# Relevance Singular Vector Machine for Low-rank Matrix Reconstruction 

Martin Sundin, Student Member, IEEE, Cristian R. Rojas Member, IEEE, Magnus Jansson, Senior Member, IEEE, and Saikat Chatterjee, Member, IEEE


#### Abstract

We develop Bayesian learning methods for lowrank matrix reconstruction and completion from linear measurements. For under-determined systems, the developed methods reconstruct low-rank matrices when neither the rank nor the noise power is known a-priori. We derive relations between the proposed Bayesian models and low-rank promoting penalty functions. The relations justify the use of Kronecker structured covariance matrices in a Gaussian based prior. In the methods, we use expectation-maximization to learn the model parameters. The performance of the methods is evaluated through extensive numerical simulations on synthetic and real data.


## I. Introduction

Reconstruction of a low-rank matrix from few measurements is a challenging problem. The low-rank matrix reconstruction (LRMR) problem is inherently under-determined and have been receiving considerable attention due to its generality over popular sparse reconstruction problems along with many application scopes [1]-[6]. A special case of LRMR is the popular low-rank matrix completion problem [1], [2], [5], [7]-[20]. Consider a matrix $\mathbf{X} \in \mathbb{R}^{p \times q}$ with $\operatorname{rank}(\mathbf{X}) \ll \min (p, q)$. In the LRMR setup [1]-[6], the low rank matrix $\mathbf{X}$ is measured by

$$
\begin{equation*}
\mathbf{y}=\mathcal{A}(\mathbf{X})+\mathbf{n} \in \mathbb{R}^{m \times 1} \tag{1}
\end{equation*}
$$

where $\mathcal{A}: \mathbb{R}^{p \times q} \rightarrow \mathbb{R}^{m}$ can be viewed as a known sampling (or sensing, or measurement) operator representing the LRMR sensing process and $\mathbf{n}$ is additive measurement noise (typically assumed to be zero-mean Gaussian with covariance $\operatorname{Cov}(\mathbf{n})=$ $\beta^{-1} \mathbf{I}_{m}$ ). The sensing operator $\mathcal{A}$ is linear and has the form:

$$
\mathcal{A}(\mathbf{X})=\left[\begin{array}{c}
\left\langle\mathbf{X}, \mathbf{A}_{1}\right\rangle  \tag{2}\\
\vdots \\
\left\langle\mathbf{X}, \mathbf{A}_{m}\right\rangle
\end{array}\right],
$$

where $\left\{\mathbf{A}_{k}\right\}_{k=1}^{m}$ is known and $\left\langle\mathbf{X}, \mathbf{A}_{k}\right\rangle \triangleq \operatorname{trace}\left(\mathbf{A}_{k}^{\top} \mathbf{X}\right)$. Hence we can write (1) in the following form

$$
\begin{equation*}
\mathbf{y}=\mathbf{A} \operatorname{vec}(\mathbf{X})+\mathbf{n}, \tag{3}
\end{equation*}
$$

where $\mathbf{A} \in \mathbb{R}^{m \times p q}$ is the effective linear measurement matrix and $\operatorname{vec}(\cdot)$ is the standard vectorization operator. In the case of low-rank matrix completion [1], [2], A has a special structure

[^0]where each row of $\mathbf{A}$ contains a (single) 1 and all other elements of the row vector are zeros, with the constraint that the rows of $\mathbf{A}$ are linearly independent; such a special $\mathbf{A}$ works as an element selection operator. For $\mathbf{X} \in \mathbb{R}^{p \times q}$ and $\operatorname{rank}(\mathbf{X})=r$, the number of free scalars in $\mathbf{X}$ is $p r+q r-r^{2}$ [1]. With $p r+q r-r^{2}<m<p q$, the setup is underdetermined and the task is the reconstruction (or estimation) of $\mathbf{X}$ from $\mathbf{y}$. To deal with the underdetermined setup, a typical and much used strategy is to use a regularization in the reconstruction cost function. Regularization brings in the information about low rank priors. A typical estimator is
\[

$$
\begin{equation*}
\hat{\mathbf{X}}=\arg \min _{\mathbf{X}}\left\{\beta\|\mathbf{y}-\mathbf{A} \operatorname{vec}(\mathbf{X})\|_{2}^{2}+g(\mathbf{X})\right\}, \tag{4}
\end{equation*}
$$

\]

where $\beta>0$ is a parameter and $g(\cdot)$ is a fixed penalty function that inculcates regularization by promoting low rank in $\hat{\mathbf{X}}$. Common low-rank promoting penalties in the literature [1], [3], [7] are

$$
\begin{array}{lr}
g(\mathbf{X})=\|\mathbf{X}\|_{*}=\operatorname{tr}\left(\left(\mathbf{X X}^{\top}\right)^{1 / 2}\right), & \text { (nuclear norm) } \\
g(\mathbf{X})=\operatorname{tr}\left(\left(\mathbf{X X}^{\top}\right)^{s / 2}\right), & \text { (Schatten s-norm) } \\
g(\mathbf{X})=\log \left|\mathbf{X X}^{\top}+\epsilon \mathbf{I}_{p}\right|, & (\text { (log-determinant penalty) }
\end{array}
$$

where $\operatorname{tr}(\cdot)$ denotes the matrix trace, $|\cdot|$ denotes determinant, $s>0, \epsilon>0, \mathbf{I}_{p}$ denotes the $p \times p$ identity matrix and $(\cdot)^{c}$ denotes the $c$ 'th symmetric power of a semidefinite matrix. Typically $0<s \leq 1$ is used to promote low-rank. Among the above three penalties, we mention that the nuclear norm penalty is a convex function.
Following the literature, LRMR algorithms can be categorized in three main types: convex optimization based [1]-[3], [8], [9], [19], [21], greedy solvers [4], [7], [10], [22], [23] and Bayesian learning methods [11]-[18]. Many of these existing algorithms are motivated from analogous algorithms used for standard sparse reconstruction problems, such as compressed sensing (in compressed sensing, we use a sparse vector in (3) instead of $\operatorname{vec}(\mathbf{X})$. Using convex optimization we can solve (4) when $g(\mathbf{X})$ is convex (e.g. the nuclear norm). Further, greedy algorithms, such as iteratively reweighted least squares [7] solves (4) using algebraic approximations. While convex optimization and greedy solvers are popular, they often need more prior information than knowledge about the structure of the signal under reconstruction; for example, convex optimization algorithms need information about the strength of the measurement noise $\mathbf{n}$ to fix the parameter $\beta$, and greedy algorithms need information about rank of $\mathbf{X}$. In absence of such a-priori information, Bayesian learning is a preferred strategy. Bayesian learning methods are typically capable of
estimating the necessary parameters from data. In Bayesian learning we evaluate the posterior $p(\mathbf{X} \mid \mathbf{y})$ with the knowledge of prior $p(\mathbf{X})$. If $\mathbf{X}$ has a prior distribution $p(\mathbf{X}) \propto e^{-\frac{1}{2} g(\mathbf{X})}$ and the noise is distributed as $\mathbf{n} \sim \mathcal{N}\left(\mathbf{0}, \beta^{-1} \mathbf{I}_{m}\right)$, then the maximum-a-posteriori (MAP) estimation of $\mathbf{X}$ takes the form of (4). As the MAP estimation requires additional information (such as $\beta$ ), type II estimators are often more useful (following [24], type II estimators are also known as empirical Bayes [25], [26], generalized maximum likelihood, evidence approximation, etc.). Type II estimation techniques use hyperparameters with appropriate prior distributions, and provide an approximation of the posterior distribution of the variable of interest. While for sparse reconstruction problems, Bayesian learning via type II estimation in the form of relevance vector machine (RVM) [27], [28] and sparse Bayesian learning (SBL) [29], [30] have gained significant popularity, the endeavor to design type II estimation algorithms for LRMR is found to be limited. To the authors' knowledge, a major difficulty arises due to a non-trivial task of defining a low-rank promoting prior. There are some efforts in the literature, discussed in section I-A.

Our objective in this paper is to develop new type II estimation methods for LRMR. Borrowing ideas from type II estimation techniques for sparse reconstruction, such as the RVM and SBL algorithms, we model a low-rank matrix by a multiplication of precision matrices and an i.i.d. Gaussian matrix. The use of precision matrices helps to realize low-rank. The main contributions of this paper are as follows.

- We introduce one-sided and two-sided precision matrix based models for priors of $\mathbf{X}$.
- We show how the Schatten s-norm and log-determinant penalty functions are related to priors of the precision matrices, establishing a connection with the MAP estimator (4).
- We design type II estimation algorithms based on expectation-maximization. The new estimation algorithms are referred to as relevance singular vector machine (RSVM) due to conceptual and technical similarities with the RVM.
The proposed RSVM algorithms are compared to several existing methods via simulations, for synthetic and real data (MovieLens data [20]).


## A. Relevant Prior Work

One of the most popular convex optimization based algorithm for LRMR is nuclear norm minimization [1]-[3], [8], [9], [21], defined as follows

$$
\begin{equation*}
\hat{\mathbf{X}}=\arg \min _{\mathbf{X}}\|\mathbf{X}\|_{*}, \text { s.t. }\|\mathbf{y}-\mathbf{A} \operatorname{vec}(\mathbf{X})\|_{2} \leq \delta \tag{5}
\end{equation*}
$$

where $\delta$ is the allowable model fit error due to measurement noise $\mathbf{n}$. A standard choice is $\delta=\beta^{-1} \sqrt{m+\sqrt{8 m}}$ according to the suggestion in [31]. Henceforth the nuclear-norm minimization estimator (5) is referred to as NN. We note that NN requires a-priori knowledge of $\delta$. The NN estimator (5) can be easily represented in the form of (4) via Lagrangian (where $g(\mathbf{X})=\|\mathbf{X}\|_{*}$ ). We will show in section II-A that there exists a prior for which RSVM is equivalent to the MAP
estimator (4) with a nuclear norm regularization. Another convex optimization based method [19] uses a weighted trace norm that is exclusively used for low-rank matrix completion.

In Bayesian learning algorithms for LRMR, a prominent approach is to use variational Bayes techniques [11], [15], [16]. Typically the variational Bayes approach uses a factorization based matrix model as $\mathbf{X}=\mathbf{F} \mathbf{B}^{\top}$ and then uses appropriate priors for $\mathbf{F}$ and $\mathbf{B}$. In [11], $\mathbf{F}$ and $\mathbf{B}$ use (block) sparsity inducing priors to promote low-rank in $\mathbf{X}$. A similar approach was adopted in [15]. By using Gaussian priors for $\mathbf{F}$ and $\mathbf{B}$, probabilistic matrix factorization (PMF) [17] formulates the inference problem as an optimization problem. Next, sampling schemes, such as Markov Chain Monte Carlo (MCMC) methods, have been used to approximate the posterior distribution [13], [18]. All methods have varying levels of trade-off between complexity and performance. Further, to the best of authors' knowledge, except [16], there is no effort to relate the priors of $\mathbf{F}$ and $\mathbf{B}$ to $g(\mathbf{X})$ in (4) (and in turn to low-rank promoting distributions $p(\mathbf{X})$ ). In this paper, we establish such kind of relations for the proposed method. We also evaluate the performance of the algorithms for LRMR vis-a-vis NN and a generalization of the variational Bayes method of [11]. Further, for low-rank matrix completion, we also compared with the variational Bayes method of [15], PMF [17] and the weighted trace norm (WTN) method [19].

## II. ONE-SIDED PRECISION BASED MODEL

In this section, we propose suitable priors for low-rank matrices and establish a relation with the MAP estimator (4). Typically coefficients of each column vector of a random lowrank matrix $\mathbf{X}$ are highly correlated. To model the correlation, we set

$$
\begin{equation*}
\mathbf{X}=\boldsymbol{\alpha}^{-1 / 2} \mathbf{U} \tag{6}
\end{equation*}
$$

where the components of $\mathbf{U} \in \mathbb{R}^{p \times q}$ are i.i.d. $\mathcal{N}(0,1)$ and $\boldsymbol{\alpha} \in \mathbb{R}^{p \times p}$ is a positive definite precision matrix (the distribution of $\boldsymbol{\alpha}$ will be described later). Note that, if $\boldsymbol{\alpha}$ is given then each column vector of $\mathbf{X}$ has a distribution $\mathcal{N}\left(\mathbf{0}, \boldsymbol{\alpha}^{-1}\right)$. The factorization in (6) is equivalent to using a matrix variate Gaussian distribution [16], [32] as follows

$$
\begin{equation*}
p(\mathbf{X} \mid \boldsymbol{\alpha})=\frac{|\boldsymbol{\alpha}|^{q / 2}}{(2 \pi)^{p q / 2}} \exp \left(-\frac{1}{2} \operatorname{tr}\left(\mathbf{X}^{\top} \boldsymbol{\alpha} \mathbf{X}\right)\right) \tag{7}
\end{equation*}
$$

Denoting $\mathbf{Z}=\mathbf{X} \mathbf{X}^{\top}$, we find that

$$
\begin{align*}
p(\mathbf{X}) & =\int_{\boldsymbol{\alpha} \succ \mathbf{0}} p(\mathbf{X} \mid \boldsymbol{\alpha}) p(\boldsymbol{\alpha}) d \boldsymbol{\alpha} \\
& =\int_{\boldsymbol{\alpha} \succ \mathbf{0}} e^{-\frac{1}{2} \operatorname{tr}(\boldsymbol{\alpha} \mathbf{Z})} \frac{|\boldsymbol{\alpha}|^{q / 2}}{(2 \pi)^{p q / 2}} p(\boldsymbol{\alpha}) d \boldsymbol{\alpha} \tag{8}
\end{align*}
$$

where we used that $\operatorname{tr}\left(\mathbf{X}^{\top} \boldsymbol{\alpha} \mathbf{X}\right)=\operatorname{tr}\left(\boldsymbol{\alpha} \mathbf{X X}^{\top}\right)=\operatorname{tr}(\boldsymbol{\alpha} \mathbf{Z})$. This gives that $p(\mathbf{X})=C e^{-\frac{1}{2} g(\mathbf{X})}=C^{\prime} e^{-\frac{1}{2} \tilde{g}(\mathbf{Z})}$ for some functions $g(\cdot), \tilde{g}(\cdot)$ and constants $C, C^{\prime}$. Estimating $\mathbf{X}$ with the marginal distribution $p(\mathbf{X})$ thus leads to the MAP estimator (4) with $g(\mathbf{X})=\tilde{g}(\mathbf{Z})+$ constant.

