Sparse Bayesian Methods for Low-Rank Matrix Estimation

S. Derin Babacan, *Member, IEEE*, Martin Luessi, *Student Member, IEEE*, Rafael Molina, *Member, IEEE*, Aggelos K. Katsaggelos, *Fellow, IEEE*

Abstract

Recovery of low-rank matrices has recently seen significant activity in many areas of science and engineering, motivated by recent theoretical results for exact reconstruction guarantees and interesting practical applications. A number of methods have been developed for this recovery problem. However, a principled method for choosing the unknown target rank is generally not provided. In this paper, we present novel recovery algorithms for estimating low-rank matrices in matrix completion and robust principal component analysis based on sparse Bayesian learning (SBL) principles. Starting from a matrix factorization formulation and enforcing the low-rank constraint in the estimates as a sparsity constraint, we develop an approach that is very effective in determining the correct rank while providing high recovery performance. We provide connections with existing methods in other similar problems and empirical results and comparisons with current state-of-the-art methods that illustrate the effectiveness of this approach.

I. Introduction

There has been recently a significant interest in problems involving the estimation of low-rank matrices. This is motivated by recent theoretical advances [1]–[4], as well as interesting practical problems where the underlying data resides in a low-dimensional linear subspace. Incorporating a low-rank constraint on the data to be processed leads to new and powerful modeling options for many applications in science and engineering.

A typical example is the *matrix completion* problem, where an unknown (approximately) low-rank matrix is estimated from its limited set of observed entries. Although this problem is not new [5], interesting and challenging problems (e.g., the *Netflix prize*) along with recently developed theoretical recovery guarantees [1], [2] created a rapidly growing interest in this area. Matrix completion finds application in many areas of engineering, including system identification [6], sensor networks [7], machine learning [8], computer vision [9], [10], and medical imaging [11].

A second important problem is *robust principal component analysis* (RPCA), where the high dimensional data is assumed to lie in a lower-dimensional subspace with a small number of the data points corrupted with (arbitrarily) large errors. Widely used classical methods, such as principal component analysis (PCA), often fail to provide meaningful results in these cases. Some earlier methods attempt to overcome these issues using robust statistics [12]–[17]. Recently, theoretical performance guarantees for RPCA have been developed in [3], where it is shown that a data matrix can be decomposed into its low-rank and sparse components via convex optimization. Robust PCA has many important applications, such as video surveillance (background/foreground separation in video), face recognition [18], latent semantic indexing [19], image alignment [20], among many others.

Mathematically, problems involving the estimation of low-rank matrices can be formulated in a common framework as follows. Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ be an unknown matrix with rank $r \ll \min(m,n)$. Suppose that one is given an observation matrix $\mathbf{Y}$ which is a function $f(\mathbf{X})$ of $\mathbf{X}$. In matrix completion, the observation is a subset Ω of its entries, that is, $\{Y_{ij} = X_{ij} : (i,j) \in \Omega\}$. In other words, the observation $\mathbf{Y}$ is a projection $\mathcal{P}_\Omega$ of $\mathbf{X}$ on a subset Ω of its entries, such that the $(i,j)^{\text{th}}$ component of $\mathbf{Y}$ is equal to X_{ij} if $(i,j) \in \Omega$ and zero otherwise. In RPCA, the

S. Derin Babacan is with the Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, Urbana, IL, USA. e-mail: dbabacan@illinois.edu

Martin Luessi is with the Department of Electrical Engineering and Computer Science, Northwestern University, IL, USA. e-mail: m-luessi@northwestern.edu

Rafael Molina is with the Departamento de Ciencias de la Computación e I.A. Universidad de Granada, Spain. e-mail: rms@decsai.ugr.es

Aggelos K. Katsaggelos is with the Department of Electrical Engineering and Computer Science, Northwestern University, IL, USA. e-mail: aggk@eecs.northwestern.edu

observation can be expressed as $\mathbf{Y} = \mathbf{X} + \mathbf{E}$, where $\mathbf{E}$ is a sparse error matrix where only a very small number of coefficients are nonzero with (arbitrarily) large magnitudes.

