and A. Hero. Space-alternating generalized Expectation-Maximization algorithm. *IEEE Transactions on Signal Processing*, 42(11):2664–2677, 1994.

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

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

A. Gunawardana and W. Byrne. Convergence theorems for generalized alternating minimization procedures. *Journal of Machine Learning Research*, 6:2049–2073, 2005.

M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

J. Josse, J. Pagès, and F. Husson. Multiple imputation in principal component analysis. *Advances in Data Analysis and Classification*, 5(3):231–246, 2011.

Y. V. Karpievitch, J. Stanley, T. Taverner, J. Huang, J. N. Adkins, C. Ansong, F. Heffron, T. O. Metz, W.-J. Qian, H. Yoon, R. D. Smith, and A. R. Dabney. A statistical framework for protein quantitation in bottom-up MS-based proteomics. *Bioinformatics*, 25(16):2028–2034, 2009.

R. Keshavan, S. Oh, and A. Montanari. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6), 2010.

K. Lange, D. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1):1–20, 2000.

R. Little and D. Rubin. *Statistical Analysis with Missing Data*. Series in Probability and Mathematical Statistics, Wiley, 1987.

P.-L. Loh and M. J. Wainwright. High-dimensional regression with noisy and missing data: Provable guarantees with nonconvexity. *Annals of Statistics*, 40(3):1637–1664, 2012.

R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 99:2287–2322, 2010.

R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1998.

S. Ng and G. McLachlan. On the choice of the number of blocks with the incremental EM algorithm for the fitting of normal mixtures. *Statistics and Computing*, 13(1):45–55, 2003.

S. Nowlan. *Soft Competitive Adaptation: Neural Network Learning Algorithms based on Fitting Statistical Mixtures*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1991.

S. Oba, M.-A. Sato, I. Takemasa, M. Monden, K.-I. Matsubara, and S. Ishii. A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19(16): 2088–2096, 2003.

M. Rosenbaum and A. Tsybakov. Sparse recovery under matrix uncertainty. *Annals of Statistics*, 38(5):2620–2651, 2010.

J. Schafer. *Analysis of Incomplete Multivariate Data*. Monographs on Statistics and Applied Probability 72, Chapman and Hall, 1997.

T. Schneider. Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values. *Journal of Climate*, 14(5):853–871, 2001.

P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9 (12):3273–97, 1998.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. The MIT Press, 2nd edition, 2000.

N. Städler and P. Bühlmann. Missing values: sparse inverse covariance estimation and an extension to sparse regression. *Statistics and Computing*, 22(1):219–235, 2012.

J.-L. Starck and J. Bobin. Astronomical data analysis and sparsity: From wavelets to compressed sensing. *Proceedings of the IEEE*, 98(6):1021–1030, 2010.

B. Thiesson, C. Meek, and D. Heckerman. Accelerating EM for large databases. *Machine Learning*, 45(3):279–299, 2001.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

A. Wille, P. Zimmermann, E. Vranova, A. Fürholz, O. Laule, S. Bleuler, L. Hennig, A. Prelic, P. Rohrvon , L. Thiele, E. Zitzler, W. Gruissem, and P. Bühlmann. Sparse graphical Gaussian modeling of the isoprenoid gene network in arabidopsis thaliana. *Genome Biology*, 5(11), 2004.