## A. Relation between priors and the MAP estimation

The motivation of investigating the relation is to bring connection between the priors $-p(\mathbf{X})$ and $p(\boldsymbol{\alpha})-$ and the MAP estimator (4). From (8), we note that $p(\mathbf{X})$ is the matrix Laplace transform of $|\boldsymbol{\alpha}|^{q / 2} p(\boldsymbol{\alpha}) /(2 \pi)^{p q / 2}$ at $\mathbf{Z} / 2$ [33], which establishes the relation between priors. For ease of readability, we report the standard definition of matrix Laplace transform and its inverse in Appendix A. This gives us that we can calculate $p(\boldsymbol{\alpha})$ from $p(\mathbf{X})=C^{\prime} e^{-\frac{1}{2} \tilde{g}(\mathbf{Z})}$ by the inverse Laplace transform [33] as follows

$$
\begin{equation*}
p(\boldsymbol{\alpha}) \propto|\boldsymbol{\alpha}|^{-q / 2} \int_{\operatorname{Re} \mathbf{Z}=\boldsymbol{\alpha}_{*}} e^{\frac{1}{2} \operatorname{tr}(\boldsymbol{\alpha} \mathbf{Z})} e^{-\frac{1}{2} \tilde{g}(\mathbf{Z})} d \mathbf{Z} \tag{9}
\end{equation*}
$$

where the integral is taken over all symmetric matrices $\mathbf{Z} \in$ $\mathbb{C}^{p \times p}$ such that $\operatorname{Re} \mathbf{Z}=\boldsymbol{\alpha}_{*}$ where $\boldsymbol{\alpha}_{*}$ is a real matrix such that the contour path of integration is in the region of convergence of the integrand. While the Laplace transform characterizes the exact relation between priors, the computation is nontrivial and often analytically intractable. In practice, a standard approach is to use the Laplace approximation [34] where typically the mode of the distribution under approximation is found first and then a Gaussian distribution is modeled around that mode. Let us write $p(\boldsymbol{\alpha})$ as $p(\boldsymbol{\alpha}) \propto e^{-\frac{1}{2} K(\boldsymbol{\alpha})}$; then the Laplace approximation becomes

$$
\begin{aligned}
\tilde{g}(\mathbf{Z})= & \min _{\boldsymbol{\alpha} \succ \mathbf{0}}\{\operatorname{tr}(\boldsymbol{\alpha} \mathbf{Z})-q \log |\boldsymbol{\alpha}|+K(\boldsymbol{\alpha})\} \\
& -\log |\mathbf{H}|+\text { constant }
\end{aligned}
$$

where $\mathbf{H}$ is the Hessian of $\operatorname{tr}(\boldsymbol{\alpha} \mathbf{Z})-q \log |\boldsymbol{\alpha}|+K(\boldsymbol{\alpha})$ evaluated at the minima. The derivation of the Laplace approximation is shown in Appendix B.

Denoting $\tilde{K}(\boldsymbol{\alpha})=q \log |\boldsymbol{\alpha}|-K(\boldsymbol{\alpha})$ and neglecting the Hessian we get

$$
\tilde{g}(\mathbf{Z})=\min _{\boldsymbol{\alpha} \succ \mathbf{0}}\{\operatorname{tr}(\boldsymbol{\alpha} \mathbf{Z})-\tilde{K}(\boldsymbol{\alpha})\}
$$

where we absorbed the constants into the normalization factor of $p(\mathbf{X})$. We find that $\tilde{g}(\boldsymbol{Z})$ is the concave conjugate of $\tilde{K}(\boldsymbol{\alpha})$ [21]. Hence, for a given $\tilde{g}(\boldsymbol{Z})$ we can recover $\tilde{K}(\boldsymbol{\alpha})$ as

$$
\begin{equation*}
\tilde{K}(\boldsymbol{\alpha})=\min _{\boldsymbol{Z} \succ \mathbf{0}}\{\operatorname{tr}(\boldsymbol{\alpha} \mathbf{Z})-\tilde{g}(\boldsymbol{Z})\} \tag{10}
\end{equation*}
$$

if $\tilde{K}(\boldsymbol{\alpha})$ is concave (which holds under the assumption that $K(\boldsymbol{\alpha})$ is convex). Further, we can find $K(\boldsymbol{\alpha})$ from $\tilde{K}(\boldsymbol{\alpha})$ and solve for the prior $p(\boldsymbol{\alpha}) \propto e^{-\frac{1}{2} K(\boldsymbol{\alpha})}$. Using the concave conjugate relation (10), we now deal with the task of finding appropriate functions $K(\boldsymbol{\alpha})$ for two examples of low-rank promoting penalty functions, as follows.

1) For Schatten $s$-norm: The Schatten $s$-norm based penalty function is $g(\mathbf{X})=\operatorname{tr}\left(\left(\mathbf{X X}^{\top}\right)^{s / 2}\right)$. We here use a regularized Schatten $s$-norm based penalty function as

$$
\begin{equation*}
g(\mathbf{X})=\operatorname{tr}\left(\left(\mathbf{X X}^{\top}+\epsilon \mathbf{I}_{\mathbf{p}}\right)^{s / 2}\right) \tag{11}
\end{equation*}
$$

where the use of $\epsilon>0$ helps to bring numerical stability to the algorithms in Section IV. For the penalty function (11), we find $K(\boldsymbol{\alpha})$ as

$$
\begin{equation*}
K(\boldsymbol{\alpha})=C_{s} \operatorname{tr}\left(\boldsymbol{\alpha}^{-\frac{s}{2-s}}\right)+q \log |\boldsymbol{\alpha}|+\epsilon \operatorname{tr}(\boldsymbol{\alpha}) \tag{12}
\end{equation*}
$$

where $C_{s}=\frac{2-s}{s}\left(\frac{2}{s}\right)^{-\frac{s}{2-s}}$. We thus find that

$$
p(\boldsymbol{\alpha})=\frac{D_{p, q, s}}{|\boldsymbol{\alpha}|^{q / 2}} e^{-\frac{1}{2} C_{s} \operatorname{tr}\left(\boldsymbol{\alpha}^{-\frac{s}{2-s}}\right)-\frac{\epsilon}{2} \operatorname{tr}(\boldsymbol{\alpha})}
$$

where $D_{p, q, s}$ is a normalization constant. The derivation of (12) is given in Appendix E. For $s=1$ and $\epsilon=0$ we note that $g(\mathbf{X})$ becomes the nuclear norm of $\mathbf{X}$.
2) Log-determinant penalty: For the log-determinant based penalty function (a scaled version)

$$
\begin{equation*}
g(\mathbf{X})=\nu \log \left|\mathbf{X} \mathbf{X}^{\top}+\epsilon \mathbf{I}_{p}\right| \tag{13}
\end{equation*}
$$

where $\nu>q-2$ is a real number, we find $K(\boldsymbol{\alpha})$ as

$$
\begin{equation*}
K(\boldsymbol{\alpha})=\epsilon \operatorname{tr}(\boldsymbol{\alpha})+(q-\nu) \log |\boldsymbol{\alpha}| \tag{14}
\end{equation*}
$$

The derivation of (14) is shown in Appendix F. As $p(\boldsymbol{\alpha}) \propto e^{-\frac{1}{2} K(\boldsymbol{\alpha})}$, we find that the precision matrix $\boldsymbol{\alpha}$ is Wishart distributed. We also find that $p(\mathbf{X}) \propto e^{-\frac{1}{2} \tilde{g}\left(\mathbf{X} \mathbf{X}^{\top}\right)}$ is a matrix $t$-distribution [32].
At this point we mention that the model (6) is based on matrix factorization. The recent method of [16] also used a matrix factorization model where Gaussian priors are used for all factorized matrices. The important aspect of our work is that we used a Gaussian prior for $\mathbf{U}$ and non-Gaussian priors for $\boldsymbol{\alpha}$ as $p(\boldsymbol{\alpha}) \propto e^{-\frac{1}{2} K(\boldsymbol{\alpha})}$ where $K(\boldsymbol{\alpha})$ is given by (12) or (14).

## B. Left and right-sided precision based models

For a low-rank matrix, components of each column vector are correlated as well as components of each row vector. In (6), the precision matrix $\boldsymbol{\alpha}^{-1 / 2}$ is used in the left side of $\mathbf{U}$. We refer to this as the left-sided precision based model, that helps to bring correlation between the components of each column of $\mathbf{X}$. Instead to bring correlations between the components of each row vector of $\mathbf{X}$, we can use

$$
\begin{equation*}
\mathbf{X}=\mathbf{U} \boldsymbol{\alpha}^{-1 / 2} \tag{15}
\end{equation*}
$$

which is defined as the right-sided precision based model. Then a natural question arises, which model to use? Our justification is that a user choice stems from the aspect of minimizing the number of variables to estimate. If the lowrank matrix is fat then the left-sided model should be used, otherwise the right-sided model. A further question arises on the prospect of developing a two sided precision based model - where both left and right precisions are used - that brings correlation between components of each column as well as components of each row. Such a two sided precision based model is described in the next section.

## III. TWO-SIDED PRECISION BASED MODEL

In this section, we propose to use precision matrices from two sides to model a random low-rank matrix, referred to as the two-sided precision based model. We set

$$
\begin{equation*}
\mathbf{X}=\boldsymbol{\alpha}_{L}^{-1 / 2} \mathbf{U} \boldsymbol{\alpha}_{R}^{-1 / 2} \tag{16}
\end{equation*}
$$

where $\boldsymbol{\alpha}_{L} \in \mathbb{R}^{p \times p}$ and $\boldsymbol{\alpha}_{R} \in \mathbb{R}^{q \times q}$ are positive definite random matrices. Our hypothesis is that the two-sided precision based model helps to inculcate correlations between
column vectors as well as row vectors, and hence promotes low-rank in a stronger manner than the one-sided precision based model of section II. Using the relation $\operatorname{vec}(\mathbf{X})=$ $\left(\boldsymbol{\alpha}_{R}^{-1 / 2} \otimes \boldsymbol{\alpha}_{L}^{-1 / 2}\right) \operatorname{vec}(\mathbf{U})=\left(\boldsymbol{\alpha}_{R} \otimes \boldsymbol{\alpha}_{L}\right)^{-1 / 2} \operatorname{vec}(\mathbf{U})$, we find that

$$
\begin{align*}
& p\left(\mathbf{X} \mid \boldsymbol{\alpha}_{L}, \boldsymbol{\alpha}_{R}\right) \\
& =\frac{\left|\boldsymbol{\alpha}_{R} \otimes \boldsymbol{\alpha}_{L}\right|^{1 / 2}}{(2 \pi)^{p q / 2}} \exp \left(-\operatorname{vec}(\mathbf{X})^{\top}\left(\boldsymbol{\alpha}_{R} \otimes \boldsymbol{\alpha}_{L}\right) \operatorname{vec}(\mathbf{X}) / 2\right) \\
& =\frac{\left|\boldsymbol{\alpha}_{L}\right|^{q / 2}\left|\boldsymbol{\alpha}_{R}\right|^{p / 2}}{(2 \pi)^{p q / 2}} \exp \left(-\operatorname{tr}\left(\mathbf{X}^{\top} \boldsymbol{\alpha}_{L} \mathbf{X} \boldsymbol{\alpha}_{R}\right) / 2\right) \tag{17}
\end{align*}
$$

To promote low-rank, we use a prior distribution $p\left(\boldsymbol{\alpha}_{L}, \boldsymbol{\alpha}_{R}\right)$. The marginal distribution of $\mathbf{X}$ then becomes

$$
\begin{equation*}
p(\mathbf{X})=\int_{\substack{\boldsymbol{\alpha}_{L} \succ \mathbf{0} \\ \boldsymbol{\alpha}_{R} \succ \mathbf{0}}} p\left(\mathbf{X} \mid \boldsymbol{\alpha}_{L}, \boldsymbol{\alpha}_{R}\right) p\left(\boldsymbol{\alpha}_{L}, \boldsymbol{\alpha}_{R}\right) d \boldsymbol{\alpha}_{R} d \boldsymbol{\alpha}_{L} . \tag{18}
\end{equation*}
$$

To the best of authors' ability, the evaluation of (18) for (17) is analytically intractable for relevant priors $p\left(\boldsymbol{\alpha}_{L}, \boldsymbol{\alpha}_{R}\right)$. Thus it is non-trivial to establish a direct connection between $p\left(\mathbf{X} \mid \boldsymbol{\alpha}_{L}, \boldsymbol{\alpha}_{R}\right)$ of (17) and the MAP estimator (4). Instead of a direct connection we establish an indirect connection by an approximation.