In both cases, most matrices can be recovered by solving the affine rank minimization problem[1] [1]–[4]

$$\begin{array}{ll} \text{minimize} & \mathrm{rank}(\mathbf{X}) \\ \text{subject to} & \mathbf{Y} = f(\mathbf{X}). \end{array} \tag{1}$$

Although this optimization guarantees exact recovery of $\mathbf{X}$ under a set of conditions [1], [3], it is NP-hard and no known polynomial-time algorithms exist (analogous to the l_0-norm based recovery approaches in compressive sensing). A popular approach is to utilize convex relaxation based on the nuclear norm, given by

$$\begin{array}{ll} \text{minimize} & \|\mathbf{X}\|_* \\ \text{subject to} & \mathbf{Y} = f(\mathbf{X}), \end{array} \tag{2}$$

where $\|\mathbf{X}\|_*$ is equal to the sum of the singular values of $\mathbf{X}$. Formal equivalence of these two problems is established in [1] and recovery guarantees exist under some conditions (see, for example, [3], [21]). Subsequent works [2], [22], [23] improved on the theoretical recovery guarantees for the matrix completion problem.

If the observed entries are corrupted by dense (non-sparse) noise, the problem in (2) becomes

$$\begin{array}{ll} \text{minimize} & \|\mathbf{X}\|_* \\ \text{subject to} & \| \mathbf{Y} - f(\mathbf{X}) \|_{\mathrm{F}}^2 < \epsilon, \end{array} \tag{3}$$

where $\| \cdot \|_{\mathrm{F}}$ denotes the Frobenius norm. Both nuclear norm based optimization problems in (2) and (3) can be recast as a semidefinite program, and can be solved with interior-point solvers [6], [24]. Although they provide good empirical results, these methods can be inefficient when the matrix size is large.

A number of methods have been developed consequently for different problems involving low-rank estimation. For matrix completion, singular value thresholding [25] and projection methods [26] are attractive in terms of computation, while they nearly optimize (2). FPCA [27] introduced an efficient nuclear norm-based regularized least-squares method, whereas OPTSPACE [22] developed a method based on optimization over the Grasmann manifold with a theoretical performance guarantee for the noiseless case. Similarly to the approaches for compressive sensing recovery, greedy approaches have been proposed for matrix completion [28]. Finally, Bayesian methods have also been developed: a nonparametric approach for symmetric positive definite matrices is proposed in [29], and a variational Bayes method is developed for collaborative filtering in [30].

For robust PCA, the original work in [3] proposed iterative thresholding methods with low complexity, but their convergence is generally very slow. Lin *et al.* [31] proposed accelerated proximal gradient (APG) methods which are faster and generally more accurate. The augmented Lagrange Multiplier Method (ALM) [32] is, to the best of our knowledge, the state-of-the-art method for robust PCA in terms of both speed and accuracy. However, algorithm parameters need to be tuned carefully to obtain the best performance. The Bayesian method proposed in [33] attempts to overcome this issue by simultaneously estimating the necessary parameters along with the unknowns, but the resulting algorithm is quite complex and slow in practice.

In this paper, we present a novel Bayesian formulation for low-rank matrix recovery based on the sparse Bayesian learning principles. We specifically consider the matrix completion and robust principal component analysis problems, but the proposed framework can be translated to other problems involving low-rank structures. Based on the low-rank factorization of the unknown matrix, we employ independent sparsity priors on the individual factors with a common sparsity profile which favors low-rank solutions. Other elements in the problems are also modeled using a hierarchical Bayesian framework for simultaneous and automated estimation.

The proposed Bayesian formulation offers several advantages over other approaches. Firstly, prior knowledge on the rank of the matrix is not required; the proposed formulation implicitly estimates the rank of the unknown matrix similarly to the automatic relevance determination principle in machine learning [34]. This property is not present in most of the existing approaches. Second, algorithmic parameters are treated as stochastic quantities in the proposed approach, and are handled with the combination of prior distributions and fully-Bayesian inference procedures. In this regard, this type of formulation frees the user from extensive parameter-tuning and data- and application-dependent supervision. Finally, empirical results demonstrate that the proposed methods provide very

[1]A sparsity term is also incorporated in the objective function in the robust PCA case, which is omitted here for generality.

good reconstruction performance compared to existing methods while accurately estimating the unknown effective rank.

