Fast Low-Rank Bayesian Matrix Completion With Hierarchical Gaussian Prior Models

Linxiao Yang, Jun Fang, *Member, IEEE*, Huiping Duan, Hongbin Li, *Senior Member, IEEE*, and Bing Zeng, *Fellow, IEEE*

***Abstract*—The problem of low-rank matrix completion is considered in this paper. To exploit the underlying low-rank structure of the data matrix, we propose a hierarchical Gaussian prior model, where columns of the low-rank matrix are assumed to follow a Gaussian distribution with zero mean and a common precision matrix, and a Wishart distribution is specified as a hyperprior over the precision matrix. We show that such a hierarchical Gaussian prior has the potential to encourage a low-rank solution. Based on the proposed hierarchical prior model, we develop a variational Bayesian matrix completion method, which embeds the generalized approximate massage passing technique to circumvent cumbersome matrix inverse operations. Simulation results show that our proposed method demonstrates superiority over some state-of-the-art matrix completion methods.**

***Index Terms*—Matrix completion, low-rank Bayesian learning, generalized approximate massage passing.**

I. Introduction

THE problem of recovering a partially observed matrix, which is referred to as matrix completion, arises in a variety of applications, including recommender systems [1]–[3], genotype prediction [4], [5], image classification [6], [7], network traffic prediction [8], and image imputation [9]. Low-rank matrix completion, which is empowered by the fact that many real-world data lie in an intrinsically low dimensional subspace, has attracted much attention over the past few years. Mathematically, a canonical form of the low-rank matrix completion problem can be presented as

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \text{rank}(\boldsymbol{X}) \\ \text{s.t.} \quad & \boldsymbol{Y} = \boldsymbol{\Omega} * \boldsymbol{X} \end{aligned} \tag{1}$$

Manuscript received August 25, 2017; revised January 20, 2018; accepted March 6, 2018. Date of publication March 16, 2018; date of current version April 18, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Lei Huang. This work was supported by the National Science Foundation of China under Grants 61522104 and U1530154. *(Corresponding author: Jun Fang.)*

L. Yang and J. Fang are with the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: yanglinxiao0929@hotmail.com; JunFang@uestc.edu.cn).

H. Duan and B. Zeng are with the School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: huipingduan@uestc.edu.cn; eezeng@uestc.edu.cn).

H. Li is with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: Hongbin.Li@stevens.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2018.2816575

where $\boldsymbol{X} \in \mathbb{R}^{M\times N}$ is an unknown low-rank matrix, $\boldsymbol{\Omega} \in \{0,1\}^{M\times N}$ is a binary matrix that indicates which entries of $\boldsymbol{X}$ are observed, $*$ denotes the Hadamard product, and $\boldsymbol{Y} \in \mathbb{R}^{M\times N}$ is the observed matrix. It has been shown that the low-rank matrix $\boldsymbol{X}$ can be exactly recovered from (1) under some mild conditions [10]. Nevertheless, minimizing the rank of a matrix is an NP-hard problem and no known polynomial-time algorithms exist. To overcome this difficulty, alternative low-rank promoting functionals were proposed. Among them, the most popular alternative is the nuclear norm which is defined as the sum of the singular values of a matrix. Replacing the rank function with the nuclear norm yields the following convex optimization problem

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \|\boldsymbol{X}\|_* \\ \text{s.t.} \quad & \boldsymbol{Y} = \boldsymbol{\Omega} * \boldsymbol{X} \end{aligned} \tag{2}$$

It was proved that the nuclear norm is the tightest convex envelope of the matrix rank, and the theoretical recovery guarantee for (2) under both noiseless and noisy cases was provided in [10]–[13]. To solve (2), a number of computationally efficient methods were developed. A well-known method is the singular value thresholding method which was proposed in [14]. Another efficient method was proposed in [15], in which an augmented Lagrange multiplier technique was employed. Apart from convex relaxation, non-convex surrogate functions, such as the log-determinant function, were also introduced to replace the rank function [16]–[19]. Non-convex methods usually claim better recovery performance, since non-convex surrogate functions behaves more like the rank function than the nuclear norm. It is noted that for both convex methods and non-convex methods, one needs to meticulously select some regularization parameters to properly control the tradeoff between the matrix rank and the data fitting error when noise is involved. However, due to the lack of the knowledge of the noise variance and the rank, it is usually difficult to determine appropriate regularization parameters.

Another important class of low-rank matrix completion methods are Bayesian methods [20]–[24], which model the problem in a Bayesian framework and have the ability to achieve automatic balance between the matrix rank and the fitting error. Specifically, in [20], the low-rank matrix is expressed as a product of two factor matrices, i.e., $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{B}^T$, and the matrix completion problem is translated to searching for these two factor matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. To encourage a low-rank solution, sparsity-promoting priors [25] are placed on the columns of two

factor matrices, which aims to promote structured-sparse factor matrices with only a few non-zero columns, and in turn leads to a low-rank matrix $\boldsymbol{X}$. Nevertheless, this Bayesian method updates the factor matrices in a row-by-row fashion and needs to perform a number of matrix inverse operations at each iteration. To address this issue, a bilinear generalized approximate message passing (GAMP) method was developed to learn the two factor matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ [22], [23], without involving any matrix inverse operations. This method, however, cannot automatically determine the matrix rank and needs to try out all possible values of the rank. Note that this factorization-based matrix completion approach was also studied in a deterministic framework, and a number of algorithms which resort to the alternating minimization technique have been developed, e.g., [26]–[28]. As indicated earlier, these optimization-based methods require to select some appropriate regularization parameter to strike a good balance between the matrix rank and the data fitting error. Recently, a new Bayesian prior model was proposed in [24], in which columns of the low-rank matrix $\boldsymbol{X}$ follow a zero mean Gaussian distribution with an unknown deterministic covariance matrix that can be estimated via Type II maximum likelihood. It was shown that maximizing the marginal likelihood function yields a low-rank covariance matrix, which implies that the prior model has the ability to promote a low-rank solution. A major drawback of this method is that it requires to perform an inverse of an $MN \times MN$ matrix at each iteration, and thus has a cubic complexity in terms of the problem size. This high computational cost prohibits its application to many practical problems.

In this paper, we develop a new Bayesian method for low-rank matrix completion. To exploit the underlying low-rank structure of the data matrix, a low-rank promoting hierarchical Gaussian prior model is proposed. Specifically, columns of the low-rank matrix $\boldsymbol{X}$ are assumed to be mutually independent and follow a common Gaussian distribution with zero mean and a precision matrix. The precision matrix is treated as a random parameter, with a Wishart distribution specified as a hyperprior over it. We show that such a hierarchical Gaussian prior model has the potential to encourage a low-rank solution. The GAMP technique is employed and embedded in the variational Bayesian (VB) inference, which results in an efficient VB-GAMP algorithm for matrix completion. Note that due to the non-factorizable form of the prior distribution, the GAMP technique cannot be directly used. To address this issue, we construct a carefully devised surrogate problem whose posterior distribution is exactly the one required for VB inference. Meanwhile, the surrogate problem has factorizable prior and noise distributions such that the GAMP can be directly applied to obtain an approximate posterior distribution. Such a trick helps achieve a substantial computational complexity reduction, and makes it possible to successfully apply the proposed method to solve large-scale matrix completion problems.

The rest of the paper is organized as follows. In Section II, we introduce a hierarchical Gaussian prior model for low-rank matrix completion. Based on this hierarchical model, a variational Bayesian method is developed in Section III. In Section IV, a GAMP-VB method is proposed to reduce the computational complexity of the proposed algorithm. Simulation results are provided in Section V, followed by concluding remarks in Section VI.

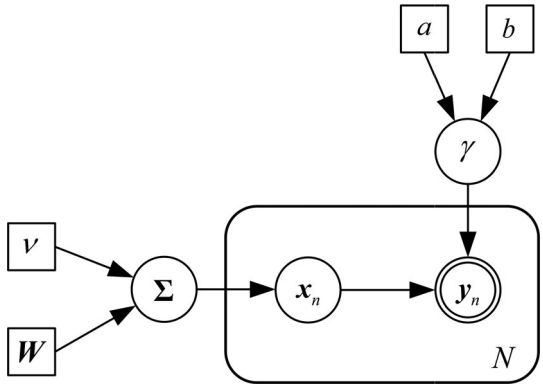

Fig. 1. Proposed low-rank promoting hierarchical Gaussian prior model, in which double circles denote the observable variable, single circles denote the hidden variable, and the boxes denote pre-specified hyperparameters.

II. Bayesian Modeling

In the presence of noise, the canonical form of the matrix completion problem can be formulated as

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \text{rank}(\boldsymbol{X}) \\ \text{s.t.} \quad & \boldsymbol{Y} = \boldsymbol{\Omega} * (\boldsymbol{X} + \boldsymbol{E}) \end{aligned} \tag{3}$$

where $\boldsymbol{E}$ denotes the additive noise, and $\boldsymbol{\Omega} \in \{0,1\}^{M \times N}$ is a binary matrix that indicates which entries are observed. Without loss of generality, we assume $M \leq N$. As indicated earlier, minimizing the rank of a matrix is an NP-hard problem. In this paper, we consider modeling the matrix completion problem within a Bayesian framework.

We assume entries of $\boldsymbol{E}$ are independent and identically distributed (i.i.d.) random variables following a Gaussian distribution with zero mean and variance γ^{-1}. To learn γ, a Gamma hyperprior is placed over γ, i.e.,

$$p(\gamma) = \text{Gamma}(\gamma|a,b) = \Gamma(a)^{-1} b^a \gamma^{a-1} e^{-b\gamma} \tag{4}$$

where $\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt$ is the Gamma function. The parameters a and b are set to small values, e.g., 10^{-10}, which makes the Gamma distribution a non-informative prior.

To promote a low-rank solution of $\boldsymbol{X}$, we propose a two-layer hierarchical Gaussian prior model (see Fig. 1). Specifically, in the first layer, the columns of $\boldsymbol{X}$ are assumed mutually independent and follow a common Gaussian distribution:

$$p(\boldsymbol{X}|\boldsymbol{\Sigma}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{\Sigma}) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{0}, \boldsymbol{\Sigma}^{-1}) \tag{5}$$

where $\boldsymbol{x}_n$ denotes the nth column of $\boldsymbol{X}$, and $\boldsymbol{\Sigma} \in \mathbb{R}^{M \times M}$ is the precision matrix. The second layer specifies a Wishart

distribution as a hyperprior over the precision matrix $\boldsymbol{\Sigma}$:

$$p(\boldsymbol{\Sigma}) \propto |\boldsymbol{\Sigma}|^{\frac{\nu-M-1}{2}} \exp\left(-\frac{1}{2}\text{tr}(\boldsymbol{W}^{-1}\boldsymbol{\Sigma})\right) \tag{6}$$

where ν and $\boldsymbol{W} \in \mathbb{R}^{M\times M}$ denote the degrees of freedom and the scale matrix of the Wishart distribution, respectively. Note that the constraint $\nu > M-1$ for the standard Wishart distribution can be relaxed to $\nu > 0$ if an improper prior is allowed, e.g., [29]. In Bayesian inference, improper prior distributes can often be used provided that the corresponding posterior distribution can be correctly normalized [30].