When marginalizing over $\boldsymbol{\alpha}_{L}$ for fixed $\boldsymbol{\alpha}_{R}$, we find that $p\left(\mathbf{X} \mid \boldsymbol{\alpha}_{R}\right)$ is a function of $\mathbf{X} \boldsymbol{\alpha}_{R} \mathbf{X}^{\top}$ alone. Following the discussion of section II-A, we find that the corresponding MAP estimator cost function for fixed $\beta$ becomes

$$
\begin{equation*}
\min _{\mathbf{X}} \beta\|\mathbf{y}-\mathbf{A} \operatorname{vec}(\mathbf{X})\|_{2}^{2}+\tilde{g}_{L}\left(\mathbf{X} \boldsymbol{\alpha}_{R} \mathbf{X}^{\top}\right), \tag{19}
\end{equation*}
$$

for some function $\tilde{g}_{L}(\cdot)$. A similar cost function can be found for a fixed $\alpha_{L}$ and $\beta$ by marginalizing over $\alpha_{R}$. Fixing $\boldsymbol{\alpha}_{R}$ thus promotes low-rank in $\mathbf{X} \boldsymbol{\alpha}_{R}^{1 / 2}$ and fixing $\boldsymbol{\alpha}_{L}$ thus promotes low-rank in $\boldsymbol{\alpha}_{L}^{1 / 2} \mathbf{X}$. When the prior is such that $p(\boldsymbol{\alpha})$ is small when $\boldsymbol{\alpha}$ is approximately rank-deficient, we find that the method promotes low rank in $\mathbf{X}$ since it is improbable that $\boldsymbol{\alpha}$ has small eigenvalues. In the next section we show that the key to promoting low-rank in $\mathbf{X}$ is to make many eigenvalues of $\boldsymbol{\alpha}$ large. We discuss the roles of $\boldsymbol{\alpha}_{L}$ and $\boldsymbol{\alpha}_{R}$ in the next section.

## A. Interpretation of the precision matrices

For a low-rank matrix, a column vector can be represented by a linear combination of few column vectors. Alternatively, we can state that the column vectors of a low-rank matrix lie in a low-dimensional subspace. Hence the objective of this section is to show that the proposed model (17) helps to establish such an argument when the inverse precision matrices have few dominant eigenvalues.

Let us denote $\left(i, j\right.$ )'th component of $\boldsymbol{\alpha}_{R}^{-1}$ by $\left[\boldsymbol{\alpha}_{R}^{-1}\right]_{i j}$ and $i$ 'th column vector of $\mathbf{X}$ by $\mathbf{x}_{i}$, respectively. From (17) we find that the covariance matrix of $\mathbf{x}_{i}$ is

$$
\begin{equation*}
\mathcal{E}\left[\mathbf{x}_{i} \mathbf{x}_{i}^{\top}\right]=\left[\boldsymbol{\alpha}_{R}^{-1}\right]_{i i} \boldsymbol{\alpha}_{L}^{-1} \tag{20}
\end{equation*}
$$

The covariance matrix of $\mathbf{x}_{i}$ is $\boldsymbol{\alpha}_{L}^{-1}$ times the scaling constant $\left[\boldsymbol{\alpha}_{R}^{-1}\right]_{i i}$. Let $\lambda_{L, k}$ denotes the $k^{\prime}$ 'th largest eigenvalue of $\boldsymbol{\alpha}_{L}^{-1}$
and $\lambda_{R, k}$ denotes the $k$ 'th largest eigenvalue of $\boldsymbol{\alpha}_{R}^{-1}$. The eigenvalues of $\boldsymbol{\alpha}_{L}^{-1}$ and $\boldsymbol{\alpha}_{R}^{-1}$ are real and non-negative.
Assumption 1. We assume that either $\boldsymbol{\alpha}_{L}^{-1}$ or $\boldsymbol{\alpha}_{R}^{-1}$ has $r$ dominant eigenvalues. That means that $\frac{\lambda_{L, r}}{\lambda_{L, r+1}} \gg 1$ or $\frac{\lambda_{R, r}}{\lambda_{R, r+1}} \gg 1$.
Under assumption 1, $\boldsymbol{\alpha}_{L}^{-1}$ can be closely approximated by a positive semi-definite covariance matrix of rank $r$. This matrix approximation also holds for the covariance matrix of $\mathbf{x}_{i}$ due to the relation (20). A natural qualitative argument is that $\mathbf{x}_{i}$ approximately lies in the subspace spanned by the eigenvectors corresponding to the $r$ dominant eigenvalues, resulting in promoting low-rank in $\mathbf{X}$. A mathematical argument follows next. Let $\mathcal{P}_{L}$ denotes the orthogonal projection onto the subspace spanned by the $r$ eigenvectors of $\boldsymbol{\alpha}_{L}^{-1}$ corresponding to the $r$ largest eigenvalues, and let $\mathcal{P}_{L}^{\perp}$ denotes the orthogonal projection onto the subspace spanned by the eigenvectors corresponding to the $p-r$ smallest eigenvalues. Note that $\mathcal{P}_{L}^{\perp}=\mathbf{I}_{p}-\mathcal{P}_{L}$, and $\left\|\mathbf{x}_{i}\right\|_{2}^{2}=\left\|\mathcal{P}_{L}^{\perp} \mathbf{x}_{i}\right\|_{2}^{2}+\left\|\mathcal{P}_{L} \mathbf{x}_{i}\right\|_{2}^{2}$. For a scalar $0 \leq \varsigma<1$, we now endeavor to establish that the probability of the event $\left(\left\|\mathcal{P}_{L}^{\perp} \mathbf{x}_{i}\right\|_{2}^{2} \leq \varsigma\left\|\mathcal{P}_{L} \mathbf{x}_{i}\right\|_{2}^{2}\right)$ is high under assumption 1. First we state the following relation

$$
\begin{align*}
\operatorname{Pr}\left(\left\|\mathcal{P}_{L}^{\perp} \mathbf{x}_{i}\right\|_{2}^{2} \leq \varsigma\left\|\mathcal{P}_{L} \mathbf{x}_{i}\right\|_{2}^{2}\right) & \geq 1-\frac{B\left(\frac{r}{2}, \frac{n}{2}, \epsilon_{1}\right)}{B\left(\frac{r}{2}, \frac{n}{2}\right)}  \tag{21}\\
& \geq 1-C_{p, r} \epsilon_{1}^{r / 2} \tag{22}
\end{align*}
$$

where $B(\cdot, \cdot, \cdot)$ is the incomplete beta function, $B(\cdot, \cdot)=$ $B(\cdot, \cdot, 1)$ is the beta function, $C_{p, r}^{-1}=\frac{r}{2} B\left(\frac{r}{2}, \frac{n}{2}\right)$ and

$$
\epsilon_{1}=\frac{1}{1+\varsigma \frac{\lambda_{L, r}}{\lambda_{L, r+1}}} .
$$

The derivation of (21) and (22) is given in Appendix C. Under some technical conditions and assumption 1, we have that $C_{p, r} \epsilon_{1}^{r / 2} \ll 1$ and therefore $\operatorname{Pr}\left(\left\|\mathcal{P} \frac{1}{L} \mathbf{x}_{i}\right\|_{2}^{2} \leq \varsigma\left\|\mathcal{P}_{L} \mathbf{x}_{i}\right\|_{2}^{2}\right)$ is high. Similar arguments also can be put forward by considering a row vector of $\mathbf{X}$ where the eigenvalues of $\boldsymbol{\alpha}_{R}^{-1}$ will play role under assumption 1 . Next we show role of both precision matrices in promoting low-rank. Using (17) we note that covariance of $\operatorname{vec}(\mathbf{X})$ is $\left(\boldsymbol{\alpha}_{R}^{-1} \otimes \boldsymbol{\alpha}_{L}^{-1}\right)$. We use $\mathbf{X}_{r}$ to denote the best $r$-rank approximation of $\mathbf{X}$ (in the sense of Frobenius norm). Then we find that probability of event $\left(\left\|\mathbf{X}-\mathbf{X}_{r}\right\|_{F}^{2} \leq \varsigma\|\mathbf{X}\|_{F}^{2}\right)$ is high. We state the following relation

$$
\begin{align*}
& \operatorname{Pr}\left(\left\|\mathbf{X}-\mathbf{X}_{r}\right\|_{F}^{2} \leq \varsigma\|\mathbf{X}\|_{F}^{2}\right) \\
& \geq 1-\frac{B\left(\frac{r^{2}}{2}, \frac{(p-r)(q-r)}{2}, \epsilon_{2}\right)}{B\left(\frac{r^{2}}{2}, \frac{(p-r)(q-r)}{2}\right)}  \tag{23}\\
& \geq 1-C_{p, q, r} \epsilon_{2}^{r^{2} / 2} \tag{24}
\end{align*}
$$

The derivation of (23) and (24) is given in Appendix C. In
(23) and (24), $C_{p, q, r}^{-1}=\frac{r^{2}}{2} B\left(\frac{r^{2}}{2}, \frac{(p-r)(q-r)}{2}\right)$ and

$$
\epsilon_{2}=\frac{1}{1+\frac{\varsigma}{1-\varsigma} \frac{\lambda_{L, r}}{\lambda_{L, r+1}} \frac{\lambda_{R, r}}{\lambda_{R, r+1}}}
$$

Under some technical conditions and assumption 1, we have $C_{p, q, r} \epsilon_{2}^{r^{2} / 2} \ll 1$, and $\operatorname{Pr}\left(\left\|\mathbf{X}-\mathbf{X}_{r}\right\|_{F}^{2} \leq \varsigma\|\mathbf{X}\|_{F}^{2}\right)$ is high. We provide a numerical example in Figure 1 where $\operatorname{Pr}\left(\left\|\mathbf{X}-\mathbf{X}_{r}\right\|_{F}^{2} \leq \varsigma\|\mathbf{X}\|_{F}^{2}\right)$ and the two lower bounds of (23) and (24) are shown. For the numerical experiments we used $p=q=25, r=3, \varsigma=0.05$ and $\frac{\lambda_{L, r}}{\lambda_{L, r+1}}=\frac{\lambda_{R, r}}{\lambda_{R, r+1}}$. In the simulation we used $\lambda_{L, i}=\lambda_{R, j}$ for $1 \leq i, j \leq r$ and $\lambda_{L, k}=\lambda_{R, l}$ for $r+1 \leq k \leq p$ and $r+1 \leq j \leq q$. We computed $\operatorname{Pr}\left(\left\|\mathbf{X}-\mathbf{X}_{r}\right\|_{F}^{2} \leq \varsigma\|\mathbf{X}\|_{F}^{2}\right)$ as an empirical probability in a frequentist manner via Monte Carlo simulations.


Fig. 1. Plots of empirical probability of $\left\|\mathbf{X}-\mathbf{X}_{r}\right\|_{F}^{2} \leq \varsigma\|\mathbf{X}\|_{F}^{2}$ in red solid line, the bound (23) in blue dotted line and the bound (24) in green dashed line, versus the value of $\frac{\lambda_{L, r}}{\lambda_{L, r+1}}$.

## IV. Practical algorithms

Considering the two-sided precision based model in section III, we denote the model parameters as $\Theta \triangleq$ $\left\{\boldsymbol{\alpha}_{L}, \boldsymbol{\alpha}_{R}, \beta\right\}$. The optimal inference problem is

$$
\hat{\mathbf{X}}, \hat{\boldsymbol{\Theta}}=\arg \max _{\mathbf{X}, \boldsymbol{\Theta}} p(\mathbf{X}, \mathbf{y}, \boldsymbol{\Theta})
$$

which is the MAP estimator for $\mathbf{X}$ and $\Theta$. We here assume that $p(\boldsymbol{\Theta})=p\left(\boldsymbol{\alpha}_{L}\right) p\left(\boldsymbol{\alpha}_{L}\right) p(\beta)$ for computational simplicity. It is natural that the optimal inference problem for the onesided precision based model (6) is a special case of the inference problem for two-sided precision based model (16) with the deterministic choice $\boldsymbol{\alpha}_{R}=\mathbf{I}_{q}$. Direct handling of the optimal inference problem is limited due to lack of analytical tractability. Therefore various approximations are used to design practical algorithms. This section is dedicated to design new algorithms (type II estimators) via the expectationmaximization (EM) approach [24], [35]. Here we assume that the noise precision $\beta$ has a Gamma distribution as

$$
p(\beta)=\operatorname{Gamma}(\beta \mid a+1, b)=\frac{b^{a+1}}{\Gamma(a+1)} \beta^{a} e^{-b \beta}
$$

with $a>-1, b>0$ and $\beta \geq 0$. We here set the shape parameter to be $a+1$ (rather than the standard convention of using $a$ ) since we later use approximately flat distributions
$(a, b \approx 0)$, giving the Gamma prior minor influence on the inference problem. However, in simulations we set the parameters $a$ and $b$ to small non-zero values to provide numerical stability (typically $10^{-5}$ ).

The objective of EM is to solve the following problem

$$
\begin{equation*}
\max _{\boldsymbol{\Theta}} \log p(\mathbf{y}, \boldsymbol{\Theta})=\max _{\boldsymbol{\Theta}}\{\log p(\mathbf{y} \mid \boldsymbol{\Theta})+\log p(\boldsymbol{\Theta})\} \tag{25}
\end{equation*}
$$

which is an approximation of the optimal inference problem. We refer to $\mathbf{X}$ as a latent variable in EM, and denote the EM help function by

$$
\begin{aligned}
Q\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\prime}\right) & =\mathcal{E}_{\mathbf{X} \mid \mathbf{y}, \boldsymbol{\Theta}^{\prime}}[\log p(\mathbf{y}, \mathbf{X} \mid \boldsymbol{\Theta})] \\
& =\int_{\mathbf{X}} p\left(\mathbf{X} \mid \mathbf{y}, \boldsymbol{\Theta}^{\prime}\right) \log p(\mathbf{y}, \mathbf{X} \mid \boldsymbol{\Theta}) d \mathbf{X}
\end{aligned}
$$

where $\mathcal{E}[\cdot]$ denotes the expectation operator. The iterative formulation of EM guarantees a locally optimum solution of (25) through the following steps.