The rest of this paper is organized as follows. We present the proposed Bayesian modeling in Section II. Section III develops the estimation algorithms based on variational Bayesian inference. We present an analysis of the proposed approach in Section IV and empirical results with synthetic and real data in Section V, and finally conclude in Section VI.

II. Bayesian Modeling

In order to simultaneously estimate all latent variables, we make use of a hierarchical Bayesian framework where all observed and unknown quantities are treated as stochastic quantities and their joint probability distribution is specified. For tractable mathematical modeling, this distribution is given in a factorized form using a generative model where each factor is a prior or a conditional distribution used to model a specific quantity. We provide the description of each distribution used in this work in the following sections.

A. Proposed Low-Rank Modeling

Our modeling is based on the low-rank parametrization of the unknown matrix $\mathbf{X}$, given by

$$\mathbf{X} = \mathbf{A}\mathbf{B}^T \,, \tag{4}$$

where $\mathbf{A}$ is an $m \times r$ matrix, and $\mathbf{B}$ an $n \times r$ matrix, such that $\mathrm{rank}(\mathbf{X}) = r \leq \min(m, n)$. Any matrix of rank r can be decomposed in this form, as can be seen by considering the singular value decomposition

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T = \left(\mathbf{U}\mathbf{S}^{1/2}\right)\left(\mathbf{S}^{1/2}\mathbf{V}^T\right) \,, \tag{5}$$

where $\mathbf{U}$ and $\mathbf{V}$ are respectively $m \times m$ and $n \times n$ matrices with orthogonal columns, and $\mathbf{S}$ is a $r \times r$ diagonal matrix of the non-zero singular values. Algorithms based on this factorization are commonly used for nonnegative matrix factorization [35] and matrix completion [36], which generally aim to find solutions to

$$\begin{array}{ll} \text{minimize} & \| \mathbf{A} \|_{\mathrm{F}}^2 + \| \mathbf{B} \|_{\mathrm{F}}^2 \\ \text{subject to} & \| \mathbf{Y} - f(\mathbf{X}) \|_{\mathrm{F}}^2 < \epsilon. \end{array} \tag{6}$$

The equivalence of this optimization problem to (3) is easy to show (see [21]). We formulate the problem in (6) using the Bayesian methodology as follows. It is clear from $\mathbf{X} = \mathbf{A}\mathbf{B}^T$ that $\mathbf{X}$ is the sum of outer-products of the columns of $\mathbf{A}$ and $\mathbf{B}$, that is,

$$\mathbf{X} = \sum_{i=1}^{k} \mathbf{a}_{\cdot i}\mathbf{b}_{\cdot i}{}^T \,, \tag{7}$$

where $k \geq r$ and we use $\mathbf{a}_{\cdot i}$ and $\mathbf{a}_{i\cdot}$ to denote the i^{th} column and row of $\mathbf{A}$, respectively. Notice that each outer-product contributes at most one to the rank to $\mathbf{X}$. Since a low-rank estimate of $\mathbf{X}$ is sought, our goal is to achieve column sparsity in $\mathbf{A}$ and $\mathbf{B}$, such that most columns in $\mathbf{A}$ and in $\mathbf{B}$ are set equal to zero. To enforce this constraint, we associate the columns of $\mathbf{A}$ and $\mathbf{B}$ with Gaussian priors of variances γ_i, that is,

$$\mathrm{p}(\mathbf{A}|\boldsymbol{\gamma}) = \prod_{i=1}^{k} \mathcal{N}\left(\mathbf{a}_{\cdot i}|\mathbf{0}, \gamma_i \mathbf{I}\right) \,, \tag{8}$$

$$\mathrm{p}(\mathbf{B}|\boldsymbol{\gamma}) = \prod_{i=1}^{k} \mathcal{N}\left(\mathbf{b}_{\cdot i}|\mathbf{0}, \gamma_i \mathbf{I}\right) \,. \tag{9}$$