The Gaussian-inverse Wishart prior has the potential to encourage a low-rank solution. To illustrate this property, we integrate out the precision matrix $\boldsymbol{\Sigma}$ and obtain the marginal distribution of $\boldsymbol{X}$ as (details of the derivation can be found in Appendix A)

$$\begin{aligned} p(\boldsymbol{X}) &= \int \prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{\Sigma})p(\boldsymbol{\Sigma})d\boldsymbol{\Sigma} \\ &\propto |\boldsymbol{W}^{-1} + \boldsymbol{X}\boldsymbol{X}^T|^{-\frac{\nu+N}{2}} \end{aligned} \tag{7}$$

From (7), we have

$$\log p(\boldsymbol{X}) \propto -\log|\boldsymbol{X}\boldsymbol{X}^T + \boldsymbol{W}^{-1}| \tag{8}$$

If we choose $\boldsymbol{W} = \epsilon^{-1}\boldsymbol{I}$, and let ϵ be a small positive value, the log-marginal distribution becomes

$$\begin{aligned} \log p(\boldsymbol{X}) &\propto -\log|\boldsymbol{X}\boldsymbol{X}^T + \epsilon\boldsymbol{I}| \\ &= -\sum_{m=1}^{M}\log(\lambda_m + \epsilon) \end{aligned} \tag{9}$$

where λ_m denotes the mth eigenvalue of $\boldsymbol{X}\boldsymbol{X}^T$. Clearly, in this case, the prior $p(\boldsymbol{X})$ encourages a low-rank solution $\boldsymbol{X}$. This is because maximizing the prior distribution $p(\boldsymbol{X})$ is equivalent to minimizing $\sum_{m=1}^{M}\log(\lambda_m+\epsilon)$ with respect to $\{\lambda_m\}$. It is well known that the log-sum function $\sum_{m=1}^{M}\log(\lambda_m+\epsilon)$ is an effective sparsity-promoting functional which encourages a sparse solution of $\{\lambda_m\}$ [31]–[33]. As a result, the resulting matrix $\boldsymbol{X}$ has a low-rank structure. We note that the prior (9) placed on the latent matrix is quite similar to the volume-minimization cost function in [34]. This fact reveals that our prior model is closely related to the volume-minimization criterion, a criterion widely used for matrix factorization due its unique identifiability under some mild conditions. The volume-minimization criterion aims to minimize the volume of the convex hull spanned by the column vectors of the matrix. It has the potential to encourage a low-rank solution because the volume of the convex hull reduces to zero when the columns of the matrix lie in a low-dimensional subspace.

In addition to $\boldsymbol{W} = \epsilon^{-1}\boldsymbol{I}$, the parameter $\boldsymbol{W}$ can otherwise be devised in order to exploit additional prior knowledge about $\boldsymbol{X}$. For example, in some applications such as image inpainting, there is a spatial correlation among neighboring coefficients of $\boldsymbol{x}_n$. To capture the smoothness between neighboring coefficients, $\boldsymbol{W}$ can be set as [35]

$$\boldsymbol{W} = \boldsymbol{F}^T\boldsymbol{F} \tag{10}$$

where $\boldsymbol{F} \in \mathbb{R}^{M\times M}$ is a second-order difference operator with its (i,j)th entry given by

$$f_{i,j} = \begin{cases} -2, & i = j \\ 1, & |i-j| = 1 \\ 0, & \text{else} \end{cases} \tag{11}$$

Another choice of $\boldsymbol{W}$ to promote a smooth solution is the Laplacian matrix [36], i.e.,

$$\boldsymbol{W} = \boldsymbol{D} - \boldsymbol{A} + \hat{\epsilon}\boldsymbol{I} \tag{12}$$

where $\boldsymbol{A}$ is the adjacency matrix of a graph with its entries given by

$$a_{ij} = \exp\left(-\frac{|i-j|^2}{\theta^2}\right) \tag{13}$$

$\boldsymbol{D}$, referred to as the degree matrix, is a diagonal matrix with $d_{ii} = \sum_j a_{ij}$, and $\hat{\epsilon}$ is a small positive value to ensure $\boldsymbol{W}$ to be full rank.

It can be shown that $\boldsymbol{W}$ defined in (10) and (12) promotes a low-rank structure as well as smoothness of $\boldsymbol{X}$. To illustrate this, we first introduce the following lemma.

Lemma 1: For a positive-definite matrix $\boldsymbol{W} \in \mathbb{R}^{M\times M}$, the following equality holds valid

$$\log|\boldsymbol{X}\boldsymbol{X}^T + \boldsymbol{W}^{-1}| = \log|\boldsymbol{W}^{-1}| + \log|\boldsymbol{I} + \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}| \tag{14}$$

for any $\boldsymbol{X} \in \mathbb{R}^{M\times N}$.

Proof: See Appendix B. ■

From Lemma 1, we have

$$\begin{aligned} \log p(\boldsymbol{X}) &\propto -\log|\boldsymbol{X}\boldsymbol{X}^T + \boldsymbol{W}^{-1}| \\ &\propto -\log|\boldsymbol{I} + \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}| \\ &= -\sum_{n=1}^{N}\log(\tilde{\lambda}_n + 1) \end{aligned} \tag{15}$$

where $\tilde{\lambda}_n$ is the nth eigenvalue associated with $\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}$. We see that maximizing the prior distribution is equivalent to minimizing $\sum_{n=1}^{N}\log(\tilde{\lambda}_n+1)$ with respect to $\{\tilde{\lambda}_n\}$. As discussed earlier, this log-sum functional is a sparsity-promoting functional which encourages a sparse solution $\{\tilde{\lambda}_n\}$. As a result, the matrix $\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}$ has a low rank. Since $\boldsymbol{W}$ is full rank, this implies that $\boldsymbol{X}$ has a low-rank structure. On the other hand, notice that $\text{tr}(\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})$ is the first-order approximation of $\log|\boldsymbol{I} + \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}|$. Therefore minimizing $\log|\boldsymbol{I} + \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}|$ will reduce the value of $\text{tr}(\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})$. Clearly, for $\boldsymbol{W}$ defined in (10) and (12), a smoother solution results in a smaller value of $\text{tr}(\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})$ [35], [36]. Therefore when $\boldsymbol{W}$ is chosen to be (10) or (12), the resulting prior distribution $p(\boldsymbol{X})$ has the potential to encourage a low-rank and smooth solution.

Remarks: Our proposed hierarchical Gaussian prior model can be considered as a generalization of the prior model in [24]. Notice that in [24], the precision matrix in the prior model is

assumed to be a deterministic parameter, whereas it is treated as a random variable and assigned a Wishart prior distribution in our model. This generalization offers more flexibility in modeling the underlying latent matrix. As discussed earlier, the parameter $\boldsymbol{W}$ can be devised to capture additional prior knowledge about the latent matrix, and such a careful choice of $\boldsymbol{W}$ can help substantially improve the recovery performance, as corroborated by our experimental results.

III. Variational Bayesian Inference

A. Review of The Variational Bayesian Methodology

Before proceeding, we firstly provide a brief review of the variational Bayesian (VB) methodology (additional details can be found in [38]). In a probabilistic model, let $\boldsymbol{y}$ and $\boldsymbol{\theta}$ denote the observed data and the hidden variables, respectively. The marginal probability of the observed data can be decomposed into two terms [30]

$$\ln p(\boldsymbol{y}) = L(q) + \mathrm{KL}(q||p), \tag{16}$$

where

$$L(q) = \int q(\boldsymbol{\theta}) \ln \frac{p(\boldsymbol{y}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} \tag{17}$$

and

$$\mathrm{KL}(q||p) = -\int q(\boldsymbol{\theta}) \ln \frac{p(\boldsymbol{\theta}|\boldsymbol{y})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta}, \tag{18}$$

where $q(\boldsymbol{\theta})$ is an arbitrary probability density function, $\mathrm{KL}(q||p)$ is the Kullback-Leibler divergence [37] between $p(\boldsymbol{\theta}|\boldsymbol{y})$ and $q(\boldsymbol{\theta})$. Since $\mathrm{KL}(q||p) \geq 0$, $L(q)$ is a lower bound for $\ln p(\boldsymbol{y})$. Moreover, notice that the left hand side of (16) is a constant and thus independent of $q(\boldsymbol{\theta})$. Therefore maximizing $L(q)$ is equivalent to minimizing $\mathrm{KL}(q||p)$, and thus the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{y})$ can be approximated by $q(\boldsymbol{\theta})$ through maximizing $L(q)$.