1. Initialize the method with the parameter values $\boldsymbol{\Theta}^{\prime}=\left\{\boldsymbol{\alpha}_{L}^{\prime}, \boldsymbol{\alpha}_{R}^{\prime}, \beta^{\prime}\right\}=\left\{\mathbf{I}_{p}, \mathbf{I}_{q}, 1\right\}$.
2. $\mathbf{E}$-step: Evaluate $p\left(\mathbf{X} \mid \mathbf{y}, \mathbf{\Theta}^{\prime}\right)$ where $\mathbf{\Theta}^{\prime}$ is the value of $\boldsymbol{\Theta}$ from the previous iteration. We find that

$$
p\left(\mathbf{X} \mid \mathbf{y}, \mathbf{\Theta}^{\prime}\right)=\mathcal{N}\left(\operatorname{vec}(\mathbf{X}) ; \operatorname{vec}(\hat{\mathbf{X}}), \boldsymbol{\Sigma}^{\prime}\right)
$$

where

$$
\begin{align*}
& \operatorname{vec}(\hat{\mathbf{X}})=\beta^{\prime} \boldsymbol{\Sigma}^{\prime} \mathbf{A}^{\top} \mathbf{y}  \tag{26}\\
& \boldsymbol{\Sigma}^{\prime}=\left(\left(\boldsymbol{\alpha}_{R}^{\prime} \otimes \boldsymbol{\alpha}_{L}^{\prime}\right)+\beta^{\prime} \mathbf{A}^{\top} \mathbf{A}\right)^{-1}
\end{align*}
$$

3. M-step: Update $\Theta$ as

$$
\begin{equation*}
\boldsymbol{\Theta}=\arg \max _{\boldsymbol{\Theta}}\left\{Q\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\prime}\right)+\log p(\boldsymbol{\Theta})\right\}, \tag{27}
\end{equation*}
$$

where for our model

$$
\begin{align*}
& Q\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\prime}\right)=\mathcal{E}_{\mathbf{X} \mid \mathbf{y}, \boldsymbol{\Theta}^{\prime}}[\log p(\mathbf{y}, \mathbf{X} \mid \boldsymbol{\Theta})]=\mathrm{constant} \\
& -\frac{\beta}{2}\|\mathbf{y}-\mathbf{A} \operatorname{vec}(\hat{\mathbf{X}})\|_{2}^{2}-\frac{1}{2} \operatorname{tr}\left(\boldsymbol{\alpha}_{L} \hat{\mathbf{X}} \boldsymbol{\alpha}_{R} \hat{\mathbf{X}}^{\top}\right) \\
& -\frac{1}{2} \operatorname{tr}\left(\boldsymbol{\Sigma}^{-1} \boldsymbol{\Sigma}^{\prime}\right)+\frac{q}{2} \log \left|\boldsymbol{\alpha}_{L}\right|+\frac{p}{2} \log \left|\boldsymbol{\alpha}_{R}\right| \\
& +\frac{m}{2} \log \beta \tag{28}
\end{align*}
$$

and $\boldsymbol{\Sigma}=\left(\left(\boldsymbol{\alpha}_{R} \otimes \boldsymbol{\alpha}_{L}\right)+\beta \mathbf{A}^{\top} \mathbf{A}\right)^{-1}$.
By maximizing (27) for the noise precision $\beta$, we find the update equation

$$
\begin{equation*}
\beta=\frac{m+2 a}{\|\mathbf{y}-\mathbf{A v e c}(\hat{\mathbf{X}})\|_{2}^{2}+\operatorname{tr}\left(\mathbf{A} \boldsymbol{\Sigma}^{\prime} \mathbf{A}^{\top}\right)+2 b} \tag{29}
\end{equation*}
$$

The left and right precisions are updated as follows.
a) For Schatten s-norm: Using the Schatten $s$-norm prior (12) gives us the update equations

$$
\begin{align*}
& \boldsymbol{\alpha}_{L}=c_{s}\left(\hat{\mathbf{X}} \boldsymbol{\alpha}_{R}^{\prime} \hat{\mathbf{X}}^{\top}+\tilde{\boldsymbol{\Sigma}}_{L}+\epsilon \mathbf{I}_{p}\right)^{(s-2) / 2} \\
& \boldsymbol{\alpha}_{R}=c_{s}\left(\hat{\mathbf{X}}^{\top} \boldsymbol{\alpha}_{L}^{\prime} \hat{\mathbf{X}}+\tilde{\boldsymbol{\Sigma}}_{R}+\epsilon \mathbf{I}_{q}\right)^{(s-2) / 2} \tag{30}
\end{align*}
$$

where $c_{s}=(s / 2)^{s / 2}$ and $\tilde{\boldsymbol{\Sigma}}_{L} \in \mathbb{R}^{p \times p}$ and $\tilde{\boldsymbol{\Sigma}}_{R} \in \mathbb{R}^{q \times q}$ are matrices with elements

$$
\begin{aligned}
{\left[\tilde{\boldsymbol{\Sigma}}_{L}\right]_{i j} } & =\operatorname{tr}\left(\boldsymbol{\Sigma}^{\prime}\left(\boldsymbol{\alpha}_{R}^{\prime} \otimes \mathbf{E}_{i j}^{(L)}\right)\right), \\
{\left[\tilde{\boldsymbol{\Sigma}}_{R}\right]_{i j} } & =\operatorname{tr}\left(\boldsymbol{\Sigma}^{\prime}\left(\mathbf{E}_{i j}^{(R)} \otimes \boldsymbol{\alpha}_{L}^{\prime}\right)\right),
\end{aligned}
$$

and $\mathbf{E}_{i j}^{(L)} \in \mathbb{R}^{p \times p}, \mathbf{E}_{i j}^{(R)} \in \mathbb{R}^{q \times q}$ are matrices with ones in position $(i, j)$ and zeros otherwise.
b) Log-determinant penalty: For the log-determinant prior (14) the update equations become

$$
\begin{align*}
& \boldsymbol{\alpha}_{L}=\nu\left(\hat{\mathbf{X}} \boldsymbol{\alpha}_{R}^{\prime} \hat{\mathbf{X}}^{\top}+\tilde{\boldsymbol{\Sigma}}_{L}+\epsilon \mathbf{I}_{p}\right)^{-1} \\
& \boldsymbol{\alpha}_{R}=\nu\left(\hat{\mathbf{X}}^{\top} \boldsymbol{\alpha}_{L}^{\prime} \hat{\mathbf{X}}+\tilde{\boldsymbol{\Sigma}}_{R}+\epsilon \mathbf{I}_{q}\right)^{-1} \tag{31}
\end{align*}
$$

The derivations of (30) and (31) are given in Appendix D.
4. Stop iterating if $\hat{\mathbf{X}}$ does not change significantly. Otherwise go to step 2.

## A. Balancing the precisions

We have found that in practical algorithms, there is a chance that one of the two precisions becomes large and the other becomes small. A small precision results in numerical instability in the Kronecker covariance structure (17). To prevent the imbalance we rescale the matrix precisions in each iteration such that 1 ) the a-priori and a-posteriori squared Frobenius norm of $\mathbf{X}$ are equal,

$$
\begin{aligned}
& \mathcal{E}\left[\|\mathbf{X}\|_{F}^{2} \mid \boldsymbol{\alpha}_{L}, \boldsymbol{\alpha}_{R}\right]=\operatorname{tr}\left(\boldsymbol{\alpha}_{L}^{-1}\right) \operatorname{tr}\left(\boldsymbol{\alpha}_{R}^{-1}\right) \\
& =\mathcal{E}\left[\|\mathbf{X}\|_{F}^{2} \mid \boldsymbol{\alpha}_{L}, \boldsymbol{\alpha}_{R}, \beta, \mathbf{y}\right]=\|\hat{\mathbf{X}}\|_{F}^{2}+\operatorname{tr}(\boldsymbol{\Sigma}),
\end{aligned}
$$

and 2) the contribution of each precision to the norm is equal,

$$
\operatorname{tr}\left(\boldsymbol{\alpha}_{L}^{-1}\right)=\operatorname{tr}\left(\boldsymbol{\alpha}_{R}^{-1}\right)
$$

The rescaling makes the algorithm more stable and often improves estimation performance.

## B. Nomenclature

Here we explain the nomenclature used for the RSVM algorithms throughout the rest of the paper. We use the name left-sided RSVM for the left-sided model where $\boldsymbol{\alpha}_{L}$ is random and $\boldsymbol{\alpha}_{R}=\mathbf{I}_{q}$ and right-sided RSVM for the right-sided model where $\boldsymbol{\alpha}_{R}$ is random and $\boldsymbol{\alpha}_{L}=\mathbf{I}_{p}$. We refer to the model where both $\boldsymbol{\alpha}_{L}$ and $\boldsymbol{\alpha}_{R}$ are random as the two-sided RSVM or simply as RSVM. Throughout the paper we assume that $\boldsymbol{\alpha}_{L}$ and $\boldsymbol{\alpha}_{R}$ are independent with distributions $p(\boldsymbol{\alpha}) \propto e^{-\frac{1}{2} K(\boldsymbol{\alpha})}$. We refer to the RSVM method with priors related to the logdeterminant penalty function (14) as RSVM-LD and to the RSVM method priors related to the Schatten $s$-norm penalty function (12) as RSVM-SN. For example, left-sided RSVMLD is RSVM with log-determinant prior (14) and a left-sided precision.

## C. Complexity analysis

One motivation for the RSVM algorithm is that it can infer low-rank directly without factorizing the matrix $\mathbf{X}$. However, one disadvantage of this is that for given precision matrices, the estimation of $\mathbf{X}$ is a Gaussian inference problem with $p q$ variables, requiring computationally demanding matrix inverses. To study how the problem size affects the runtime, we here study the (asymptotic) computational complexity of the algorithm.
The most computationally demanding operation of the RSVM algorithm is the computation of $\boldsymbol{\Sigma}$. A direct approach
to computing $\boldsymbol{\Sigma}$ is to invert a $p q \times p q$ matrix, this requires $\mathcal{O}\left((p q)^{3}\right)$ multiplications. However, this does not use the fact that $\operatorname{rank}\left(\mathbf{A}^{\top} \mathbf{A}\right) \leq m \ll p q$. By using the fact and following Woodbury matrix inversion formula [24] we can write

$$
\boldsymbol{\Sigma}=\left(\boldsymbol{\alpha}_{R}^{-1} \otimes \boldsymbol{\alpha}_{L}^{-1}\right)-\left(\boldsymbol{\alpha}_{R}^{-1} \otimes \boldsymbol{\alpha}_{L}^{-1}\right) \mathbf{A}^{\top} \mathbf{C}_{\mathbf{y}}^{-1} \mathbf{A}\left(\boldsymbol{\alpha}_{R}^{-1} \otimes \boldsymbol{\alpha}_{L}^{-1}\right)
$$

where

$$
\mathbf{C}_{\mathbf{y}}=\beta^{-1} \mathbf{I}_{m}+\mathbf{A}\left(\boldsymbol{\alpha}_{R}^{-1} \otimes \boldsymbol{\alpha}_{L}^{-1}\right) \mathbf{A}^{\top} .
$$

The computation of $\boldsymbol{\Sigma}$ thus reduces to computing $\left(\boldsymbol{\alpha}_{R_{1}}^{-1} \otimes \boldsymbol{\alpha}_{L_{1}}^{-1}\right)$ (requiring $\mathcal{O}\left(p^{3}+q^{3}+p^{2} q^{2}\right)$ multiplications), $\mathbf{A}\left(\boldsymbol{\alpha}_{R}^{-1} \otimes \boldsymbol{\alpha}_{L}^{-1}\right)$ and $\mathbf{A}\left(\boldsymbol{\alpha}_{R}^{-1} \otimes \boldsymbol{\alpha}_{L}^{-1}\right) \mathbf{A}^{\top}$ (requiring $\mathcal{O}\left(m p^{2} q^{2}\right)$ multiplications) and $\mathbf{C}_{\mathbf{y}}^{-1}$ (requiring $\mathcal{O}\left(m^{3}\right)$ multiplications). Therefore the total complexity is

$$
\mathcal{O}\left(p^{3}+q^{3}+p^{2} q^{2}+m^{3}+m p^{2} q^{2}\right) \approx \mathcal{O}\left(m p^{2} q^{2}\right)
$$

where $\mathcal{O}\left(m p^{2} q^{2}\right)$ dominates other terms. As $m \ll p q$, we note that $\mathcal{O}\left(m p^{2} q^{2}\right) \ll \mathcal{O}\left((p q)^{3}\right)$, and hence the use of Woodbury matrix inversion helps to reduce computational complexity in contrast to direct matrix inversion. Further we note that the complexity does not depend on the rank of $\mathbf{X}$. This is because RSVM, in contrast to factorization based methods [11], does not use rank information as an a-priori information.

## V. Simulation experiments

## A. Simulations setup

Here we describe datasets, performance measure, experimental setup, competing algorithms and computational resources. First, we used synthetic data for low-rank matrix reconstruction (LRMR) and low-rank matrix completion (LRMC). For real data we used the MovieLens dataset [20]. Second, to compare the algorithms for synthetic data, we use the normalized-mean-square-error

$$
\mathrm{NMSE} \triangleq \mathcal{E}\left[\|\hat{\mathbf{X}}-\mathbf{X}\|_{F}^{2}\right] / \mathcal{E}\left[\|\mathbf{X}\|_{F}^{2}\right]
$$

as the performance measure. The NMSE is evaluated by averaging over many instances of $\mathbf{n}, \mathbf{A}$ and $\mathbf{X}$ using Monte-Carlo simulations. Third is the experimental setup. For experiments with synthetic data, we evaluated the NMSE as follows.