Thus, the columns of $\mathbf{A}$ and $\mathbf{B}$ have the same sparsity profile enforced by the common variances γ_i. As shown later, many of the variances γ_i will assume very small values during inference, which effectively removes the corresponding outer-products from $\mathbf{X}$, and hence reduces the rank of the estimate. This formulation is therefore the analog of sparse Bayesian learning formulation (or automatic relevance determination) [34], [37] successfully utilized for compressive sensing reconstruction, where sparsity-inducing Gaussian priors are employed on each of the coefficients of the unknown vector.

Note also that (8) is equivalent to

$$\mathrm{p}(\mathbf{A}|\boldsymbol{\gamma}) \propto \exp\left(-\frac{1}{2}\mathrm{Tr}(\mathbf{A}^T\boldsymbol{\Gamma}\mathbf{A})\right) = \exp\left(-\frac{1}{2}\sum_{i=1}^{k}\gamma_i^{-1}\,\sigma_{A,i}^2\right), \tag{10}$$

with $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_k)$, $\boldsymbol{\Gamma} = \mathrm{diag}(\gamma_i^{-1})$ and $\sigma_{A,i}$ the i^{th} singular value of $\mathbf{A}$. Similarly, (9) is equivalent to

$$\mathrm{p}(\mathbf{B}|\boldsymbol{\gamma}) \propto \exp\left(-\frac{1}{2}\mathrm{Tr}(\mathbf{B}^T\boldsymbol{\Gamma}\mathbf{B})\right) = \exp\left(-\frac{1}{2}\sum_{i=1}^{k}\gamma_i^{-1}\,\sigma_{B,i}^2\right), \tag{11}$$

with $\sigma_{B,i}$ the i^{th} singular value of $\mathbf{B}$. Based on these expressions, the priors on $\mathbf{A}$ and $\mathbf{B}$ are equivalent to utilizing sparsity priors on the singular values of $\mathbf{A}$ and $\mathbf{B}$.

In addition to (8) and (9), we incorporate the conjugate inverse Gamma hyperprior on the variances γ_i

$$\mathrm{p}(\gamma_i) \propto \left(\frac{1}{\gamma_i}\right)^{a+1}\exp\left(-\frac{b}{\gamma_i}\right). \tag{12}$$

The parameters a and b are treated as deterministic whose values are set by the user (their selection is discussed later).

B. Observation and Noise Models

In this work, the prior structure in (8), (9) and (12) is used as a common low-rank matrix model for $\mathbf{X}$ in the matrix completion and robust PCA problems. The descriptions of the distributions used to model other latent and observed variables are provided in the following sections.

1) Matrix Completion: In matrix completion, the observations are generated according to

$$Y_{ij} = X_{ij} + N_{ij}, \quad (i,j) \in \Omega, \tag{13}$$

or in a more compact form as

$$\mathbf{Y} = \mathcal{P}_{\Omega}\left(\mathbf{X} + \mathbf{N}\right), \tag{14}$$

where $\mathbf{N}$ is the dense error matrix with coefficients N_{ij}. The cardinality of the set Ω is pmn, with p the fraction of observed coefficients. Using this model, we follow the standard assumption and incorporate white Gaussian noise in the observations, such that

$$\mathrm{p}(\mathbf{Y}|\mathbf{A},\mathbf{B},\beta) = \prod_{(i,j)\in\Omega}\mathcal{N}\left(Y_{ij}|X_{ij},\beta^{-1}\right), \tag{15}$$

with $\beta = 1/\epsilon$ the noise precision. The noise precision β is assigned the improper uniform prior

$$\mathrm{p}(\beta) = \mathrm{const.} \tag{16}$$

The joint distribution, therefore, is expressed as

$$\mathrm{p}(\mathbf{Y},\mathbf{A},\mathbf{B},\boldsymbol{\gamma},\beta) = \mathrm{p}(\mathbf{Y}|\mathbf{A},\mathbf{B},\beta)\,\mathrm{p}(\mathbf{A}|\boldsymbol{\gamma})\,\mathrm{p}(\mathbf{B}|\boldsymbol{\gamma})\mathrm{p}(\boldsymbol{\gamma})\,\mathrm{p}(\beta). \tag{17}$$