The above decomposition (16) helps circumvent the difficulty of computing the posterior probability $p(\boldsymbol{\theta}|\boldsymbol{y})$, when it is computationally intractable. Specifically, we could assume some specific parameterized functional form for $q(\boldsymbol{\theta})$ and then maximize $L(q)$ with respect to the parameters of the distribution [38]. A particular form of $q(\boldsymbol{\theta})$ that has been widely used with great success is the factorized form over the component variables $\{\theta_i\}$ in $\boldsymbol{\theta}$ [38], i.e., $q(\boldsymbol{\theta}) = \prod_i q_i(\theta_i)$. We therefore can compute the posterior distribution approximation by finding $q(\boldsymbol{\theta})$ of the factorized form that maximizes the lower bound $L(q)$, which leads to [38]

$$q_i(\theta_i) = \frac{e^{\langle \ln p(\boldsymbol{y}, \boldsymbol{\theta}) \rangle_{k \neq i}}}{\int e^{\langle \ln p(\boldsymbol{y}, \boldsymbol{\theta}) \rangle_{k \neq i}} d\theta_i}, \tag{19}$$

where $\langle \cdot \rangle_{k \neq i}$ denotes the expectation with respect to the distributions $q_k(\theta_k)$ for all $k \neq i$. By taking the logarithm on both sides of (19), it can be equivalently written as

$$\ln q_i(\theta_i) = \langle \ln p(\boldsymbol{y}, \boldsymbol{\theta}) \rangle_{k \neq i} + \text{constant}. \tag{20}$$

B. Proposed Algorithm

We now proceed to perform variational Bayesian inference for the proposed hierarchical model. Let $\boldsymbol{\theta} \triangleq \{\boldsymbol{X}, \boldsymbol{\Sigma}, \gamma\}$ denote all hidden variables. Our objective is to find the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{y})$. Since $p(\boldsymbol{\theta}|\boldsymbol{y})$ is usually computationally intractable, we, following the idea of [38], approximate $p(\boldsymbol{\theta}|\boldsymbol{y})$ as $q(\boldsymbol{X}, \boldsymbol{\Sigma}, \gamma)$ which has a factorized form over the hidden variables $\{\boldsymbol{X}, \boldsymbol{\Sigma}, \gamma\}$, i.e.,

$$q(\boldsymbol{X}, \boldsymbol{\Sigma}, \gamma) = q_x(\boldsymbol{X}) q_\Sigma(\boldsymbol{\Sigma}) q_\gamma(\gamma). \tag{21}$$

As mentioned in the previous subsection, the maximization of $L(q)$ can be conducted in an alternating fashion for each latent variable, which leads to (details of the derivation can be found in [38])

$$\begin{aligned}
\ln q_x(\boldsymbol{X}) &= \langle \ln p(\boldsymbol{\Sigma}, \gamma) \rangle_{q_\Sigma(\boldsymbol{\Sigma}) q_\gamma(\gamma)} + \text{constant}, \\
\ln q_\Sigma(\boldsymbol{\Sigma}) &= \langle \ln p(\boldsymbol{X}, \gamma) \rangle_{q_x(\boldsymbol{X}) q_\gamma(\gamma)} + \text{constant}, \\
\ln q_\gamma(\gamma) &= \langle \ln p(\boldsymbol{X}, \boldsymbol{\Sigma},) \rangle_{q_x(\boldsymbol{X}) q_\Sigma(\boldsymbol{\Sigma})} + \text{constant},
\end{aligned}$$

where $\langle \rangle_{q_1(\cdot) \ldots q_K(\cdot)}$ denotes the expectation with respect to (w.r.t.) the distributions $\{q_k(\cdot)\}_{k=1}^K$. Details of this Bayesian inference scheme are provided next.

1) Update of $q_x(\boldsymbol{X})$: The calculation of $q_x(\boldsymbol{X})$ can be decomposed into a set of independent tasks, with each task computing the posterior distribution approximation for each column of $\boldsymbol{X}$, i.e., $q_x(\boldsymbol{x}_n)$. We have

$$\begin{aligned}
\ln q_x(\boldsymbol{x}_n) &= \langle \ln[p(\boldsymbol{y}_n|\boldsymbol{x}_n) p(\boldsymbol{x}_n|\boldsymbol{\Sigma})] \rangle_{q_\Sigma(\boldsymbol{\Sigma}) q_\gamma(\gamma)} + \text{const} \\
&= \langle -\gamma (\boldsymbol{y}_n - \boldsymbol{x}_n)^T \boldsymbol{O}_n (\boldsymbol{y}_n - \boldsymbol{x}_n) - \boldsymbol{x}_n^T \boldsymbol{\Sigma} \boldsymbol{x}_n \rangle + \text{const} \\
&= -\boldsymbol{x}_n^T (\langle \gamma \rangle \boldsymbol{O}_n + \langle \boldsymbol{\Sigma} \rangle) \boldsymbol{x}_n + 2 \langle \gamma \rangle \boldsymbol{x}_n^T \boldsymbol{O}_n \boldsymbol{y}_n + \text{const}
\end{aligned} \tag{22}$$

where $\boldsymbol{y}_n$ denotes the nth column of $\boldsymbol{Y}$ and $\boldsymbol{O}_n \triangleq \mathrm{diag}(\boldsymbol{o}_n)$, with $\boldsymbol{o}_n$ being the nth column of $\boldsymbol{\Omega}$. From (22), it can be seen that $\boldsymbol{x}_n$ follows a Gaussian distribution

$$q_x(\boldsymbol{x}_n) = \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_n, \boldsymbol{Q}_n) \tag{23}$$

with $\boldsymbol{\mu}_n$ and $\boldsymbol{Q}_n$ given as

$$\boldsymbol{\mu}_n = \langle \gamma \rangle \boldsymbol{Q}_n \boldsymbol{O}_n \boldsymbol{y}_n \tag{24}$$

$$\boldsymbol{Q}_n = (\langle \gamma \rangle \boldsymbol{O}_n + \langle \boldsymbol{\Sigma} \rangle)^{-1} \tag{25}$$

We see that to calculate $q_x(\boldsymbol{x}_n)$, we need to perform an inverse operation of an $M \times M$ matrix which involves a computational complexity of $\mathcal{O}(M^3)$.

2) Update of $q_\Sigma(\boldsymbol{\Sigma})$: The approximate posterior $q_\Sigma(\boldsymbol{\Sigma})$ can be obtained as

$$\begin{aligned}
&\ln q_\Sigma(\boldsymbol{\Sigma}) \\
&= \left\langle \ln\left[\prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{\Sigma})p(\boldsymbol{\Sigma})\right]\right\rangle_{q_x(\boldsymbol{X})} + \text{const} \\
&= \left\langle \frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}\text{tr}(\boldsymbol{X}^T\boldsymbol{\Sigma}\boldsymbol{X}) + \frac{\nu-M-1}{2}\ln|\boldsymbol{\Sigma}| \right. \\
&\quad \left. - \frac{1}{2}\text{tr}(\boldsymbol{W}^{-1}\boldsymbol{\Sigma})\right\rangle + \text{const} \\
&= \frac{\nu+N-M-1}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}\text{tr}((\boldsymbol{W}^{-1} + \langle\boldsymbol{X}\boldsymbol{X}^T\rangle)\boldsymbol{\Sigma}) + \text{const}
\end{aligned} \tag{26}$$

From (26), it can be seen that $\boldsymbol{\Sigma}$ follows a Wishart distribution, i.e.,

$$q_\Sigma(\boldsymbol{\Sigma}) = \text{Wishart}(\boldsymbol{\Sigma}; \hat{\boldsymbol{W}}, \hat{\nu}) \tag{27}$$

where

$$\hat{\boldsymbol{W}} = (\boldsymbol{W}^{-1} + \langle\boldsymbol{X}\boldsymbol{X}^T\rangle)^{-1} \tag{28}$$

$$\hat{\nu} = \nu + N \tag{29}$$

3) Update of $q_\gamma(\boldsymbol{\gamma})$: The variational optimization of $q_\gamma(\gamma)$ yields

$$\begin{aligned}
\ln q_\gamma(\gamma) &= \langle \ln p(\boldsymbol{Y}|\boldsymbol{X},\gamma)p(\gamma)\rangle_{q_x(\boldsymbol{X})} + \text{const} \\
&= \left\langle \ln \prod_{(m,n)\in\mathbb{S}} p(y_{mn}|x_{mn},\gamma)p(\gamma)\right\rangle + \text{const} \\
&= \left\langle \frac{L}{2}\ln\gamma - \frac{\gamma}{2}\sum_{(m,n)\in\mathbb{S}}(y_{mn}-x_{mn})^2 \right. \\
&\quad \left. + (c-1)\ln\gamma - d\gamma\right\rangle + \text{const} \\
&= \left(\frac{L}{2}+c-1\right)\ln\gamma \\
&\quad - \left(\frac{1}{2}\sum_{(m,n)\in\mathbb{S}}\langle(y_{mn}-x_{mn})^2\rangle + d\right)\gamma + \text{const}
\end{aligned} \tag{30}$$

where x_{mn} and y_{mn} denote the (m,n)th entry of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively, $\mathbb{S} \triangleq \{(m,n)|\Omega_{mn}=1\}$ is an index set consisting of indices of those observed entries, and $L \triangleq |\mathbb{S}|$ is the cardinality of the set $\mathbb{S}$, in which Ω_{mn} denotes the (m,n)th entry of $\boldsymbol{\Omega}$.

It is easy to verify that $q_\gamma(\gamma)$ follows a Gamma distribution

$$q_\gamma(\gamma) = \text{Gamma}(\gamma|\tilde{c},\tilde{d}) \tag{31}$$

with the parameters $\tilde{c}$ and $\tilde{d}$ given respectively by

$$\begin{aligned}
\tilde{c} &= \frac{L}{2} + c, \\
\tilde{d} &= \frac{1}{2}\sum_{(m,n)\in\mathbb{S}}\langle(y_{mn}-x_{mn})^2\rangle + d
\end{aligned} \tag{32}$$

Algorithm 1: VB Algorithm for Matrix Completion.

Input: $\boldsymbol{Y}$, $\boldsymbol{\Omega}$, ν and $\boldsymbol{W}$.
Output: $q_x(\boldsymbol{X})$, $q_\Sigma(\boldsymbol{\Sigma})$, $q_\gamma(\gamma)$.
Initialize $\langle\boldsymbol{\Sigma}\rangle$ and $\langle\gamma\rangle$;
while not converge **do**
  for $n = 1$ to N **do**
    Update $q_x(\boldsymbol{x}_n)$ via (23), with $q_\Sigma(\boldsymbol{\Sigma})$ and $q_\gamma(\gamma)$ fixed;
  end for
  Update $q_\Sigma(\boldsymbol{\Sigma})$ via (27), with $q_x(\boldsymbol{X})$ and $q_\gamma(\gamma)$ fixed;
  Update $q_\gamma(\gamma)$ via (31);
end while

where

$$\langle(y_{mn}-x_{mn})^2\rangle = y_{mn}^2 - 2y_{mn}\langle x_{mn}\rangle + \langle x_{mn}^2\rangle \tag{33}$$

Some of the expectations and moments used during the update are summarized as

$$\langle\boldsymbol{\Sigma}\rangle = \hat{\boldsymbol{W}}\hat{\nu} \tag{34}$$

$$\langle\boldsymbol{X}\boldsymbol{X}^T\rangle = \langle\boldsymbol{X}\rangle\langle\boldsymbol{X}\rangle^T + \sum_{n=1}^{N}\boldsymbol{Q}_n \tag{35}$$

$$\langle x_{mn}^2\rangle = \langle x_{mn}\rangle^2 + Q_n(m,m) \tag{36}$$

where $Q_n(m,m)$ denotes the mth diagonal entry of $\boldsymbol{Q}_n$.

For clarity, we summarize our algorithm in Algorithm 1.