1) For LRMR, the random measurement matrix $\mathbf{A} \in \mathbb{R}^{m \times p q}$ was generated by independently drawing the elements from $\mathcal{N}(0,1)$ and normalizing the column vectors to unit norm. For LRMC, each row of A contains a 1 in a random position and zero otherwise with the constraint that the rows are linearly independent.
2) Matrices $\mathbf{F} \in \mathbb{R}^{p \times r}$ and $\mathbf{B} \in \mathbb{R}^{q \times r}$ with elements drawn from $\mathcal{N}(0,1)$ were randomly generated and the matrix $\mathbf{X}$ was formed as $\mathbf{X}=\mathbf{F B}^{\top}$. Note that $\mathbf{X}$ has rank $r$.
3) Generate the measurement $\mathbf{y}=\mathbf{A v e c}(\mathbf{X})+\mathbf{n}$, where $\mathbf{n} \sim$ $\mathcal{N}\left(\mathbf{0}, \beta^{-1} \mathbf{I}_{m}\right)$ and $\beta^{-1}$ is chosen such that the signal-to-measurement-noise ratio is

$$
\mathrm{SNR} \triangleq \frac{\mathcal{E}\left[\|\mathbf{A} \operatorname{vec}(\mathbf{X})\|_{2}^{2}\right]}{\mathcal{E}\left[\|\mathbf{n}\|_{2}^{2}\right]}=\beta \frac{r p q}{m}
$$

for LRMR and $\mathrm{SNR}=\beta r$ for LRMC.
4) Estimate $\hat{\mathbf{X}}$ using competing algorithms and calculate the error $\|\hat{\mathbf{X}}-\mathbf{X}\|_{F}^{2}$.
5) Repeat steps $2-4$ for $T_{1}$ times.
6) Repeat steps $1-5$ for $T_{2}$ times.
7) Then compute the NMSE by averaging.

In the simulations we chose $T_{1}=T_{2}=10$, which means that the averaging was done over 100 realizations. Fourth, we evaluate the performance of RSVM vis-a-vis other methods for LRMR and LRMC. For LRMR, we compared with NN and the variational Bayes method of [11] (referred henceforth as VB-1). The use of VB-1 for LRMR was not addressed in [11] and hence we derive the algorithm in Appendix G. Further, we also compare with the performance of the Schatten Norm Algorithm (SNA) where $\hat{\mathbf{X}}$ is the solution to

$$
\begin{equation*}
\min _{\mathbf{X}} \operatorname{tr}\left(\left(\mathbf{X X}^{\top}\right)^{s / 2}\right) \text { s.t. }\|\mathbf{y}-\mathbf{A v e c}(\mathbf{X})\|_{2} \leq \delta \tag{32}
\end{equation*}
$$

with $s=0.5$ and $\delta=\beta^{-1} \sqrt{m+\sqrt{8 m}}$ as in (5), where $\beta$ denotes the true noise precision. Both NN and SNA thus require knowledge of the noise power. The SNA problem is non-convex and we use a gradient search method to find the solution; to initialize the method, we used $\operatorname{vec}(\hat{\mathbf{X}})=$ $\mathbf{A}^{\top}\left(\mathbf{A A}^{\top}\right)^{-1} \mathbf{y}$. Then, for LRMC, we compare with NN, VB1, PMF [17], WTN [19] and the variational Bayes method of [15] (referred to as VB-2). Finally, we mention our computational resources. For the experiments, we used a Dell Latitude E6400 laptop computer with a 3 GHz processor and 8GB memory.

## B. Experiments using synthetic data for LRMR

The objective of our first experiment is to compare the performance of the two-sided precision based model the onesided precision based models. For the one-sided model, we also have two choices - left-sided and right-sided - and hence it interesting to know which choice is better for a particular setup. In this experiment, we fixed $\operatorname{rank}(\mathbf{X})=r=3, p=15$, $q=30, \mathrm{SNR}=20 \mathrm{~dB}$ and varied $m$. The results are shown in Figure 2 where NMSE is plotted against the normalized number of measurements $m /(p q)$. We note that for RSVMLD, the right-sided model performs best for $m / p q<0.6$, the left-sided model performs best for $m / p q>0.6$ and that the two-sided RSVM-LD gives a good compromise between the left- and right-sided RSVM-LD. For RSVM-SN we find that the left-sided model performs better than both the right- and two-sided model for $m / p q \neq 0.7$. However, the two-sided model has a more consistent performance improvement with increasing $m / p q$ while the left and right-sided models show performance degradation for $m / p q=0.7$ and $m / p q=0.8$, respectively. Henceforth we use the two-sided models because of their reasonable good performance. It is possible that the one-sided models are preferable in other scenarios. We mention that, for RSVM-SN, it was empirically found that $s=0.5$ provides good performance. The same trend also repeats for LRMC, reported in section V-C. Henceforth RSVM-SN with $s=0.5$ is used unless stated otherwise.

In the second experiment we report the LRMR performance of RSVM-LD and RSVM-SN vis-a-vis NN, VB-1 and SNA. We mention that NN and SNA know the measurement noise power (see (5) and (32)), and VB-1 knows the rank of $\mathbf{X}$. The results are shown in Figure 3. With the parameters of


Fig. 2. NMSE vs. $m /(p q)$ for LRMR at $\mathrm{SNR}=20 \mathrm{~dB}$.
the first experiment $(\operatorname{rank}(\mathbf{X})=r=3, p=15, q=30$, SNR $=20 \mathrm{~dB}$ ), we show NMSE vs. $m /(p q)$ in Figure 3 (a). We observe that RSVM-LD provides the best performance for $m / p q<0.9$, whereas NN, SNA and RSVM-SN are close to each other with NN performing better than SNA and RSVMSN. In Figure 3 (b), we report performance NMSE vs. SNR while rank $r=3$ and $m /(p q)=0.7$ are fixed. We find that RSVM-LD shows best performance in the middle SNR region $(15<$ SNR $<35)$, while NN and SNA perform best in the low and high SNR regions. Next, in Figure 3 (c), we report the NMSE vs. rank for $m /(p q)=0.7$ and SNR $=20 \mathrm{~dB}$. We find that RSVM-LD is the best, and RSVM-SN and NN are comparable. At this point it is interesting to investigate the performance at higher SNR, shown in Figure 4 for SNR $=40$ dB. While NN performs better for the case of NMSE versus $m /(p q)$ in Figure 4 (a) where $r=3$, we notice that RSVMLD is promising when rank is higher as reported in Figure 4 (b). Finally, in Figure 5 we show the cpu execution times of all competing algorithms for the setup reported in Figure 3 (c). From Figure 5, we conclude that NN, implemented using the cvx toolbox [36], is the fastest algorithm followed by the RSVM algorithms (implemented in Matlab).

## C. Experiments using synthetic data for LRMC

For LRMC, the objective of the first experiment is to empirically find a good choice of $s$ for RSVM-SN. Like the first experiment for LRMR in section V-B, we fixed $\operatorname{rank}(\mathbf{X})=r=3, p=15, q=30, \mathrm{SNR}=20 \mathrm{~dB}$ and varied $m$. The performance of RSVM-SN for different $s$ is shown in Figure 6. We found that $s=0.5$ is a good choice, and decided to use that throughout all relevant experiments.
Like the second experiment for LRMR in section V-B, we conducted the second experiment here to evaluate the LRMC performance of RSVM-LD and RSVM-SN vis-a-vis NN, VB1, VB-2, PMF and WTN. We used Matlab codes for VB-1 and PMF from their respective authors. For other methods, we used our own codes. We manually tuned the competing algorithms to improve their performance. The results are


Fig. 3. Comparison of algorithms for LRMR. (a) NMSE versus. $m /(p q)$ at $\mathrm{SNR}=20 \mathrm{~dB}$ and $r=3$. (b) NMSE versus SNR at rank $r=3$ and $m /(p q)=0.7$. (c) NMSE versus rank $r$ at $m /(p q)=0.7$ and SNR $=20 \mathrm{~dB}$.


Fig. 4. Comparison of algorithms for LRMR at $\mathrm{SNR}=40 \mathrm{~dB}$. (a) NMSE versus. $m /(p q)$ for $r=3$. (b) NMSE versus rank $r$ for $m /(p q)=0.7$.


Fig. 5. Mean cputime versus rank $r$ for the setup in Figure 3 (c).
shown in Figure 7. Figure 7 (a) shows NMSE vs. $m /(p q)$ at $r=3, p=15, q=30$, $\mathrm{SNR}=20 \mathrm{~dB}$. We observe that for $m / p q \leq 0.5$, RSVM-LD, PMF and VB-2 provide the best performance while for $m / p q \geq 0.6 \mathrm{VB}-1$ performs


Fig. 6. LRMC performance: NMSE versus $m /(p q)$ for RSVM-SN at different choice of $s$.
the best. Then, in Figure 7 (b), we report performance NMSE versus. SNR where rank $r=3$ and $m /(p q)=0.7$ are fixed, and find that VB-1 shows best performance closely followed by PMF and RSVM-LD. Next, in Figure 7 (c), we report the NMSE versus rank for fixed $m /(p q)=0.7$ and $\mathrm{SNR}=20 \mathrm{~dB}$, we find that VB-1 gave the best performance for rank $\leq 3$ but then quickly degrades in performance. For higher ranks, while PMF showed the best performance, the proposed RSVM showed good performance.

## D. Experiments using MovieLens data for LRMC - Movie rating prediction

To evaluate the performance on real data we used the MovieLens 100 K dataset [20]. The dataset contains usermovie pairs where in each pair, a user provides an integer rating between $1-5$ to a movie. Each user has only rated a few movies according to the features of movies, and the underlying assumption is that the rating matrix has a factorized representation, leading to low-rank. Now let us assume that we observe few ratings of the large rating matrix randomly. The


Fig. 7. Comparison of algorithms for LRMC. (a) NMSE versus. $m /(p q)$ at $\mathrm{SNR}=20 \mathrm{~dB}$ and $r=3$. (b) NMSE versus SNR at rank $r=3$ and $m /(p q)=0.7$. (c) NMSE versus rank $r$ at $m /(p q)=0.7$ and SNR $=20 \mathrm{~dB}$.

TABLE I
RMSE of algorithms for Movielens

| Algorithm | RMSE |
| :--- | :---: |
| RSVM-LD | 0.0410 |
| RSVM-SN | 0.0588 |
| NN | 0.0757 |
| VB-1 | 0.0389 |
| SNA | 0.1255 |
| VB-2 | 0.0389 |
| PMF | 0.1258 |
| WTN | 0.1249 |

goal is to infer the missing ratings from the observed ratings and hence the problem is an LRMC problem.

In our experimental study, we used the u1 datasets (both training and testing) from the MovieLens dataset. Then we used a portion of the u 1 datasets as $\mathbf{X}$. The dimensions of $\mathbf{X}$ are $p=q=100$. According to the MovieLens dataset instructions, we used $m=307$ measurements. The measurements are collected via a pre-determined element picking matrix $\mathbf{A}$. Using the measurements and $\mathbf{X}$ from the $u 1$ training data, we learn the model parameters for all competing algorithms. Then, using the learned parameters we perform LRMC for the u1 test data. For performance comparison, we use root-mean-square-error

$$
\text { RMSE }=\frac{1}{\left|J_{\text {test }}\right|} \sqrt{\sum_{(i, j) \in J_{\text {test }}}\left(X_{i j}-\hat{X}_{i j}\right)^{2}}
$$

where $J_{\text {test }}$ denotes the set of unknown ratings, as this is a standard performance measure for movie rating prediction. The performance of all competing algorithms are shown in Table I. It can be observed that RSVM algorithms provide good performance.

## E. Reproducible research

In the spirit of reproducible research, we provide code necessary for reproducing the results in the website:
https://github.com/MartinSundin/rsvm_simulation_code and https://www.kth.se/ees/omskolan/organisation/avdelningar/sp /research/reproducibleresearch-1.433797. The code can be used to reproduce the figures $2,3,6$ and 7 .

## VI. Conclusion

In this paper we develop Bayesian learning algorithms for low-rank matrix reconstruction in an under-determined setup. The framework relates low-rank penalty functions (MAP estimators) to type II estimators with either left- or rightsided precisions through the matrix Laplace transform and the concave conjugate formula. The model was further extended to the two-sided precision based model. Using expectationmaximization (EM) approach, we designed estimation algorithms that are capable to learn relevant parameters from data. Through simulations, for both real data and synthetic data, we have shown that the new Bayesian learning algorithms provide good performance vis-a-vis several existing methods.