2) Robust PCA: In this case, the generative model can be expressed as $\mathbf{Y} = \mathbf{X} + \mathbf{E} + \mathbf{N}$, where $\mathbf{E}$ is the sparse error matrix with arbitrarily large coefficients, and $\mathbf{N}$ is the dense error matrix with relatively smaller coefficients. Using white Gaussian noise modeling on $\mathbf{N}$, we obtain the following conditional distribution for the observations

$$\begin{aligned}\mathrm{p}(\mathbf{Y}|\mathbf{A},\mathbf{B},\mathbf{E},\beta) &= \mathcal{N}\left(\mathbf{Y}|\mathbf{A}\mathbf{B}^T + \mathbf{E}, \beta^{-1}\mathbf{I}\right)\\ &\propto \exp\left(\frac{\beta}{2}\parallel \mathbf{Y} - \mathbf{A}\mathbf{B}^T - \mathbf{E} \parallel_{\mathrm{F}}^2\right).\end{aligned} \tag{18}$$

As in the matrix completion case, we assign the uniform hyperprior in (16) to β. The modeling of the sparse component $\mathbf{E}$ is done by employing independent Gaussian priors on each of the coefficients E_{ij} of the matrix $\mathbf{E}$, that is,

$$\mathrm{p}(\mathbf{E}|\boldsymbol{\alpha}) = \prod_{i=1}^{m}\prod_{j=1}^{n} \mathcal{N}\left(E_{ij}|0,\alpha_{ij}^{-1}\right) , \tag{19}$$

where $\boldsymbol{\alpha} = \{\alpha_{ij}\}$ and α_{ij} is the precision of the Gaussian on the $(i,j)^{\text{th}}$ coefficient. As with the noise precision, we use uniform priors on α_{ij}

$$\mathrm{p}(\alpha_{ij}) = \text{const}, \;\; \forall i,j\,. \tag{20}$$

Notice that when an individual precision goes to infinity, i.e., $\alpha_{ij}^{-1} \to 0$, the corresponding coefficient E_{ij} goes to zero. Hence, the sparsity in $\mathbf{E}$ is achieved when a large number of precision variables are set to high values. As in the original formulation of sparse Bayesian learning, this is achieved in this work by simultaneously estimating the coefficients E_{ij} and the precision variables α_{ij}, as shown later.

Finally, the joint distribution is expressed as

$$\mathrm{p}(\mathbf{Y},\mathbf{A},\mathbf{B},\mathbf{E},\boldsymbol{\gamma},\boldsymbol{\alpha},\beta) = \mathrm{p}(\mathbf{Y}|\mathbf{A},\mathbf{B},\mathbf{E},\beta)\,\mathrm{p}(\mathbf{A}|\boldsymbol{\gamma})\,\mathrm{p}(\mathbf{B}|\boldsymbol{\gamma})\,\mathrm{p}(\mathbf{E}|\boldsymbol{\alpha})\,\mathrm{p}(\boldsymbol{\gamma})\,\mathrm{p}(\boldsymbol{\alpha})\,\mathrm{p}(\beta). \tag{21}$$