It can be easily checked that the computational complexity of our proposed method is dominated by the update of the posterior distribution $q_x(\boldsymbol{X})$, which requires computing an $M \times M$ matrix inverse N times and therefore has a computational complexity scaling as $\mathcal{O}(M^3N)$. This makes the application of our proposed method to large data sets impractical. To address this issue, in the following, we develop a computationally efficient algorithm which obtains an approximation of $q_x(\boldsymbol{X})$ by resorting to the generalized approximate message passing (GAMP) technique [39].

IV. VB-GAMP

GAMP is a low-complexity Bayesian iterative technique recently developed in [39], [40] for obtaining approximate marginal posteriors. Note that the GAMP algorithm requires that both the prior distribution and the noise distribution have factorized forms [39]. Nevertheless, in our model, the prior distribution $p(\boldsymbol{x}_n|\boldsymbol{\Sigma})$ has a non-factorizable form, in which case the GAMP technique cannot be directly applied. To address this issue, we first construct a surrogate problem which aims to recover $\boldsymbol{x} \in \mathbb{R}^M$ from linear measurements $\boldsymbol{b} \in \mathbb{R}^M$:

$$\boldsymbol{b} = \boldsymbol{U}^T\boldsymbol{x} + \boldsymbol{e} \tag{37}$$

where $\boldsymbol{U} \in \mathbb{C}^{M\times M}$ is obtained by performing a singular value decomposition of $\langle\boldsymbol{\Sigma}\rangle = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T$, $\boldsymbol{U}$ is a unitary matrix and $\boldsymbol{S}$ is a diagonal matrix with its diagonal elements equal to the singular values of $\langle\boldsymbol{\Sigma}\rangle$, and $\boldsymbol{e}$ denotes the additive Gaussian noise with zero mean and covariance matrix $\boldsymbol{S}^{-1}$. We assume that

entries of $\boldsymbol{x}$ are mutually independent and follow the following distribution:

$$p(x_m) = \begin{cases} \mathcal{N}(\kappa_m, \xi^{-1}) & \text{if } \pi_m = 1 \\ C, & \text{if } \pi_m = 0 \end{cases} \tag{38}$$

where π_m, x_m, and κ_m denote the mth entry of $\boldsymbol{\pi}$, $\boldsymbol{x}$, and $\boldsymbol{\kappa}$, respectively, C is a constant, $\boldsymbol{\pi}$, $\boldsymbol{\kappa} \in \mathbb{R}^{M\times 1}$ and ξ are known parameters. It is noted that although $p(x_m) = C$ is an improper prior distribution, it can often be used provided the corresponding posterior distribution can be correctly normalized [30]. Also, when $\pi_m = 0$, κ_m can be any arbitrary value since we only use C to characterize the distribution of $p(x_m)$. Considering the surrogate problem (37), the posterior distribution of $\boldsymbol{x}$ can be calculated as

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{b}) &\propto p(\boldsymbol{b}|\boldsymbol{x})p(\boldsymbol{x}) \\ &\propto p(\boldsymbol{b}|\boldsymbol{x}) \prod_{m\in S} p(x_m) \\ &= \mathcal{N}(\boldsymbol{U}^T\boldsymbol{x}, \boldsymbol{S}^{-1}) \prod_{m\in S} \mathcal{N}(\kappa_m, \xi^{-1}) \end{aligned} \tag{39}$$

where $S \triangleq \{m|\pi_m = 1\}$.

Taking the logarithm of $p(\boldsymbol{x}|\boldsymbol{b})$, we have

$$\begin{aligned} \ln p(\boldsymbol{x}|\boldsymbol{b}) &\propto -\frac{1}{2}(\boldsymbol{b} - \boldsymbol{U}^T\boldsymbol{x})^T \boldsymbol{S}(\boldsymbol{b} - \boldsymbol{U}^T\boldsymbol{x}) \\ &\quad - \frac{1}{2}\xi \sum_{m\in S} (x_m - \kappa_m)^2 \\ &= -\frac{1}{2}(\boldsymbol{b} - \boldsymbol{U}^T\boldsymbol{x})^T \boldsymbol{S}(\boldsymbol{b} - \boldsymbol{U}^T\boldsymbol{x}) \\ &\quad - \frac{1}{2}\xi(\boldsymbol{x} - \boldsymbol{\kappa})^T \boldsymbol{\Pi}(\boldsymbol{x} - \boldsymbol{\kappa}) \\ &\propto -\frac{1}{2}\boldsymbol{x}^T(\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T + \xi\boldsymbol{\Pi})\boldsymbol{x}^T + (\boldsymbol{b}^T\boldsymbol{S}\boldsymbol{U}^T + \xi\boldsymbol{\kappa}^T\boldsymbol{\Pi})\boldsymbol{x} \end{aligned} \tag{40}$$

where $\boldsymbol{\Pi}$ is a diagonal matrix with its mth diagonal entry equal to π_m. Clearly, $p(\boldsymbol{x}|\boldsymbol{b})$ follows a Gaussian distribution with its mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{Q}$ given by

$$\boldsymbol{\mu} = \boldsymbol{Q}(\boldsymbol{U}\boldsymbol{S}\boldsymbol{b} + \xi\boldsymbol{\Pi}\boldsymbol{\kappa}) \tag{41}$$

$$\boldsymbol{Q} = (\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T + \xi\boldsymbol{\Pi})^{-1} = (\langle\boldsymbol{\Sigma}\rangle + \xi\boldsymbol{\Pi})^{-1} \tag{42}$$

Comparing (24)–(25) with (41)–(42), we can readily verify that when $\boldsymbol{b} = \boldsymbol{0}$, $\boldsymbol{\kappa} = \boldsymbol{y}_n$, $\boldsymbol{\pi} = \boldsymbol{o}_n$ (i.e., $\boldsymbol{\Pi} = \boldsymbol{O}_n$), and $\xi = \langle\gamma\rangle$, $p(\boldsymbol{x}|\boldsymbol{b})$ is exactly the desired posterior distribution $q_x(\boldsymbol{x}_n)$. Meanwhile, notice that for the surrogate problem (37), both the prior distribution and the noise distribution are factorizable. Hence the GAMP algorithm can be directly applied to (37) to find an approximation of the posterior distribution $p(\boldsymbol{x}|\boldsymbol{b})$. By setting $\boldsymbol{b} = \boldsymbol{0}$, $\boldsymbol{\kappa} = \boldsymbol{y}_n$, $\boldsymbol{\pi} = \boldsymbol{o}_n$, $\xi = \langle\gamma\rangle$, an approximate of $q_x(\boldsymbol{x}_n)$ in (23) can be efficiently obtained. We now proceed to derive the GAMP algorithm for the surrogate problem (37).

A. Solving (37) via GAMP

GAMP was developed in a message passing-based framework. It was shown in [39, 40] that the loopy belief propagation on the underlying factor graph can be greatly simplified and efficiently performed via using central-limit-theorem approximations. Following [39, 40], the GAMP algorithm tailored to our problem can be described as follows.

Firstly, GAMP approximates the true marginal posterior distribution $p(x_m|\boldsymbol{b})$ by

$$\hat{p}(x_m|\boldsymbol{b}, \hat{r}_m, \tau_m^r) = \frac{p(x_m)\mathcal{N}(x_m|\hat{r}_m, \tau_m^r)}{\int_x p(x_m)\mathcal{N}(x_m|\hat{r}_m, \tau_m^r)} \tag{43}$$

where $\hat{r}_m$ and τ_m^r are quantities iteratively updated during the iterative process of the GAMP algorithm. Here, we have dropped their explicit dependence on the iteration number k for simplicity. For the case $\pi_m = 1$, substituting the prior distribution (38) into (43), it can be easily verified that the approximate posterior $\hat{p}(x_m|\boldsymbol{b}, \hat{r}_m, \tau_m^r)$ follows a Gaussian distribution with its mean and variance given respectively as

$$\mu_m^x = \phi_m^x(\xi\kappa_m + \hat{r}_m/\tau_m^r) \tag{44}$$

$$\phi_m^x = \frac{\tau_m^r}{1 + \xi\tau_m^r} \tag{45}$$

Similarly, for the case $\pi_m = 0$, substituting the prior distribution (38) into (43), the approximate posterior $\hat{p}(x_m|\boldsymbol{b}, \hat{r}_m, \tau_m^r)$ follows a Gaussian distribution with its mean and variance given respectively as

$$\mu_m^x = \hat{r}_m \tag{46}$$

$$\phi_m^x = \tau_m^r \tag{47}$$

Another approximation is made to the noiseless output $z_i \triangleq \boldsymbol{u}_i^T\boldsymbol{x}$, where $\boldsymbol{u}_i^T$ denotes the ith row of $\boldsymbol{U}^T$. GAMP approximates the true marginal posterior $p(z_i|\boldsymbol{b})$ by

$$\hat{p}(z_i|\boldsymbol{b}, \hat{p}_i, \tau_i^p) = \frac{p(b_i|z_i)\mathcal{N}(z_i|\hat{p}_i, \tau_i^p)}{\int_z p(b_i|z_i)\mathcal{N}(z_i|\hat{p}_i, \tau_i^p)} \tag{48}$$

where $\hat{p}_i$ and τ_i^p are quantities iteratively updated during the iterative process of the GAMP algorithm. Again, here we dropped their explicit dependence on the iteration number k. Under the additive white Gaussian noise assumption, we have $p(b_i|z_i) = \mathcal{N}(b_i|z_i, s_i^{-1})$, where s_i denotes the ith diagonal element of $\boldsymbol{S}$. Thus $\hat{p}(z_i|\boldsymbol{b}, \hat{p}_i, \tau_i^p)$ also follows a Gaussian distribution with its mean and variance given by

$$\mu_i^z = \frac{\tau_i^p s_i b_i + \hat{p}_i}{1 + s_i\tau_i^p} \tag{49}$$

$$\phi_i^z = \frac{\tau_i^p}{1 + s_i\tau_i^p} \tag{50}$$

With the above approximations, we can now define the following two scalar functions: $g_{\text{in}}(\cdot)$ and $g_{\text{out}}(\cdot)$ that are used in the GAMP algorithm. The input scalar function $g_{\text{in}}(\cdot)$ is simply defined as the posterior mean μ_m^x, i.e.,

$$g_{\text{in}}(\hat{r}_m, \tau_m^r) = \mu_m^x = \begin{cases} \phi_m^x(\xi\kappa_m + \hat{r}_m/\tau_m^r) & \text{if } \pi_m = 1 \\ \hat{r}_m & \text{if } \pi_m = 0 \end{cases} \tag{51}$$

Algorithm 2: GAMP Algorithm.

Input: $\boldsymbol{\kappa}$, $\boldsymbol{\pi}$, $\boldsymbol{b}$, and ξ.