## Appendix <br> DERIVATIONS

## A. The Laplace transform for positive definite matrices

We here summarize the definition of the Laplace transform for positive definite matrices. Further details can be found in [33]. Let $\mathbb{S}_{+}^{n}=\left\{\mathbf{Z} \in \mathbb{R}^{n \times n}: \mathbf{Z} \succeq \mathbf{0}\right\}$ be the space of $n \times n$ positive definite matrices and let $f$ be a real valued function on $\mathbb{S}_{+}^{n}$. The matrix Laplace transform of $f$ at $\mathbf{Y} \in \mathbb{S}_{+}^{n}$ is
$\mathcal{L} f(\mathbf{Y})=\int_{\mathbf{Z} \succeq \mathbf{0}} f(\mathbf{Z}) e^{-\operatorname{tr}(\mathbf{Z Y})} d \mathbf{Z}$, where $d \mathbf{Z}=\prod_{1 \leq i \leq j \leq n} d Z_{i j}$. The transform is defined for sufficiently nice functions [33] for which it converges when $\operatorname{Re} \mathbf{Y} \succeq \mathbf{Z}_{*}$ for some $\mathbf{Z}_{*}$. The inverse Laplace transform can be expressed as [33]

$$
\begin{gathered}
\frac{1}{(2 \pi i)^{n(n+1) / 2}} \int_{\operatorname{Re} \mathbf{Y}=\mathbf{Z}_{*}} \mathcal{L} f(\mathbf{Y}) e^{\operatorname{tr}(\mathbf{Y Z})} d \mathbf{Y} \\
\quad=\left\{\begin{array}{ll}
f(\mathbf{Z}) & , \text { if } \mathbf{Z} \in \mathbb{S}_{+}^{n} \\
0 & , \text { otherwise }
\end{array} .\right.
\end{gathered}
$$

## B. Derivation of the Laplace Approximation

The Laplace approximation is an approximation of the integral

$$
I=\int e^{-\frac{1}{2} f(\mathbf{a})} d \mathbf{a}
$$

where the integral is over $\mathbf{a} \in \mathbb{R}^{n}$. The function $f(\mathbf{a})$ is approximated by a second order polynomial around its minima $\mathbf{a}_{0}$ as

$$
f(\mathbf{a}) \approx f\left(\mathbf{a}_{0}\right)+\frac{1}{2}\left(\mathbf{a}-\mathbf{a}_{0}\right)^{\top} \mathbf{H}\left(\mathbf{a}-\mathbf{a}_{0}\right),
$$

where $\mathbf{H}=\left.\nabla^{2} f(\mathbf{a})\right|_{\mathbf{a}=\mathbf{a}_{0}}$ is the Hessian of $f(\mathbf{a})$ at $\mathbf{a}_{0}$. The term linear in a vanishes and $\mathbf{H} \succ \mathbf{0}$ at $\mathbf{a}_{0}$ since we expand around a minima. With this approximation, the integral becomes

$$
I \approx \int e^{-\frac{1}{2} f\left(\mathbf{a}_{0}\right)-\frac{1}{4}\left(\mathbf{a}-\mathbf{a}_{0}\right)^{\top} \mathbf{H}\left(\mathbf{a}-\mathbf{a}_{0}\right)} d \mathbf{a}=\sqrt{\frac{(4 \pi)^{n}}{|\mathbf{H}|}} e^{-\frac{1}{2} f\left(\mathbf{a}_{0}\right)}
$$

In (8), the integral is given by

$$
I=\frac{1}{(2 \pi)^{p q / 2}} \int_{\boldsymbol{\alpha} \succ \mathbf{0}} e^{-\frac{1}{2}[\operatorname{tr}(\boldsymbol{\alpha} \mathbf{Z})-q \log |\boldsymbol{\alpha}|+K(\boldsymbol{\alpha})]} d \boldsymbol{\alpha}
$$

Set $f(\mathbf{a})=\operatorname{tr}(\boldsymbol{\alpha} \mathbf{Z})-q \log |\boldsymbol{\alpha}|+K(\boldsymbol{\alpha})$, where $\mathbf{a}=\operatorname{vech}(\boldsymbol{\alpha})$ and $\operatorname{vech}(\cdot)$ is the half-vectorization operator for symmetric matrices, e.g.

$$
\operatorname{vech}\left(\begin{array}{ll}
a & b \\
b & c
\end{array}\right)=\left(\begin{array}{l}
a \\
b \\
c
\end{array}\right)
$$

Let $\boldsymbol{\alpha}_{0} \succ \mathbf{0}$ denote the minima of $f(\mathbf{a})$ and $\mathbf{H}$ the Hessian at $\boldsymbol{\alpha}_{0}$. Assuming that $\boldsymbol{\alpha}_{0}$ and $\mathbf{H}$ are "large" in the sense that the integral over $\boldsymbol{\alpha} \succ \mathbf{0}$ can be approximated by the integral over $\boldsymbol{\alpha} \in \mathbb{R}^{p \times p}$ we find that

$$
\begin{aligned}
I & \approx \frac{1}{(2 \pi)^{p q / 2}} \int e^{-\frac{1}{2} f\left(\mathbf{a}_{0}\right)-\frac{1}{4}\left(\mathbf{a}-\mathbf{a}_{0}\right)^{\top} \mathbf{H}\left(\mathbf{a}-\mathbf{a}_{0}\right)} d \mathbf{a} \\
& =\frac{(4 \pi)^{p^{2} / 2}}{(2 \pi)^{p q / 2}|\mathbf{H}|^{1 / 2}} e^{-\frac{1}{2} f\left(\mathbf{a}_{0}\right)}
\end{aligned}
$$

where $\mathbf{a}_{0}=\operatorname{vech}\left(\boldsymbol{\alpha}_{0}\right)$.

## C. Derivation of (21), (22), (23) and (24)

Let $\mathrm{P}_{1} \triangleq \operatorname{Pr}\left(\left\|\mathcal{P}_{L}^{\perp} \mathbf{x}_{i}\right\|_{2}^{2} \leq \varsigma\left\|\mathcal{P}_{L} \mathbf{x}_{i}\right\|_{2}^{2}\right)$ and assume that $1 \leq r<p$ and let $n=p-r$. The random variables $\mathcal{P}_{L} \mathbf{x}_{i}$ and $\mathcal{P}_{L}^{\perp} \mathbf{x}_{i}$ are independent zero-mean Gaussian variables. Let $w_{1}, w_{2}, \ldots, w_{r}, z_{1}, z_{2}, \ldots, z_{n}$ be i.i.d. $\mathcal{N}(0,1)$ variables. For the proofs of (21), (22), we find that

$$
\begin{align*}
P_{1} & =\operatorname{Pr}\left(\frac{\lambda_{L, r+1} z_{1}^{2}+\lambda_{L, r+2} z_{2}^{2}+\ldots \lambda_{L, p} z_{n}^{2}}{\lambda_{L, 1} w_{1}^{2}+\lambda_{L}, 2 w_{2}^{2}+\cdots+\lambda_{L, r} w_{r}^{2}} \leq \varsigma\right) \\
& (a) \\
& \geq \operatorname{Pr}\left(\frac{\lambda_{L, r+1}\left(z_{1}^{2}+z_{2}^{2}+\cdots+z_{n}^{2}\right)}{\lambda_{L, r}\left(w_{1}^{2}+w_{2}^{2}+\cdots+\cdots+w_{r}^{2}\right)} \leq \varsigma\right) \\
& =1-\operatorname{Pr}\left(\frac{\lambda_{L, r+1}\left(z_{1}^{2}+z_{2}^{2}+\cdots+z_{n}^{2}\right)}{\lambda_{L, r}\left(w_{1}^{2}+w_{2}^{2}+\cdots+w_{r}^{2}\right)} \geq \varsigma\right)  \tag{33}\\
& =1-\operatorname{Pr}\left(\frac{\left(w_{1}^{2}+w_{2}^{2}+\cdots+w_{r}^{2}\right) / r}{\left(z_{1}^{2}++_{2}^{2}+\cdots+\cdots+z_{n}^{2}\right) / n} \leq \frac{n \lambda_{L, r+1}}{\varsigma r \lambda_{L, r}}\right) \\
& \stackrel{(b)}{\geq} 1-\frac{B\left(\frac{r}{2}, \frac{n}{2}, \epsilon_{1}\right)}{B\left(\frac{r}{2}, \frac{n}{2}\right)} \\
& \stackrel{(c)}{\geq} 1-C_{p, r} \epsilon_{1}^{r / 2},
\end{align*}
$$

where $B(\cdot, \cdot, \cdot)$ is the incomplete beta function, $B(\cdot, \cdot)=$ $B(\cdot, \cdot, 1)$ is the beta function and

$$
C_{p, r}^{-1}=\frac{r}{2} B\left(\frac{r}{2}, \frac{n}{2}\right), \text { and } \epsilon_{1}=\frac{1}{1+\varsigma \frac{\lambda_{L, r}}{\lambda_{L, r+1}}} .
$$

In (a), $\frac{\lambda_{L, r+1} z_{1}^{2}+\lambda_{L, r+2} z_{2}^{2}+\ldots \lambda_{L, p} z_{n}^{2}}{\lambda_{L, 1} w_{1}^{2}+\lambda_{L, 2} w_{2}^{2}+\cdots+\lambda_{L, r} w_{r}^{2}} \leq \frac{\lambda_{L, r+1}\left(z_{1}^{2}+z_{2}^{2}+\cdots+z_{n}^{2}\right)}{\lambda_{L, r}\left(w_{1}^{2}+w_{2}^{2}+\cdots+w_{r}^{2}\right)}$ is used. In (b) we note that $\frac{\left(w_{1}^{2}+w_{2}^{2}+\cdots+w_{r}^{2}\right) / r}{\left(z_{1}^{2}+z_{2}^{2}+\cdots+z_{n}^{2}\right) / n}$ is F-distributed. Then, in (c), we use the following relation

$$
\begin{align*}
B\left(\frac{r}{2}, \frac{n}{2}, \epsilon_{1}\right) & =\int_{0}^{\epsilon_{1}} t^{r / 2-1}(1-t)^{n / 2-1} d t \\
& =\epsilon_{1}^{r / 2} \int_{0}^{1} u^{r / 2-1}\left(1-\epsilon_{1} u\right)^{n / 2-1} d u  \tag{34}\\
& \leq \frac{2}{r} \epsilon_{1}^{r / 2}
\end{align*}
$$

This shows (22).
Suppose $\mathrm{P}_{2} \triangleq \operatorname{Pr}\left(\left\|\mathbf{X}-\mathbf{X}_{r}\right\|_{F}^{2} \leq \varsigma\|\mathbf{X}\|_{F}^{2}\right)$. To show (24) we introduce the projection $\mathcal{P}_{R}$ onto the subspace spanned by the $r$ eigenvectors of $\boldsymbol{\alpha}_{R}^{-1}$ corresponding to the $r$ largest eigenvalues and $\mathcal{P}_{R}^{\perp}=\mathbf{I}_{q}-\mathcal{P}_{R}$. Using the projection operators $\mathcal{P}_{L}$ and $\mathcal{P}_{R}$, we have the following relation

$$
\begin{aligned}
\|\mathbf{X}\|_{F}^{2}= & \left\|\mathcal{P}_{L} \mathbf{X} \mathcal{P}_{R}\right\|_{F}^{2}+\left\|\mathcal{P}_{L} \mathbf{X} \mathcal{P}_{R}^{\perp}\right\|_{F}^{2} \\
& +\left\|\mathcal{P}_{L}^{\perp} \mathbf{X} \mathcal{P}_{R}\right\|_{F}^{2}+\left\|\mathcal{P}_{L}^{\perp} \mathbf{X} \mathcal{P} \frac{1}{R}\right\|_{F}^{2} .
\end{aligned}
$$

Next, $\mathbf{X}_{r}$ can be found using singular value decomposition of $\mathbf{X}$ because $\mathbf{X}_{r}$ is the best rank-r approximation of $\mathbf{X}$. If $\mathbf{P}_{1}$ and $\mathbf{P}_{2}$ are two projection operators with properties $\operatorname{rank}\left(\mathbf{P}_{1}\right)=p-r$ and $\operatorname{rank}\left(\mathbf{P}_{2}\right)=q-r$, then we have the following relation

$$
\begin{aligned}
\left\|\mathbf{X}-\mathbf{X}_{r}\right\|_{F}^{2} & =\min _{\mathbf{P}_{1}, \mathbf{P}_{2}}\left\|\mathbf{P}_{1} \mathbf{X} \mathbf{P}_{2}\right\|_{F}^{2} \\
& \leq\left\|\mathcal{P}_{L}^{\perp} \mathbf{X} \mathcal{P}_{R}^{\perp}\right\|_{F}^{2} .
\end{aligned}
$$

We now find following relation

$$
\begin{aligned}
\mathrm{P}_{2} & \triangleq \operatorname{Pr}\left(\left\|\mathbf{X}-\mathbf{X}_{r}\right\|_{F}^{2} \leq \varsigma\|\mathbf{X}\|_{F}^{2}\right) \\
& \geq \operatorname{Pr}\left(\left\|\mathcal{P} \perp \perp \mathbf{X} \mathcal{P}_{R}^{\perp}\right\|_{F}^{2} \leq \varsigma\|\mathbf{X}\|_{F}^{2}\right) \\
& \geq \operatorname{Pr}\left(\left\|\mathcal{P}_{L}^{\perp} \mathbf{X} \mathcal{P}_{R}^{\perp}\right\|_{F}^{2} \leq \varsigma\left(\left\|\mathcal{P}_{L} \mathbf{X} \mathcal{P}_{R}\right\|_{F}^{2}+\left\|\mathcal{P}_{L}^{\perp} \mathbf{X} \mathcal{P}_{R}^{\perp}\right\|_{F}^{2}\right)\right) \\
& =\operatorname{Pr}\left(\left\|\mathcal{P}_{L}^{\perp} \mathbf{X} \mathcal{P}_{R}^{\perp}\right\|_{F}^{2} \leq \frac{\varsigma}{1-\varsigma}\left\|\mathcal{P}_{L} \mathbf{X} \mathcal{P}_{R}\right\|_{F}^{2}\right) \triangleq P_{3}
\end{aligned}
$$