III. Approximate Bayesian Inference

As is widely known, exact full-Bayesian inference using joint distributions such as (17) and (21) is intractable, since $\mathrm{p}(\mathbf{y})$ cannot be computed by marginalizing all latent variables. Therefore, approximation methods must be utilized. In this work, we present an inference procedure based on mean field variational Bayes [38], [39]. Our goal is to compute posterior distribution approximations by minimizing the Kullback-Leibler (KL) divergence in an alternating fashion for each latent variable. Let $\mathbf{z}$ be the vector of all latent variables such that $\mathbf{z} = (\mathbf{A},\mathbf{B},\boldsymbol{\gamma},\beta)$ for the matrix completion case, and $\mathbf{z} = (\mathbf{A},\mathbf{B},\mathbf{E},\boldsymbol{\gamma},\boldsymbol{\alpha},\beta)$ for robust PCA. The posterior approximation $\mathrm{q}(\mathbf{z}_k)$ of each latent variable $\mathbf{z}_k \in \mathbf{z}$ is found using

$$\log \mathrm{q}(\mathbf{z}_k) = \left\langle \log \mathrm{p}(\mathbf{Y},\mathbf{z}) \right\rangle_{\mathbf{z}\backslash\mathbf{z}_k} + \text{const}, \tag{22}$$

where $\mathbf{z}\backslash\mathbf{z}_k$ denotes the set $\mathbf{z}$ with $\mathbf{z}_k$ removed. The distribution $\mathrm{p}(\mathbf{Y},\mathbf{z})$ is the joint probability distribution given in (17) for the matrix completion problem, and in (21) for robust PCA.

Using mean field approximation, we employ the posterior factorization $\mathrm{q}(\mathbf{z}) = \prod \mathrm{q}(\mathbf{z}_k)$ such that the posterior distribution of each unknown is estimated by holding the others fixed using their most recent distributions. Thus, for each latent variable, the expectations of all parameters (excluding the current one) in the joint distribution are taken with respect to their most recent distributions, and the result is normalized to find the approximate posterior distribution. Since all distributions in the hierarchical model presented in the previous section are in the conjugate exponential family, the form of each posterior approximation can be found without major difficulties. We present the update rules resulting from this inference scheme for each problem in the following subsections.

A. Inference for Matrix Completion

1) Estimation of factors $\mathbf{A}$ *and* $\mathbf{B}$*:* With some algebra, it follows from (22) that the approximation to the posterior distributions of $\mathbf{A}$ and $\mathbf{B}$ decompose as independent distributions of their rows. By combining the prior in (8) and the observation model in (15), the posterior density of the i^{th} row $\mathbf{a}_{i\cdot}$ of $\mathbf{A}$ is found as

$$\mathrm{q}(\mathbf{a}_{i\cdot}) = \mathcal{N}\left(\mathbf{a}_{i\cdot}|\langle\mathbf{a}_{i\cdot}\rangle, \boldsymbol{\Sigma}_i^a\right) , \tag{23}$$

with mean and covariance

$$\langle\mathbf{a}_{i\cdot}\rangle^T = \langle\beta\rangle\, \boldsymbol{\Sigma}_i^a \,\langle\mathbf{B}_i\rangle^T\, \mathbf{y}_{i\cdot}{}^T , \tag{24}$$

$$\boldsymbol{\Sigma}_i^a = \left(\langle\beta\rangle\,\langle\mathbf{B}_i^T\mathbf{B}_i\rangle + \boldsymbol{\Gamma}\right)^{-1} , \tag{25}$$

where the matrix $\mathbf{B}_i$ contains only the j^{th} rows of $\mathbf{B}$ for which $(i,j) \in \Omega$, such that,

$$\langle\mathbf{B}_i^T\mathbf{B}_i\rangle = \sum_{j:(i,j)\in\Omega} \langle\mathbf{b}_{j\cdot}{}^T\,\mathbf{b}_{j\cdot}\rangle = \sum_{j:(i,j)\in\Omega} \left(\langle\mathbf{b}_{j\cdot}{}^T\rangle\langle\mathbf{b}_{j\cdot}\rangle + \boldsymbol{\Sigma}_j^b\right) , \tag{26}$$

with $\mathbf{\Sigma}_j^b$ the posterior covariance of the j^{th} row of $\mathbf{B}$. Additionally, the row vector $\mathbf{y}_{i\cdot}$ contains the observed entries in the i^{th} row of $\mathbf{Y}$. Similarly, by combining the prior in (9) and the observation model in (15), the posterior density of the j^{th} row $\mathbf{b}_{j\cdot}$ of $\mathbf{B}$ is found as a normal distribution