Output: $\{\hat{r}_m, \tau_m^r\}$, $\{\hat{p}_i, \tau_i^p\}$, and $\{\mu_m^x, \phi_m^x\}$.

Initialization: Set $\hat{\psi}_i = 0, \forall i \in \{1, \ldots, M\}$; $\{\mu_m^x\}_{m=1}^M$ are initialized as the mean variance of the prior distribution, and $\{\phi_m^x\}_{m=1}^M$ are set to small values, say 10^{-5}.

while not converge **do**

Step 1. $\forall i \in \{1, \ldots, M\}$:

$$\hat{z}_i = \sum_m u_{i,m} \mu_m^x$$

$$\tau_i^p = \sum_m u_{i,m}^2 \phi_m^x$$

$$\hat{p}_i = \hat{z}_i - \tau_i^p \hat{\psi}_i$$

Step 2. $\forall i \in \{1, \ldots, M\}$:

$$\hat{\psi}_i = g_{\text{out}}(\hat{p}_i, \tau_i^p)$$

$$\tau_i^s = -\frac{\partial}{\partial \hat{p}_i} g_{\text{out}}(\hat{p}_i, \tau_i^p)$$

Step 3. $\forall m \in \{1, \ldots, M\}$:

$$\tau_m^r = \left(\sum_i u_{i,m}^2 \tau_i^s \right)^{-1}$$

$$\hat{r}_m = \mu_m^x + \tau_m^r \sum_i u_{i,m} \hat{\psi}_i$$

Step 4. $\forall m \in \{1, \ldots, M\}$:

$$\mu_m^x = g_{\text{in}}(\hat{r}_m, \tau_m^r)$$

$$\phi_m^x = \tau_m^r \frac{\partial}{\partial \hat{r}_m} g_{\text{in}}(\hat{r}_m, \tau_m^r)$$

end while

The scaled partial derivative of $\tau_m^r g_{\text{in}}(\hat{r}_m, \tau_m^r)$ with respect to $\hat{r}_m$ is the posterior variance ϕ_m^x, i.e.,

$$\tau_m^r \frac{\partial}{\partial \hat{r}_m} g_{\text{in}}(\hat{r}_m, \tau_m^r) = \phi_m^x = \begin{cases} \frac{\tau_m^r}{1+\xi\tau_m^r} & \text{if } \pi_m = 1 \\ \tau_m^r & \text{if } \pi_m = 0 \end{cases} \quad (52)$$

The output scalar function $g_{\text{out}}(\cdot)$ is related to the posterior mean μ_i^z as follows

$$g_{\text{out}}(\hat{p}_i, \tau_{i,n}^p) = \frac{1}{\tau_i^p}(\mu_i^z - \hat{p}_i) = \frac{s_i(b_i - \hat{p}_i)}{1 + s_i \tau_i^p} \quad (53)$$

The partial derivative of $g_{\text{out}}(\hat{p}_i, \tau_i^p)$ is related to the posterior variance $\phi_{i,n}^z$ in the following way

$$\frac{\partial}{\partial \hat{p}_i} g_{\text{out}}(\hat{p}_i, \tau_i^p) = \frac{\phi_i^z - \tau_i^p}{(\tau_i^p)^2} = \frac{-s_i}{(1 + s_i \tau_i^p)} \quad (54)$$

Given the above definitions of $g_{\text{in}}(\cdot)$ and $g_{\text{out}}(\cdot)$, the GAMP algorithm tailored to the considered problem (37) can now be summarized as follows (details of the derivation of the GAMP algorithm can be found in [39]), in which $u_{i,m}$ denotes the (i,m)th entry of $\boldsymbol{U}^T$.

Algorithm 3: VB-GAMP Algorithm for Matrix Completion.

Input: $\boldsymbol{Y}$, $\boldsymbol{\Omega}$, ν and $\boldsymbol{W}$.

Output: $q_x(\boldsymbol{X})$, $q_\Sigma(\boldsymbol{\Sigma})$, and $q_\gamma(\gamma)$.

1: Initialize $\langle \boldsymbol{X} \rangle$, $\langle \boldsymbol{\Sigma} \rangle$;
2: **while** not converge **do**
3: Calculate singular value decomposition of $\langle \boldsymbol{\Sigma} \rangle$;
4: **for** $n = 1$ to N **do**
5: Obtain an approximation of $q_x(\boldsymbol{x}_n)$ via Algorithm 2;
6: **end for**
7: Update $q_\Sigma(\boldsymbol{\Sigma})$ via (27);
8: Update $q_\gamma(\gamma)$ via (31);
9: **end while**

B. Discussions

We have now derived an efficient algorithm to obtain an approximate posterior distribution of $\boldsymbol{x}$ for (37). Specifically, the true marginal posterior distribution of x_m is approximated by a Gaussian distribution $\hat{p}(x_m|\boldsymbol{b}, \hat{r}_m, \tau_m^r)$ with its mean and variance given by (44)–(45) or (46)–(47), depending on the value of π_m. The joint posterior distribution $p(\boldsymbol{x}|\boldsymbol{b})$ can be approximated as a product of approximate marginal posterior distributions:

$$p(\boldsymbol{x}|\boldsymbol{b}) \approx \hat{p}(\boldsymbol{x}|\boldsymbol{b}) = \prod_{m=1}^M \hat{p}(x_m|\boldsymbol{b}, \hat{r}_m, \tau_m^r) \quad (55)$$

As indicated earlier, by setting $\boldsymbol{b} = \boldsymbol{0}, \boldsymbol{\kappa} = \boldsymbol{y}_n, \boldsymbol{\pi} = \boldsymbol{o}_n$, and $\xi = \langle \gamma \rangle$, the posterior distribution $\hat{p}(\boldsymbol{x}|\boldsymbol{b})$ obtained via the GAMP algorithm can be used to approximate $q_x(\boldsymbol{x}_n)$ in (23).

We see that to approximate $q_x(\boldsymbol{X})$ by using the GAMP, we first need to perform a singular value decomposition (SVD) of $\langle \boldsymbol{\Sigma} \rangle$, which has a computational complexity of $\mathcal{O}(M^3)$. The GAMP algorithm used to approximate $q_x(\boldsymbol{x}_n)$ involves very simple matrix-vector multiplications which has a computational complexity scaling as $\mathcal{O}(M^2)$. Therefore the overall computational complexity for updating $q_x(\boldsymbol{X})$ is of order $\mathcal{O}(M^2 N)$. In contrast, using (24)–(25) to update $q_x(\boldsymbol{X})$ requires a computational complexity of $\mathcal{O}(NM^3)$. Thus the GAMP technique can help achieve a significant reduction in the computational complexity as compared with a direct calculation of $q_x(\boldsymbol{X})$. For some practical problems involving completion of matrices with very large dimensions, a computational complexity of $O(M^2)$ may be still high. Also, in such cases, the memory efficiency is an important issue that should be considered. Since the computational complexity scales linearly with N, the proposed method may be more suitable for cases where M is relatively small while N can be large.

For clarity, the VB-GAMP algorithm for matrix completion is summarized as Algorithm 3.

The proposed method proceeds in a double-loop manner, the outer loop calculate the variational posterior distributions $q_\gamma(\gamma)$ and $q_\Sigma(\boldsymbol{\Sigma})$, and the inner loop computes an approximation of $q_x(\boldsymbol{X})$. It is noted that there is no need to wait until the GAMP converges. Experimental results show that GAMP provides a

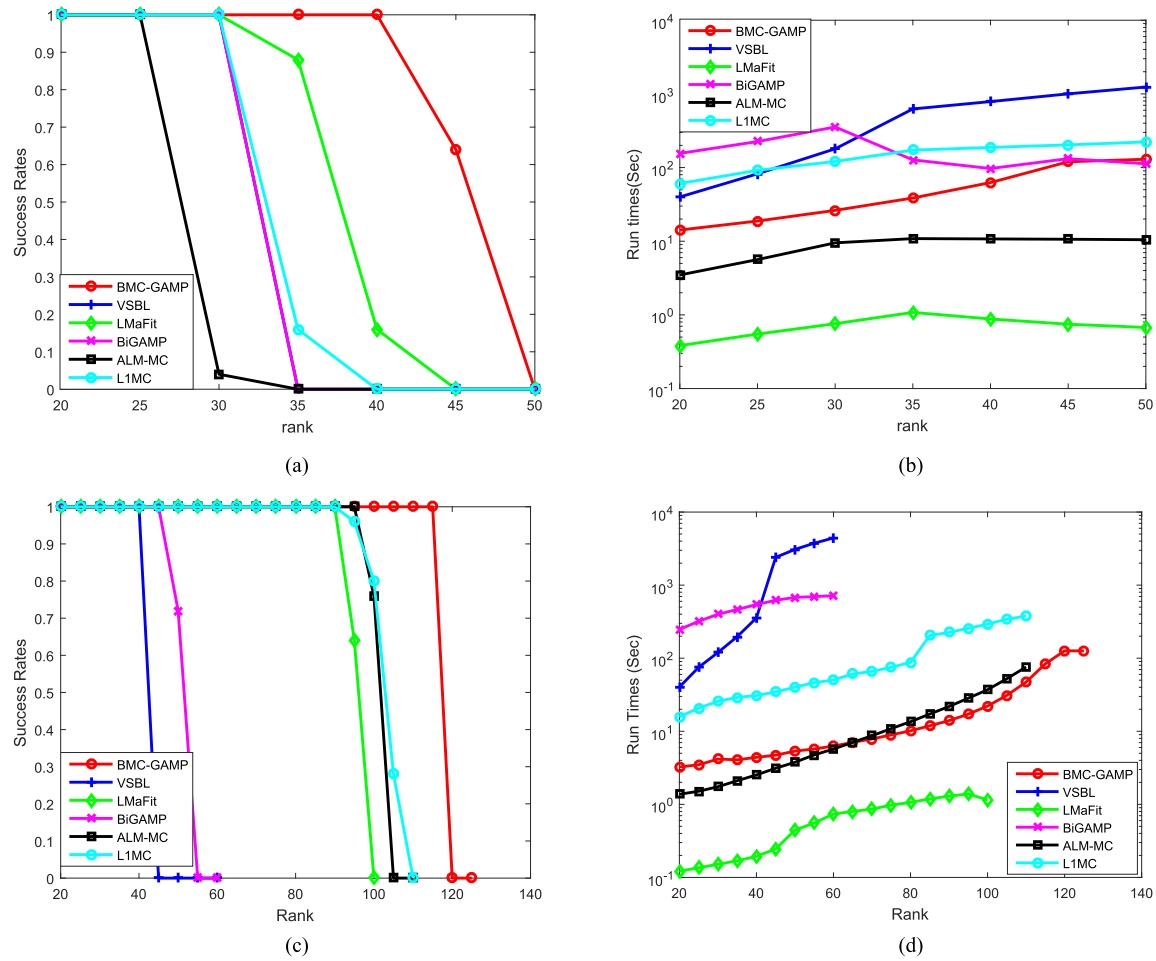

Fig. 2. Synthetic data: (a) Success rates vs. the rank of the matrix, $\rho = 0.2$; (b) Run times vs. the rank of the matrix, $\rho = 0.2$; (c) Success rates vs. the rank of the matrix $\rho = 0.5$; (d) Run times vs. the rank of the matrix, $\rho = 0.5$.

reliable approximation of $q_x(\boldsymbol{x}_n)$ even if only a few iterations are performed. In our experiments, only one iteration is used to implement GAMP.