The random variables $\mathcal{P}_{L} \mathbf{X} \mathcal{P}_{R}$ and $\mathcal{P}_{L}^{\perp} \mathbf{X} \mathcal{P} \frac{1}{R}$ are independent zero-mean Gaussian variables. Let $\left\{w_{i, j}\right\}$, and $\left\{z_{k, l}\right\}$ be i.i.d. $\mathcal{N}(0,1)$ variables, where $1 \leq i, j \leq r, 1 \leq k \leq p-r$ and $1 \leq l \leq q-r$. This gives us that [32]

$$
\mathrm{P}_{3}=\operatorname{Pr}\left(\frac{\sum_{k=1}^{p-r} \sum_{l=1}^{q-r} \lambda_{L, r+k} \lambda_{L, r+l} w_{k, l}^{2}}{\sum_{1 \leq i, j \leq r} \lambda_{L, i} \lambda_{R, j} z_{i, j}^{2}} \leq \frac{\varsigma}{1-\varsigma}\right) .
$$

Using that

$$
\begin{aligned}
& \sum_{1 \leq i, j \leq r} \lambda_{L, i} \lambda_{R, j} z_{i, j}^{2} \geq \lambda_{L, r} \lambda_{R, r} \sum_{1 \leq i, j \leq r} z_{i, j}^{2}, \\
& \sum_{k=1}^{p-r} \sum_{l=1}^{q-r} \lambda_{L, r+k} \lambda_{R, r+l} w_{k, l}^{2} \leq \lambda_{L, r+1} \lambda_{R, r+1} \sum_{k=1}^{p-r} \sum_{l=1}^{q-r} w_{k, l}^{2},
\end{aligned}
$$

we find that

$$
\mathrm{P}_{3} \geq \operatorname{Pr}\left(\frac{\sum_{k=1}^{p-r} \sum_{l=1}^{q-r} w_{k, l}^{2}}{\sum_{1 \leq i, j \leq r} z_{i, j}^{2}} \leq \frac{\varsigma}{1-\varsigma} \frac{\lambda_{L, r} \lambda_{R, r}}{\lambda_{L, r+1} \lambda_{R, r+1}}\right) .
$$

Since

$$
\frac{\frac{1}{(p-r)(q-r)} \sum_{k=1}^{p-r} \sum_{l=1}^{q-r} w_{k, l}^{2}}{\frac{1}{r^{2}} \sum_{1 \leq i, j \leq r} z_{i, j}^{2}}
$$

is F-distributed, we find that

$$
P_{2} \geq P_{3} \geq 1-\frac{B\left(\frac{r^{2}}{2}, \frac{(p-r)(q-r)}{2}, \epsilon_{2}\right)}{B\left(\frac{r^{2}}{2}, \frac{(p-r)(q-r)}{2}\right)} \geq 1-C_{r, p, q} \epsilon_{2}^{\frac{r^{2}}{2}}
$$

where

$$
\begin{aligned}
& C_{r, p, q}^{-1}=\frac{r^{2}}{2} B\left(\frac{r^{2}}{2}, \frac{(p-r)(q-r)}{2}\right), \\
& \epsilon_{2}=\frac{1}{1+\frac{\varsigma}{1-\varsigma} \frac{\lambda_{L, r}}{\lambda_{L, r+1}} \frac{\lambda_{R, r}}{\lambda_{R, r+1}}}
\end{aligned}
$$

## D. The EM help function

The EM help function $Q\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\prime}\right)$ is given by

$$
\begin{aligned}
& Q\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\prime}\right)=\mathcal{E}_{\mathbf{X} \mid \mathbf{y}, \boldsymbol{\Theta}^{\prime}}[\log p(\mathbf{X} \mid \mathbf{y}, \boldsymbol{\Theta})]=c+\frac{m}{2} \log \beta \\
& -\frac{\beta}{2} \mathcal{E}\left[\|\mathbf{y}-\mathbf{A v e c}(\mathbf{X})\|_{2}^{2}-\frac{1}{2} \mathcal{E}\left[\operatorname{tr}\left(\boldsymbol{\alpha}_{L} \mathbf{X} \boldsymbol{\alpha}_{R} \mathbf{X}^{\top}\right)\right]\right. \\
& +\frac{q}{2} \log \left|\boldsymbol{\alpha}_{L}\right|+\frac{p}{2} \log \left|\boldsymbol{\alpha}_{R}\right|
\end{aligned}
$$

where $c$ is a constant. Using that

$$
\begin{aligned}
& \mathcal{E}\left[\|\mathbf{y}-\mathbf{A} \operatorname{vec}(\mathbf{X})\|_{2}^{2}\right]=\|\mathbf{y}\|_{2}^{2}-2 \mathbf{y}^{\top} \mathbf{A} \operatorname{vec}(\hat{\mathbf{X}}) \\
& +\operatorname{tr}\left(\mathbf{A}^{\top} \mathbf{A}\left(\operatorname{vec}(\hat{\mathbf{X}}) \operatorname{vec}(\hat{\mathbf{X}})^{\top}+\boldsymbol{\Sigma}^{\prime}\right)\right) \\
& =\|\mathbf{y}-\mathbf{A} \operatorname{vec}(\hat{\mathbf{X}})\|_{2}^{2}+\operatorname{tr}\left(\mathbf{A}^{\top} \mathbf{A} \boldsymbol{\Sigma}^{\prime}\right)
\end{aligned}
$$

and

$$
\begin{aligned}
& \mathcal{E}\left[\operatorname{tr}\left(\boldsymbol{\alpha}_{L} \mathbf{X} \boldsymbol{\alpha}_{R} \mathbf{X}^{\top}\right)\right] \\
& =\operatorname{tr}\left(\left(\boldsymbol{\alpha}_{R} \otimes \boldsymbol{\alpha}_{L}\right)\left(\operatorname{vec}(\hat{\mathbf{X}}) \operatorname{vec}(\hat{\mathbf{X}})^{\top}+\boldsymbol{\Sigma}^{\prime}\right)\right) \\
& =\operatorname{tr}\left(\boldsymbol{\alpha}_{L} \hat{\mathbf{X}} \boldsymbol{\alpha}_{R} \hat{\mathbf{X}}^{\top}\right)+\operatorname{tr}\left(\left(\boldsymbol{\alpha}_{R} \otimes \boldsymbol{\alpha}_{L}\right) \boldsymbol{\Sigma}^{\prime}\right)
\end{aligned}
$$

we recover the expression (28) for the EM help function.

## E. Details for the RSVM with the Schatten s-norm penalty

We here set $\mathbf{S}=\epsilon \mathbf{I}_{q}$ to keep the derivation more general. The regularized Schatten $s$-norm penalty is given by

$$
\tilde{g}(\mathbf{Z})=\operatorname{tr}\left(\left(\mathbf{X}^{\top} \mathbf{X}+\mathbf{S}\right)^{s / 2}\right)
$$

For the concave conjugate formula (10) we find that the minimum over $\mathbf{Z}$ occurs when

$$
\boldsymbol{\alpha}-\frac{s}{2}(\mathbf{Z}+\mathbf{S})^{s / 2-1}=\mathbf{0} .
$$

Solving for $\mathbf{Z}$ gives that

$$
\tilde{K}(\boldsymbol{\alpha})=-\operatorname{tr}(\boldsymbol{\alpha} \mathbf{S})-\frac{2-s}{s}\left(\frac{2}{s}\right)^{-2 /(2-s)} \operatorname{tr}\left(\boldsymbol{\alpha}^{-2 /(2-s)}\right),
$$

which results in (12).
Using (28), we find that the minimum of (27) for the Schatten $s$-norm occurs when

$$
\hat{\mathbf{X}} \boldsymbol{\alpha}_{R} \hat{\mathbf{X}}^{\top}+\tilde{\boldsymbol{\Sigma}}_{R}-\left(\frac{2}{s}\right)^{-s /(2-s)} \boldsymbol{\alpha}_{L}^{-2 /(2-s)}=\mathbf{0}
$$

By solving for $\boldsymbol{\alpha}_{L}$ we get (30). The update equation for $\boldsymbol{\alpha}_{R}$ is derived in a similar manner.

## F. Details for RSVM-LD

The log-determinant penalty is given by

$$
g(\mathbf{X})=\nu \log |\mathbf{Z}+\mathbf{S}| .
$$

For the concave conjugate formula (10) we find that the minimum over $\mathbf{Z}$ occurs when

$$
\boldsymbol{\alpha}-\nu(\mathbf{Z}+\mathbf{S})^{-1}=\mathbf{0} .
$$

Solving for $\mathbf{Z}$ gives that

$$
\tilde{K}(\boldsymbol{\alpha})=-\operatorname{tr}(\boldsymbol{\alpha} \mathbf{S})+\nu \log |\boldsymbol{\alpha}|+\nu p-\nu \log \nu .
$$

By removing the constants we recover (14).
Using (28), we find that the minimum of (27) with respect to $\alpha_{L}$ for the log-determinant penalty occurs when

$$
\hat{\mathbf{X}} \boldsymbol{\alpha}_{R} \hat{\mathbf{X}}^{\top}+\tilde{\boldsymbol{\Sigma}}_{R}+\mathbf{S}_{L}-\nu \boldsymbol{\alpha}_{L}^{-1}=\mathbf{0}
$$

Solving for $\alpha_{L}$ gives us (31). The derivation of the update equation for $\boldsymbol{\alpha}_{R}$ is found in a similar way.

## G. Update equations of VB-1 for LRMR

Here we generalize the update equations of the Variational Bayes method from [11] to LRMR, referred to as VB-1 in section V-A. Similar methods were used in [15]-[18]. The VB-1 method factorizes the matrix $\mathbf{X} \in \mathbb{R}^{p \times q}$ as

$$
\mathbf{X}=\mathbf{F B}^{\top},
$$

were the column vectors of $\mathbf{F}=\left[\mathbf{f}_{1} \mathbf{f}_{2} \ldots \mathbf{f}_{r}\right] \in \mathbb{R}^{p \times r}, \mathbf{B}=$ $\left[\mathbf{b}_{1} \mathbf{b}_{2} \ldots \mathbf{b}_{r}\right] \in \mathbb{R}^{q \times r}(r \leq \min (p, q)$ is a user parameter $)$ are given Gaussian priors as

$$
\begin{aligned}
& p(\mathbf{F} \mid \boldsymbol{\gamma})=\prod_{i=1}^{r} \mathcal{N}\left(\mathbf{f}_{i} \mid \mathbf{0}, \gamma_{i}^{-1} \mathbf{I}_{p}\right) \\
& p(\mathbf{B} \mid \boldsymbol{\gamma})=\prod_{j=1}^{r} \mathcal{N}\left(\mathbf{b}_{j} \mid \mathbf{0}, \gamma_{j}^{-1} \mathbf{I}_{q}\right),
\end{aligned}
$$

where $\gamma_{i}>0$ are the precisions of $\mathbf{f}_{i}$ and $\mathbf{b}_{i}$. We usually set $r=\operatorname{rank}(\mathbf{X})$ when the rank is known, otherwise $r$ can be used to upper bound the rank of $\hat{\mathbf{X}}=\hat{\mathbf{F}} \hat{\mathbf{B}}^{\top}$. The additive noise in (3) is modeled as a zero-mean white Gaussian with the unknown precision $\beta>0$. The precisions are given Gamma and Jeffreys priors as

$$
\begin{aligned}
& p\left(\gamma_{i}\right) \propto \gamma_{i}^{a-1} \exp \left(-b \gamma_{i}\right) \\
& p(\beta) \propto \beta^{-1}
\end{aligned}
$$

In the variational Bayes framework, blocks of variables are assumed to have independent posterior distributions allowing to approximate the posterior. Assume that we want to approximate a distribution $p(\mathbf{z})$ using variational Bayes. Let $\mathbf{z}_{I}$ denote the variables with indices's in a set $I$, the variational Bayes approximates the distribution $p\left(\mathbf{z}_{I}\right)$ by $q\left(\mathbf{z}_{I}\right)$ as [24]

$$
\log q\left(\mathbf{z}_{I}\right)=\mathcal{E}_{\mathbf{z}_{I^{c}} \mid \mathbf{z}_{I}}\left[\log p\left(\mathbf{z}_{I}, \mathbf{z}_{I^{c}}\right)\right]+\text { constant }
$$

Different choices of blocks of parameters can be made, we here chose to use independent rows in $\mathbf{F}$ and $\mathbf{B}$ (as in [11]) since it gives good (empirical) performance. We here use $\mathbf{A}_{k}$ to denote the $k$ 'th sensing matrix in (2), $\left[\mathbf{A}_{k}\right]_{. i}$ to denote the
$i$ 'th column vector of $\mathbf{A}_{k},\left[\mathbf{A}_{k}\right]_{i}$. to denote the $i$ 'th row vector of $\mathbf{A}_{k}$ and $\left[\mathbf{A}_{k}\right]_{i j}$ to denote the $(i, j)$ 'th component of $\mathbf{A}_{k}$. We also set $\boldsymbol{\Gamma}=\operatorname{diag}\left(\gamma_{1}, \gamma_{2}, \ldots, \gamma_{r}\right)$.