V. Experiments

In this section, we carry out experiments to illustrate the performance of our proposed GAMP-assisted Bayesian matrix completion method with hierarchical Gaussian priors (referred to as BMC-GP-GAMP). Note that our method involves selecting a, b, ν, and $\boldsymbol{W}$. To provide non-informative hyperpriors, we usually set a, b, to be small values, say, $a = b = 10^{-10}$, and $\boldsymbol{W} = 10^{10}\boldsymbol{I}$. Also, we choose a small value of ν ($\nu = 1$ in our experiments) in order to encourage a low-rank precision matrix. If we want to capture the smoothness between neighboring coefficients, the matrix parameter $\boldsymbol{W}$ can be chosen according to (10) or (12), where (10) does not need to specify any parameter, and (12) only involves a parameter θ whose choice can be easily determined by trying different values of θ on a validation set and selecting the best one.

We compare our method with several state-of-the-art methods, namely, the variational sparse Bayesian learning method (also referred to as VSBL) [20] which models the low rank of the matrix as the structural sparsity of its two factor matrices, the bilinear GAMP-based matrix completion method (also referred to as BiGAMP-MC) [23] which implements the VSBL using bilinear GAMP, the inexact version of the augmented Lagrange multiplier based matrix completion method (also referred to ALM-MC) [15], a low-rank matrix fitting method (also referred to as LMaFit) [41] which iteratively minimizes the fitting error and estimates the rank of the matrix, and an L1-norm regularized rank-one matrix completion method with automatic rank estimation (also referred to L1MC) [28]. It should be noted that VSBL, LMaFit, and L1MC require to set an overestimated rank. Codes of our proposed algorithm along with other competing algorithms are available at http://www.junfanguestc.net/codes/LRMC.rar, in which codes of other competing algorithms are obtained from their respective websites.

A. Synthetic Data

We first examine the effectiveness of our proposed method on synthetic data. We generate the test rank-k matrix $\boldsymbol{X}$ of size 500×500 by multiplying $\boldsymbol{A} \in \mathbb{R}^{500\times k}$ by the transpose of $\boldsymbol{B} \in \mathbb{R}^{500\times k}$, i.e., $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{B}^T$. All the entries of $\boldsymbol{A}$ and $\boldsymbol{B}$ are sampled from a normal distribution. We consider the scenarios where

TABLE I
SUCCESS RATES/RUN TIMES VS. RANK

	3	5	7	9
BMC-GP-GAMP	100%/0.04	100%/0.07	88%/0.14	0/0.29
BARM	100%/1.20	96%/7.00	0/2.95	0/4.14

20% ($\rho = 0.2$) and 50% ($\rho = 0.5$) entries of $\boldsymbol{X}$ are observed. Here ρ denotes the sampling ratio. The success rates as well as the run times of respective algorithms as a function of the rank of $\boldsymbol{X}$, i.e., k, are plotted in Fig. 2 Results are averaged over 25 independent trials. A trial is considered to be successful if the relative error is smaller than 10^{-2}, i.e., $||\boldsymbol{X} - \hat{\boldsymbol{X}}||_F / ||\boldsymbol{X}||_F < 10^{-2}$, where $\hat{\boldsymbol{X}}$ denotes the estimated matrix. For our proposed method, the matrix parameter $\boldsymbol{W}$ is set to $10^{10}\boldsymbol{I}$. The pre-defined overestimated rank for VSBL, LMaFit and L1MC is set to be twice the true rank. For the case $\rho = 0.2$, VSBL and BiGAMP-MC present the same recovery performance with their curves overlapping each other. From Fig. 2, we can see that

1) Our proposed method presents the best performance for both sampling ratio cases. Meanwhile, it has a moderate computational complexity. When the sampling ratio is set to 0.5, our proposed method has a run time similar to the ALM-MC method, while provides a clear performance improvement over the ALM-MC.
2) The LMaFit method is the most computationally efficient. But its performance is not as good as our proposed method.
3) The proposed method outperforms the other two Bayesian methods, namely, the VSBL and the BiGAMP-MC, by a big margin in terms of both recovery accuracy and computational complexity. Since the BiGAMP cannot automatically determine the matrix rank, it needs to try all possible values of the rank, which makes running the BiGAMP-MC time costly.

Since the algorithm proposed in [24] (referred as to BRAM) has a prohibitive computational complexity when the matrix dimension is large, we only report its results on a small-size data set. Specifically, we randomly generate a 50×50 matrix with rank-k, and randomly select 40% entries as the observed data. The success rates and average run times of our proposed method and the BRAM are given in Table I. From Table I, we see that our proposed method achieves a higher success rate than the BARM. Also, it takes much less time for our proposed algorithm to perform the matrix completion task.

B. Gene Data

We carry out experiments on gene data for genotype estimation. The dataset [4] is a matrix of size 790×112 provided by Wellcome Trust Case Control Consortium (WTCCC) and contains the genetic information from chromosome 22. The dataset, which is referred to as "Chr22", has been shown in [4] to be approximately low-rank. We randomly select 20% or 50% of the entries of the dataset as observations, and recover the rest entries using low-rank matrix completion methods. Again, for our proposed method, the matrix parameter $\boldsymbol{W}$ is set to $10^{10}\boldsymbol{I}$. The pre-defined ranks used for VSBL, LMaFit and L1MC are all set to 100. Following [4], we use a metric termed as "allelic-imputation error rate" to evaluate the performance of respective methods. The error rate is defined as

$$\text{Error Rate} = \frac{\text{nnz}(|\boldsymbol{X} - \text{round}(\hat{\boldsymbol{X}})|)}{T} \tag{56}$$

where $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$ denotes the true and the estimated matrices, respectively, the operation round($\boldsymbol{X}$) returns a matrix with each entry of $\boldsymbol{X}$ rounded to its nearest integer, nnz($\boldsymbol{X}$) counts the number of non-zero entries of $\boldsymbol{X}$, and T denotes the number of unobserved entries. We report the average error rates as well as average run times of respective algorithms in Table II. From Table II, we see that all methods, except the LMaFit method, present similar results and the proposed method slightly outperforms other methods when 50% entries are observed. Despite the superior performance on synthetic data, the LMaFit method incurs large estimation errors for this dataset.

TABLE II
ERROR RATE/RUN TIME(S) FOR CHR22 DATASET

	20%	50%
BMC-GP-GAMP	0.0560/2.80	**0.0226**/3.28
VSBL	0.0568/261.19	0.0324/1026
LMaFit	0.2511/0.05	0.2479/0.11
BiGAMP-MC	0.0637/11.89	0.0346/33.81
ALM-MC	**0.0506**/0.57	0.0248/0.58
L1MC	0.0748/27.62	0.0403/2.22

TABLE III
NMAE/RUN TIME(S) FOR 100K MOVIELENS DATASET

	20%	50%
BMC-GP-GAMP	**0.1932**/43.61	0.1861/43.84
VSBL	0.2004/433.1	**0.1851**/694.8
LMaFit	0.2677/3.48	0.2315/3.59
BiGAMP-MC	0.2007/14.78	0.1863/158.9
ALM-MC	0.2005/31.15	0.1894/38.49
L1MC	0.2015/586.5	0.1898/597.9

C. Collaborative Filtering

In this experiment, we study the performance of respective methods on the task of collaborative filtering. We use the MovieLens 100 k dataset,[1] which consists of 10^5 ratings ranging from 1 to 5 on 1682 movies from 943 users. The ratings can form a matrix of size 943×1682. We randomly choose 20% or 50% of available ratings as training data, and predict the rest ratings using respective matrix completion methods. The matrix parameter $\boldsymbol{W}$ used in the proposed method is set to $10^{10}\boldsymbol{I}$. The pre-defined ranks used for VSBL, LMaFit and L1MC are all set to 100. The performance is evaluated by the normalized mean absolute error (NMAE), which is calculated as

$$\text{NMAE} = \frac{\sum_{(i,j)\in S} |x_{ij} - \hat{x}_{ij}|}{(r_{\max} - r_{\min})|S|} \tag{57}$$

where S is a set containing the indexes of those unobserved available ratings, $r_{\max}$ and $r_{\min}$ denote the maximum and

[1] Available at http://www.grouplens.org/node/73/

TABLE IV
IMAGE INPAINTING (PSNR/SSIM/RUN TIMES (S))

	Monarch		Lena	
	30%	50%	20%	30%
BMC-GP-GAMP-I	19.4607/0.4991/0.66	23.9122/0.7114/0.62	23.3995/0.4923/4.21	25.6079/0.5977/4.14
BMC-GP-GAMP-II	20.9671/**0.6986**/0.76	**25.8883/0.8733**/0.79	**25.3964/0.7517**/4.59	**27.9966/0.8340**/4.51
BMC-GP-GAMP-III	**21.2819**/0.6752/0.73	25.8204/0.8445/0.75	25.3598/0.7423/4.61	27.9346/0.8230/4.55
VSBL	15.8836/0.3420/170.1	19.9975/0.5459/236.5	21.2220/0.5001/331.4	23.5909/0.5905/699.8
LMaFit	17.7710/0.4063/0.08	19.2804/0.5131/0.05	21.6259/0.4611/0.22	22.3392/0.5462/0.19
BiGAMP-MC	18.8113/0.4521/18.78	22.3116/0.6323/26.67	22.6746/0.4916/68.4	24.8125/0.5944/200.8
ALM-MC	19.6285/0.5175/2.62	23.9110/0.7340/2.65	23.1001/0.5177/10.69	25.5364/0.6451/10.46
L1MC	16.0975/0.3916/34.26	22.72/0.6701/37.67	22.2453/0.4745/149.62	24.7623/0.6069/162.2

Fig. 3. Top row (from left to right): observed Butterfly image with missing pixels ($\rho = 0.3$), images recovered by BMC-GP-GAMP-I, BMC-GP-GAMP-II, BMC-GP-GAMP-III, VSBL, respectively. Bottom row (from left to right): images recovered by LMaFit, BiGAMP-MC, ALM-MC, and L1MC, respectively.

minimum ratings, respectively. The results of NMAE and run times are shown in Table III, from which we see that the proposed method achieves the most accurate rating prediction when the number of observed ratings is small.

D. Image Inpainting

Lastly, we evaluate the performance of different methods on image inpainting. The objective of image inpainting is to complete an image with missing pixels. We conduct experiments on the benchmark images Butterfly and Lena, which are of size 256×256 and 512×512, respectively. In our experiments, we examine the performance of our proposed method under different choices of $\boldsymbol{W}$. As usual, we can set $\boldsymbol{W} = 10^{10}\boldsymbol{I}$. Such a choice of $\boldsymbol{W}$ is referred to as BMC-GP-GAMP-I. We can also set $\boldsymbol{W}$ according to (10) and (12), which are respectively referred to as BMC-GP-GAMP-II and BMC-GP-GAMP-III. The parameters $\hat{\epsilon}$ and θ in (12) are set to 10^{-6} and $\sqrt{3}$, respectively. As discussed earlier in our paper, the latter two choices exploit both the low-rank structure and the smoothness of the signal. For the Butterfly image, we consider cases where 30% and 50% of pixels in the image are observed. For the Lena image, we consider cases where 20% and 30% of pixels are observed. We report the peak signal to noise ratio (PSNR) as well as the structural similarity (SSIM) index of each algorithm in Table IV. The original image with missing pixels and these images reconstructed by respective algorithms are shown in Fig. 3, 4, 5, and 6. From Table IV, we see that with a common choice of $\boldsymbol{W} = 10^{10}\boldsymbol{I}$, our proposed method, BMC-GP-GAMP-I, outperforms other methods in most cases. When $\boldsymbol{W}$ is more carefully devised, our proposed method, i.e., BMC-GP-GAMP-II and BMC-GP-GAMP-III, surpasses other methods by a substantial margin in terms of both PSNR and SSIM metrics. This result indicates that a careful choice of $\boldsymbol{W}$ that captures both the low-rank structure as well as the smoothness of the latent matrix can help substantially improve the recovery performance. From the reconstructed images, we also see that our proposed method, especially BMC-GP-GAMP-II and BMC-GP-GAMP-III, provides the best visual quality among all these methods. We observed that the images recovered by our proposed method are not very smooth along the row dimension, particularly when the matrix $\boldsymbol{W}$ is chosen to encourage the columnwise smoothness of the solution. The loss of smoothness along the row dimension is possibly because columns of $\boldsymbol{X}$ are assumed to be mutually independent in our prior model. On the other hand, due to its potential to promote a low-rank solution, our prior model is supposed to enhance the similarity of columns and improve the row-wise smoothness. Although imposing column-wise smoothness results in a certain amount of loss of smoothness along the row direction, it generally helps obtain a better visual quality.

Fig. 4. Top row (from left to right): observed Butterfly image with missing pixels ($\rho = 0.5$), images recovered by BMC-GP-GAMP-I, BMC-GP-GAMP-II, BMC-GP-GAMP-III, VSBL, respectively. Bottom row (from left to right): images recovered by LMaFit, BiGAMP-MC, ALM-MC, and L1MC, respectively.

Fig. 5. Top row (from left to right): observed Lena image with missing pixels ($\rho = 0.2$), images recovered by BMC-GP-GAMP-I, BMC-GP-GAMP-II, BMC-GP-GAMP-III, VSBL, respectively. Bottom row (from left to right): images recovered by LMaFit, BiGAMP-MC, ALM-MC, and L1MC, respectively.

Fig. 6. Top row (from left to right): observed Lena image with missing pixels ($\rho = 0.3$), images recovered by BMC-GP-GAMP-I, BMC-GP-GAMP-II, BMC-GP-GAMP-III, VSBL, respectively. Bottom row (from left to right): images recovered by LMaFit, BiGAMP-MC, ALM-MC, and L1MC, respectively.

VI. Conclusion

The problem of low-rank matrix completion was studied in this paper. A hierarchical Gaussian prior model was proposed to promote the low-rank structure of the underlying matrix, in which columns of the low-rank matrix are assumed to be mutually independent and follow a common Gaussian distribution with zero mean and a precision matrix. The precision matrix is treated as a random parameter, with a Wishart distribution specified as a hyperprior over it. Based on this hierarchical prior model, we developed a variational Bayesian method for matrix completion. To avoid cumbersome matrix inverse operations, the GAMP technique was used and embedded in the variational Bayesian inference, which resulted in an efficient VB-GAMP algorithm. Empirical results on synthetic and real datasets show that our proposed method offers competitive performance for matrix completion, and meanwhile achieves a significant reduction in computational complexity.

Appendix A
Detailed Derivation of (7)

We provide a detailed derivation of (7). We have

$$\begin{aligned} p(\boldsymbol{X}) &= \int \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\Sigma}) p(\boldsymbol{\Sigma}) d\boldsymbol{\Sigma} \\ &\propto \int \left(\frac{|\boldsymbol{\Sigma}|}{(2\pi)^M}\right)^{\frac{N}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}(\boldsymbol{X}^T\boldsymbol{\Sigma}\boldsymbol{X})\right) \\ &\quad \times |\boldsymbol{\Sigma}|^{\frac{\nu-M-1}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}(\boldsymbol{W}^{-1}\boldsymbol{\Sigma})\right) d\boldsymbol{\Sigma} \\ &\propto 2^{\frac{\nu M}{2}} \pi^{-\frac{MN}{2}} \Gamma_M\left(\frac{\nu+N}{2}\right) |\boldsymbol{W}^{-1}+\boldsymbol{X}\boldsymbol{X}^T|^{-\frac{\nu+N}{2}} \\ &\quad \times \int \frac{|\boldsymbol{\Sigma}|^{\frac{\nu+N-M-1}{2}} \exp(-\frac{1}{2}\mathrm{Tr}((\boldsymbol{W}^{-1}+\boldsymbol{X}\boldsymbol{X}^T)\boldsymbol{\Sigma}))}{2^{\frac{(\nu+N)M}{2}} |(\boldsymbol{W}^{-1}+\boldsymbol{X}\boldsymbol{X}^T)^{-1}|^{\frac{\nu+N}{2}} \Gamma_M(\frac{\nu+N}{2})} d\boldsymbol{\Sigma} \end{aligned} \tag{58}$$

where

$$\Gamma_M(x) = \pi^{\frac{M(M-1)}{4}} \prod_{j=1}^{M} \Gamma\left(x + \frac{1-j}{2}\right) \tag{59}$$

Note that the term in the integral of (58) is a standard Wishart distribution with $\nu+N$ degrees of freedom and variance matrix $(\boldsymbol{W}+\boldsymbol{X}\boldsymbol{X}^T)^{-1}$. Thus we arrive at

$$\begin{aligned} p(\boldsymbol{X}) &\propto 2^{\frac{\nu M}{2}} \pi^{-\frac{MN}{2}} \Gamma_M\left(\frac{\nu+N}{2}\right) |\boldsymbol{W}^{-1}+\boldsymbol{X}\boldsymbol{X}^T|^{-\frac{\nu+N}{2}} \\ &\propto |\boldsymbol{W}^{-1}+\boldsymbol{X}\boldsymbol{X}^T|^{-\frac{\nu+N}{2}} \end{aligned} \tag{60}$$

Appendix B
Proof of Lemma 1

Since we have $|\boldsymbol{X}\boldsymbol{X}^T+\boldsymbol{W}^{-1}| = |\boldsymbol{W}^{-1}||\boldsymbol{W}\boldsymbol{X}\boldsymbol{X}^T+\boldsymbol{I}|$, we only need to prove

$$|\boldsymbol{W}\boldsymbol{X}\boldsymbol{X}^T+\boldsymbol{I}| = |\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}+\boldsymbol{I}| \tag{61}$$

Recalling the determinant of block matrices, we have

$$|\boldsymbol{W}\boldsymbol{X}\boldsymbol{X}^T+\boldsymbol{I}| = \begin{vmatrix} \boldsymbol{I} & \boldsymbol{X}^T \\ \boldsymbol{0} & \boldsymbol{W}\boldsymbol{X}\boldsymbol{X}^T+\boldsymbol{I} \end{vmatrix} \tag{62}$$

and

$$|\boldsymbol{I}| = \begin{vmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{W}\boldsymbol{X} & \boldsymbol{I} \end{vmatrix} = \begin{vmatrix} \boldsymbol{I} & -\boldsymbol{X}^T \\ \boldsymbol{0} & \boldsymbol{I} \end{vmatrix} \tag{63}$$

which yields

$$\begin{aligned} &|\boldsymbol{W}\boldsymbol{X}\boldsymbol{X}^T+\boldsymbol{I}| \\ &= \begin{vmatrix} \boldsymbol{I} & \boldsymbol{X}^T \\ \boldsymbol{0} & \boldsymbol{W}\boldsymbol{X}\boldsymbol{X}^T+\boldsymbol{I} \end{vmatrix} \\ &= \left| \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{X}^T \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{W}\boldsymbol{X} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{X}^T \\ \boldsymbol{0} & \boldsymbol{W}\boldsymbol{X}\boldsymbol{X}^T+\boldsymbol{I} \end{bmatrix} \right| \\ &= \begin{vmatrix} \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}+\boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{W}\boldsymbol{X} & \boldsymbol{I} \end{vmatrix} \\ &= |\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}+\boldsymbol{I}| \end{aligned} \tag{64}$$

Thus we have

$$\log|\boldsymbol{X}\boldsymbol{X}^T+\boldsymbol{W}^{-1}| = \log|\boldsymbol{W}^{-1}| + \log|\boldsymbol{I}+\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}| \tag{65}$$

This completes the proof.

References

[1] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering," in *Proc. 4th ACM Conf. Recommender Syst.*, Barcelona, Spain, Sep. 2010, pp. 79–86.

[2] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with Bayesian probabilistic tensor factorization," in *Proc. SIAM Int. Conf. Data Mining*, 2010, vol. 10, pp. 211–222.

[3] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender Systems Handbook*. Berlin, Germany: Springer-Verlag, 2011, pp. 217–253.

[4] B. Jiang *et al.*, "Sparrec: An effective matrix completion framework of missing data imputation for GWAS," *Sci. Rep.*, vol. 6, 2016.

[5] E. C. Chi, H. Zhou, G. K. Chen, D. O. D. Vecchyo, and K. Lange, "Genotype imputation via matrix completion," *Genome Res.*, vol. 23, no. 3, pp. 509–518, 2013.