Given means $\hat{\boldsymbol{\gamma}}, \hat{\beta}$ of $\gamma$ and $\beta$, that $\mathbf{b}_{k} \sim \mathcal{N}\left(\hat{\mathbf{b}}_{k}, \boldsymbol{\Sigma}_{k}^{(B)}\right)$ for all $k$ and $\mathbf{f}_{j} \sim \mathcal{N}\left(\hat{\mathbf{f}}_{j}, \boldsymbol{\Sigma}_{j}^{(F)}\right)$ for all $j \neq i$, we find that

$$
\begin{aligned}
& \log q\left(\mathbf{f}_{i}\right)=-\frac{\hat{\beta}}{2}\left(\sum_{k=1}^{m} y_{k}^{2}-2 \mathbf{f}_{i}^{\top} \hat{\mathbf{B}}^{\top}\left[\mathbf{A}_{k}\right]_{i .} y_{k}\right. \\
& +2 \sum_{j \neq i} \mathbf{f}_{i}^{\top} \mathcal{E}\left[\mathbf{B}^{\top}\left[\mathbf{A}_{k}\right]_{i .}\left[\mathbf{A}_{k}\right]_{j .}^{\top} \mathbf{B}\right] \hat{\mathbf{f}}_{j} \\
& \left.+\mathbf{f}_{i}^{\top} \mathcal{E}\left[\mathbf{B}^{\top}\left[\mathbf{A}_{k}\right]_{i .}\left[\mathbf{A}_{k}\right]_{i .}^{\top} \mathbf{B}\right] \mathbf{f}_{i}\right)-\frac{1}{2} \mathbf{f}_{i}^{\top} \boldsymbol{\Gamma} \mathbf{f}_{i}+\text { constant }
\end{aligned}
$$

where

$$
\begin{aligned}
\mathcal{E}\left[\mathbf{B}^{\top}\left[\mathbf{A}_{k}\right]_{i .}\left[\mathbf{A}_{k}\right]_{j .}^{\top} \mathbf{B}\right]= & \hat{\mathbf{B}}^{\top}\left[\mathbf{A}_{k}\right]_{i .}\left[\mathbf{A}_{k}\right]_{j .}^{\top} \hat{\mathbf{B}} \\
& +\sum_{c}\left[\mathbf{A}_{k}\right]_{i c}\left[\mathbf{A}_{k}\right]_{j c} \boldsymbol{\Sigma}_{d}^{(B)}
\end{aligned}
$$

This gives us that $\mathbf{f}_{i} \sim \mathcal{N}\left(\hat{\mathbf{f}}_{i}, \boldsymbol{\Sigma}_{i}^{(F)}\right)$ with

$$
\begin{aligned}
\hat{\mathbf{f}}_{i}=\hat{\beta} \boldsymbol{\Sigma}_{i}^{(F)} & \left(\hat{\mathbf{B}}^{\top} \sum_{k} y_{k}\left[\mathbf{A}_{k}\right]_{i .}\right. \\
& \left.\left.-\sum_{j \neq i, k} \mathcal{E}\left[\mathbf{B}^{\top}\left[\mathbf{A}_{k}\right]_{i .} \mathbf{A}_{k}\right]_{j .}^{\top} \mathbf{B}\right] \hat{\mathbf{f}}_{j}\right) \\
\boldsymbol{\Sigma}_{i}^{(F)}= & \left(\hat{\beta} \sum_{k} \mathcal{E}\left[\mathbf{B}^{\top}\left[\mathbf{A}_{k}\right]_{i .}\left[\mathbf{A}_{k}\right]_{i .}^{\top} \mathbf{B}\right]+\Gamma\right)^{-1}
\end{aligned}
$$

Similarly, when the distributions of the other variables are fixed, we get that $\mathbf{b}_{i} \sim \mathcal{N}\left(\hat{\mathbf{b}}_{i}, \boldsymbol{\Sigma}_{i}^{(B)}\right)$ with

$$
\begin{aligned}
\hat{\mathbf{b}}_{i}= & \hat{\beta} \boldsymbol{\Sigma}_{i}^{(B)}\left(\hat{\mathbf{F}}^{\top} \sum_{k} y_{k}\left[\mathbf{A}_{k}\right]_{. i}\right. \\
& \left.-\sum_{j \neq i, k} \mathcal{E}\left[\mathbf{F}^{\top}\left[\mathbf{A}_{k}\right]_{. i}\left[\mathbf{A}_{k}\right]_{. j}^{\top} \mathbf{F}\right] \hat{\mathbf{b}}_{j}\right), \\
\boldsymbol{\Sigma}_{i}^{(B)}= & \left(\hat{\beta} \sum_{k} \mathcal{E}\left[\mathbf{F}^{\top}\left[\mathbf{A}_{k}\right]_{. i}\left[\mathbf{A}_{k}\right]_{. i}^{\top} \mathbf{F}\right]+\Gamma\right)^{-1},
\end{aligned}
$$

where now

$$
\begin{aligned}
\mathcal{E}\left[\mathbf{F}^{\top}\left[\mathbf{A}_{k}\right]_{. i}\left[\mathbf{A}_{k}\right]_{. j}^{\top} \mathbf{F}\right]= & \hat{\mathbf{F}}^{\top}\left[\mathbf{A}_{k}\right]_{. i}\left[\mathbf{A}_{k}\right]_{. j} \hat{\mathbf{F}} \\
& +\sum_{d}\left[\mathbf{A}_{k}\right]_{d i}\left[\mathbf{A}_{k}\right]_{d j} \boldsymbol{\Sigma}_{d}^{(F)} .
\end{aligned}
$$

We also find that the precisions $\gamma_{i}$ are Gamma distributed with posterior parameters

$$
\begin{aligned}
\hat{a}_{i} & =\frac{p+q+2 a}{2} \\
\hat{b}_{i} & =\frac{1}{2}\left(\left\|\hat{\mathbf{f}}_{i}\right\|_{2}^{2}+\left\|\hat{\mathbf{b}}_{i}\right\|_{2}^{2}+\operatorname{tr}\left(\boldsymbol{\Sigma}_{i}^{(F)}\right)+\operatorname{tr}\left(\boldsymbol{\Sigma}_{i}^{(B)}\right)+2 b\right) .
\end{aligned}
$$

The posterior mean of $\gamma_{i}$ is $\hat{\gamma}_{i}=\hat{a}_{i} / \hat{b}_{i}$. Similary we find that the posterior distribution of $\beta$ is $\operatorname{Gamma}(\hat{c}, \hat{d})$ with

$$
\begin{aligned}
\hat{c} & =m / 2, \\
\hat{d} & =\frac{1}{2}\left[\left\|\mathbf{y}-\operatorname{vec}\left(\hat{\mathbf{F}} \hat{\mathbf{B}}^{\top}\right)\right\|_{2}^{2}+\sum_{i, k}\left(\hat{\mathbf{f}}_{i}^{\top} \mathbf{A}_{k} \boldsymbol{\Sigma}_{i}^{(B)} \mathbf{A}_{k}^{\top} \hat{\mathbf{f}}_{i}\right.\right. \\
& \left.\left.+\hat{\mathbf{b}}_{i}^{\top} \mathbf{A}_{k}^{\top} \boldsymbol{\Sigma}_{i}^{(F)} \mathbf{A}_{k} \hat{\mathbf{b}}_{i}+\operatorname{tr}\left(\boldsymbol{\Sigma}_{i}^{(F)} \mathbf{A}_{k} \boldsymbol{\Sigma}_{i}^{(B)} \mathbf{A}_{k}^{\top}\right)\right)\right] .
\end{aligned}
$$

The posterior mean of $\beta$ is thus $\hat{\beta}=\hat{c} / \hat{d}$.

## References

[1] E.J. Candès and Y. Plan, "Matrix Completion With Noise," Proceedings of the IEEE, vol. 98, no. 6, pp. 925-936, June 2010.
[2] E.J. Candès and B. Recht, "Exact matrix completion via convex optimization," Foundations of Computational mathematics, vol. 9, no. 6 pp . 717-772, December 2009.
[3] M. Fazel, "Matrix rank minimization with applications," Diss. PhD thesis, Stanford University, 2002.
[4] D. Zachariah, M. Sundin, M. Jansson and S. Chatterjee, "Alternating Least-Squares for Low-Rank Matrix Reconstruction," IEEE Signal Processing Letters, vol. 19, no. 4, pp. 231-234, April 2012.
[5] K. Yu, J. Lafferty, S. Zhu and Y. Gong, "Large-scale collaborative prediction using a nonparametric random effects model," Proceedings of the 26th Annual International Conference on Machine Learning. ACM, 2009.
[6] J. Fan, Y. Fan, and J. Lv, "High dimensional covariance matrix estimation using a factor model," Journal of Econometrics, vol. 147, issue 1, pp. 186-197, 2008.
[7] M. Fornasier, H. Rauhut and R. Ward, "Low-rank matrix recovery via iteratively reweighted least squares minimization," SIAM Journal on Optimization, vol. 21, no. 4, pp.1614-1640, 2011.
[8] E.J. Candès and T. Tao, "The Power of Convex Relaxation: Near-Optimal Matrix Completion," IEEE Transactions on Information Theory, vol. 56, no. 5, pp. 2053-2080, May 2010.
[9] C. Jian-Feng, E.J. Candès and Z. Shen. "A singular value thresholding algorithm for matrix completion," SIAM Journal on Optimization, vol. 20, no. 4, pp. 1956-1982, March 2010.
[10] K. Lee, and Y. Bresler. "Admira: Atomic decomposition for minimum rank approximation," IEEE Transactions on Information Theory, vol. 56, no. 9, pp. 4402-4416, September 2010.
[11] S.D. Babacan, M. Luessi, R. Molina and A.K. Katsaggelos, "Sparse Bayesian Methods for Low-Rank Matrix Estimation," IEEE Transactions on Signal Processing, vol. 60, no. 8, pp. 3964-3977, August 2012.
[12] D. Wipf, "Non-Convex Rank Minimization via an Empirical Bayesian Approach," Uncertainty in Artificial Intelligence (UAI), 2012.
[13] D. Xinghao, L. He, and L. Carin, "Bayesian robust principal component analysis," IEEE Transactions on Image Processing, vol. 20, no. 12, pp. 3419-3430, 2011.
[14] T. Raiko, A. Ilin and J. Karhunen, "Principal component analysis for large scale problems with lots of missing values," Machine Learning: ECML 2007. Springer Berlin Heidelberg, 2007. 691-698.
[15] Y.J. Lim and Y.W. Teh, "Variational Bayesian approach to movie rating prediction," Proceedings of KDD Cup and Workshop, Vol. 7, 2007.
[16] S. Nakajima, M. Sugiyama and S.D. Babacan, "Variational Bayesian sparse additive matrix factorization," Machine Learning, vol. 92, no. 2-3, pp. 319-347, 2013.
[17] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," Advances in neural information processing systems, pp.1257-1264, 2007.
[18] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," Proceedings of the 25th international conference on Machine learning (ACM), 2008.
[19] N. Srebro and R. Salakhutdinov, "Collaborative filtering in a nonuniform world: Learning with the weighted trace norm," Advances in Neural Information Processing Systems, 2010.
[20] J.L. Herlocker, J.A. Konstan, A. Borchers and J. Riedl, "An algorithmic framework for performing collaborative filtering," Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 1999.
[21] S. Boyd and L. Vandenberghe, "Convex optimization," Cambridge, Cambridge university press, 2009.
[22] G. Tang and A. Nehorai, "Lower Bounds on the Mean-Squared Error of Low-Rank Matrix Reconstruction," IEEE Transactions on Signal Processing, vol. 59, no. 10, pp. 4559-4571, October 2011.
[23] I. Johnstone and A.Y. Lu, "On consistency and sparsity for principal components analysis in high dimensions," Journal of the American Statistical Association vol. 104, no. 486, 2009.
[24] C.M. Bishop, "Pattern recognition and machine learning," New York, Springer, 2006.
[25] J.O. Berger, "Statistical decision theory and Bayesian analysis," Springer Verlag New York Inc., 1985.
[26] A. Gelman, J.B. Carlin, H.S. Stern and D.B. Rubin, "Bayesian data analysis. Vol. 2", Chapman \& Hall, London, 2014.
[27] M.E. Tipping, "Sparse Bayesian learning and the relevance vector machine," The journal of machine learning research, vol. 1, pp. 211-244, January 2001.
[28] M. Tipping and A. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models," Proceedings of the ninth international workshop on artificial intelligence and statistics, vol. 1, no. 3, 2003.
[29] Z. Zhang and B.D. Rao, "Extension of sbl algorithms for the recovery of block sparse signals with intra-block correlation," IEEE Transactions on Signal Processing, vol. 61, no. 8, pp. 2009-2015, April 2013.
[30] Z. Zhang and B.D. Rao, "Sparse Signal Recovery With Temporally Correlated Source Vectors Using Sparse Bayesian Learning," IEEE Journal of Selected Topics in Signal Processing, vol. 5, no. 5, pp. 912-926, Sept. 2011.
[31] E.J. Candès, J.K. Romberg and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," Communications on pure and applied mathematics, vol. 59, no. 8, pp. 1207-1223, March 2006.
[32] A.K. Gupta and D.K. Nagar, "Matrix variate distributions", Vol. 104, CRC Press, 1999.
[33] A. Terras and A. Terras, "Harmonic analysis on symmetric spaces and applications," vol. 1985, Berlin, Springer, 1985.
[34] D. MacKay, "Information theory, inference, and learning algorithms," vol. 7, Cambridge, Cambridge university press, 2003.
[35] A.P. Dempster, N.M. Laird and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," Journal of the Royal Statistical Society, Series B (methodological), pp. 1-38, 1977.
[36] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming," version 2.0 beta, http://cvxr.com/cvx, September 2013.
[37] M. Sundin, S. Chatterjee, M. Jansson and C.R. Rojas, "Relevance Singular Vector Machine for low-rank matrix sensing", SPCOM 2014. Avaliable online from http://arxiv.org/abs/1407.0013.


[^0]:    The authors are with the ACCESS Linnaeus Centre, KTH Royal Institute of Technology, Stockholm. E-mail: masundi@kth.se, crro@kth.se, magnus.jansson@ee.kth.se and sach@kth.se.
    This work was supported in part by the Swedish Research Council under contract 2015-05484.
    Parts of this paper was presented at the International Conference on Signal Processing and Communications (SPCOM), July 2014, Bangalore, India [37].