[6] R. Cabral, F. D. la Torre, J. P. Costeira, and A. Bernardino, "Matrix completion for weakly-supervised multi-label image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 121–135, Jan. 2015.

[7] Y. Luo, T. Liu, D. Tao, and C. Xu, "Multiview matrix completion for multilabel image classification," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2355–2368, Aug. 2015.

[8] W. Ye, L. Chen, G. Yang, H. Dai, and F. Xiao, "Anomaly-tolerant traffic matrix estimation via prior information guided matrix completion," *IEEE Access*, vol. 5, pp. 3172–3182, 2017.

[9] K. He and J. Sun, "Image completion approaches using the statistics of similar patches," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2423–2435, Dec. 2014.

[10] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.

[11] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.

[12] M. Fazel, E. J. Candès, B. Recht, and P. A. Parrilo, "Compressed sensing and robust recovery of low rank matrices," in *Proc. IEEE 2008 42nd Asilomar Conf. Signals, Syst. Comput*, 2008, pp. 1043–1047.

[13] E. J. Candès and Yaniv, "Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2342–2359, Apr. 2011.

[14] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.
[15] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," arXiv:1009.5055, 2010.
[16] K. Mohan and M. Fazel, "Iterative reweighted least squares for matrix rank minimization," in *Proc. IEEE 48th Annu. Allerton Conf. Annu., Control, Comput.*, 2010, pp. 653–661.
[17] D. Wipf, "Non-convex rank minimization via an empirical bayesian approach," arXiv:1408.2054, 2014.
[18] Z. Kang, C. Kang, J. Cheng, and Q. Cheng, "Logdet rank minimization with application to subspace clustering," *Comput. Intell. Neurosci.*, vol. 2015, pp. 68–77, 2015.
[19] Z. Yang and L. Xie, "Enhancing sparsity and resolution via reweighted atomic norm minimization," *IEEE Trans. Signal Process.*, vol. 64, no. 4, pp. 995–1006, Feb. 2016.
[20] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Sparse Bayesian methods for low-rank matrix estimation," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 3964–3977, Aug. 2012.
[21] M. Zhou, C. Wang, M. Chen, J. Paisley, D. Dunson, and L. Carin, "Nonparametric Bayesian matrix completion," in *Proc. IEEE Sensor Array Multichannel Signal Process. Workshop*, 2010, pp. 213–216.
[22] J. T. Parker, P. Schniter, and V. Cevher, "Bilinear generalized approximate message passing—Part I: Derivation," *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 5839–5853, Nov. 2014.
[23] J. T. Parker, P. Schniter, and V. Cevher, "Bilinear generalized approximate message passing—Part II: Applications," *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 5854–5867, Nov. 2014.
[24] B. Xin, Y. Wang, W. Gao, and D. Wipf, "Exploring algorithmic limits of matrix rank minimization under affine constraints," *IEEE Trans. Signal Process.*, vol. 64, no. 19, pp. 4960–4974, Oct. 2016.
[25] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 912–926, Sep. 2011.
[26] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Proc. 44th Annu. ACM Symp. Theory Comput.*, San Jose, CA, USA, Jun. 2013, pp. 665–674.
[27] R. Sun and Z.-Q. Luo, "Guaranteed matrix completion via non-convex factorization," *IEEE Trans. Inf. Theory*, vol. 62, no. 11, pp. 6535–6579, Nov. 2016.
[28] Q. Shi, H. Lu, and Y.-M. Cheung, "Rank-one matrix completion with automatic rank estimation via L1-norm regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, To be published.
[29] P. J. Everson and C. N. Morris, "Inference for multivariate normal hierarchical models," *J. Roy. Statist. Soc., B (Statist. Methodology)*, vol. 62, pp. 399–412, Mar. 2000.
[30] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer-Verlag, 2007.
[31] E. Candés, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, pp. 877–905, Dec. 2008.
[32] Y. Shen, J. Fang, and H. Li, "Exact reconstruction analysis of log-sum minimization for compressed sensing," *IEEE Signal Process. Lett.*, vol. 20, no. 12, pp. 1223–1226, Dec. 2013.
[33] X. Fu, K. Huang, W.-K. Ma, N. D. Sidiropoulos, and R. Bro, "Joint tensor factorization and outlying slab suppression with applications," *IEEE Trans. Signal Process.*, vol. 63, no. 23, pp. 6315–6328, Dec. 2015.
[34] X. Fu, K. Huang, B. Yang, W.-K. Ma, and N. D. Sidiropoulos, "Robust volume minimization-based matrix factorization for remote sensing and document clustering," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6254–6268, Dec. 2016.
[35] Z. Chen, R. Molina, and A. K. Katsaggelos, "Robust recovery of temporally smooth signals from under-determined multiple measurements," *IEEE Trans. Signal Process.*, vol. 63, no. 7, pp. 1779–1791, Apr. 2015.
[36] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.
[37] S. Kullback and R. A. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, pp. 79–86, 1951.
[38] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for Bayesian inference," *IEEE Signal Process. Mag.*, vol. 25, no. 6, pp. 131–146, Nov. 2008.
[39] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *Proc. IEEE Int. Symp. Inf. Theory Proc.*, 2011, pp. 2168–2172.
[40] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. Motivation and construction," in *Proc. Inf. Theory Workshop*, Cairo, Egypt, Jan. 2010, pp. 1–5.
[41] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Math. Program. Comput.*, vol. 4, no. 4, pp. 333–361, Dec. 2012.

Linxiao Yang received the B.S. degree from the Southwest Jiaotong University, Chengdu, China, in 2013. Since August 2015, he has been working toward the Ph.D. degree at the University of Electronic of Science and Technology of China, Chengdu, China. His current research interests include compressed sensing, sparse theory, tensor analysis, and machine learning.

Jun Fang (M'08) received the B.S. and M.S. degrees from the Xidian University, Xi'an, China, in 1998 and 2001, respectively, and the Ph.D. degree from the National University of Singapore, Singapore, in 2006, all in electrical engineering.

During 2006, he was a Postdoctoral Research Associate with the Department of Electrical and Computer Engineering, Duke University. From January 2007 to December 2010, he was a Research Associate with the Department of Electrical and Computer Engineering, Stevens Institute of Technology. Since 2011, he has been with the University of Electronic of Science and Technology of China, Chengdu, China. His research interests include compressed sensing and sparse theory, massive MIMO/mmWave communications, and statistical inference.

Dr. Fang received the IEEE Jack Neubauer Memorial Award in 2013 for the best systems paper published in the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He is an Associate Technical Editor for IEEE Communications Magazine, and an Associate Editor for IEEE SIGNAL PROCESSING LETTERS.

Huiping Duan received the Ph.D. degree in electrical engineering from Nanyang Technological University in 2008, and the M.S. and B.S. degrees from XiDian University, Xi'an, China, in 2001 and 1998, respectively. Since 2011, she has been with the University of Electronic of Science and Technology of China, Chengdu, China, where she is currently an Associate Professor. Her research interests include compressed sensing and sparse theory, array signal processing, and statistical inference.

Hongbin Li (M'99–SM'08) received the B.S. and M.S. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 1991 and 1994, respectively, and the Ph.D. degree from the University of Florida, Gainesville, FL, USA, in 1999, all in electrical engineering.

From July 1996 to May 1999, he was a Research Assistant with the Department of Electrical and Computer Engineering, University of Florida. Since July 1999, he has been with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA, where he became a Professor in 2010. He was a Summer Visiting Faculty Member with the Air Force Research Laboratory in the summers of 2003, 2004, and 2009. His general research interests include statistical signal processing, wireless communications, and radars.

Dr. Li received the IEEE Jack Neubauer Memorial Award in 2013 from the IEEE Vehicular Technology Soceity, the Outstanding Paper Award from the IEEE AFICON Conference in 2011, the Harvey N. Davis Teaching Award in 2003, and the Jess H. Davis Memorial Award for excellence in research in 2001 from Stevens Institute of Technology, and the Sigma Xi Graduate Research Award from the University of Florida in 1999. He has been a member of the IEEE SPS Signal Processing Theory and Methods Technical Committee (TC) and the IEEE SPS Sensor Array and Multichannel TC, an Associate Editor for *Signal Processing* (Elsevier), IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE SIGNAL PROCESSING LETTERS, and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, as well as a Guest Editor for IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING and *EURASIP Journal on Applied Signal Processing*. He has been involved in various conference organization activities, including serving as a General Co-Chair for the 7th IEEE Sensor Array and Multichannel Signal Processing (SAM) Workshop, Hoboken, NJ, USA, June 17–20, 2012. He is a member of Tau Beta Pi and Phi Kappa Phi.

Bing Zeng (M'91–SM'13–F'15) received the B.S. and M.S. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1983 and 1986, respectively, and the Ph.D. degree in electrical engineering from Tampere University of Technology, Tampere, Finland, in 1991.

He was a Postdoctoral Fellow with the University of Toronto from September 1991 to July 1992 and as a Researcher with Concordia University from August 1992 to January 1993. He then joined the Hong Kong University of Science and Technology (HKUST). After 20 years of service at HKUST, he joined UESTC in the summer of 2013, through China "1000-Talent-Scheme." At UESTC, he leads the Institute of Image Processing to work on image and video processing, 3-D and multiview video technology, and visual big data.

During his tenure at HKUST and UESTC, he graduated more than 30 Master and Ph.D. students, received 25 research grants, filed eight international patents, and published more than 200 papers. Three representing works are as follows: one paper on fast block motion estimation, published in IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT) in 1994, has so far been SCI-cited more than 880 times (Google-cited more than 1980 times) and currently stands at the 7th position among all papers published in this Transactions; one paper on smart padding for arbitrarily-shaped image blocks, published in IEEE TCSVT in 2001, leads to a patent that has been successfully licensed to companies; and one paper on directional discrete cosine transform (DDCT), published in IEEE TCSVT in 2008, receives the 2011 IEEE TCSVT Transactions Best Paper Award. He also received the best paper award at ChinaCom three times (2009 Xi'an, 2010 Beijing, and 2012 Kunming). He served as an Associate Editor for IEEE TCSVT for eight years and received the Best Associate Editor Award in 2011. He is currently on the Editorial Board of *Journal of Visual Communication and Image Representation* and serves as the General Co-Chair of IEEE VCIP-2016 to be held in Chengdu, China, in November 2016. He received a 2nd Class Natural Science Award (the first recipient) from Chinese Ministry of Education in 2014 and was elected as an IEEE Fellow in 2015 for contributions to image and video coding.