$$\mathrm{q}(\mathbf{b}_{j\cdot}) = \mathcal{N}\left(\mathbf{b}_{j\cdot} | \langle \mathbf{b}_{j\cdot} \rangle, \mathbf{\Sigma}_j^b\right) \tag{27}$$

with parameters

$$\langle \mathbf{b}_{j\cdot} \rangle^T = \langle \beta \rangle \, \mathbf{\Sigma}_j^b \, \langle \mathbf{A}_j \rangle^T \, \mathbf{y}_{\cdot j} \,, \tag{28}$$

$$\mathbf{\Sigma}_j^b = \left(\langle \beta \rangle \, \langle \mathbf{A}_j^T \mathbf{A}_j \rangle + \mathbf{\Gamma} \right)^{-1} \,, \tag{29}$$

where $\mathbf{A}_j$ contains the i^{th} rows of $\mathbf{A}$ for which $(i,j) \in \Omega$, and the vector $\mathbf{y}_{\cdot j}$ is constructed from the observed entries in the j^{th} column of $\mathbf{Y}$. It can be observed that the uncertainty in the estimate of $\mathbf{B}$ is incorporated in the estimation of $\mathbf{A}$ through the covariance matrices $\mathbf{\Sigma}_i^b$ (and vice versa).

In the case of no noise, that is, $\langle \beta \rangle^{-1} \to 0$, the posterior moments become

$$\langle \mathbf{a}_{i\cdot} \rangle^T = \mathbf{\Gamma}^{-1/2} \left(\langle \mathbf{B}_i \rangle \mathbf{\Gamma}^{-1/2} \right)^{\dagger} {\mathbf{y}_{i\cdot}}^T, \tag{30}$$

$$\mathbf{\Sigma}_i^a = \left[\mathbf{I} - \mathbf{\Gamma}^{-1/2} \left(\langle \mathbf{B}_i \rangle \mathbf{\Gamma}^{-1/2} \right)^{\dagger} \langle \mathbf{B}_i \rangle \right] \mathbf{\Gamma}^{-1} \,, \tag{31}$$

$$\langle \mathbf{b}_{j\cdot} \rangle^T = \mathbf{\Gamma}^{-1/2} \left(\langle \mathbf{A}_j \rangle \mathbf{\Gamma}^{-1/2} \right)^{\dagger} \mathbf{y}_{\cdot j}, \tag{32}$$

$$\mathbf{\Sigma}_j^b = \left[\mathbf{I} - \mathbf{\Gamma}^{-1/2} \left(\langle \mathbf{A}_j \rangle \mathbf{\Gamma}^{-1/2} \right)^{\dagger} \langle \mathbf{A}_j \rangle \right] \mathbf{\Gamma}^{-1} \,, \tag{33}$$

where $(\cdot)^{\dagger}$ denotes the Moore-Penrose pseudo-inverse. If *a priori* knowledge is available that the observations are noiseless (such as the problem in (2)), these expressions can be applied very effectively for drawing inference.

2) Estimation of hyperparameters $\boldsymbol{\gamma}$: By combining $\mathrm{p}(\mathbf{A}|\boldsymbol{\gamma})$, $\mathrm{p}(\mathbf{B}|\boldsymbol{\gamma})$ and $\mathrm{p}(\gamma_i)$, the posterior density of γ_i becomes an inverse Gamma distribution

$$\mathrm{q}(\gamma_i) \propto \left(\frac{1}{\gamma_i} \right)^{a+1+\frac{m+n}{2}} \exp\left(-\frac{2b + \langle {\mathbf{a}_{\cdot i}}^T \mathbf{a}_{\cdot i} \rangle + \langle {\mathbf{b}_{\cdot i}}^T \mathbf{b}_{\cdot i} \rangle}{2\gamma_i} \right) \tag{34}$$

with mean

$$\langle \gamma_i \rangle = \frac{2b + \langle {\mathbf{a}_{\cdot i}}^T \mathbf{a}_{\cdot i} \rangle + \langle {\mathbf{b}_{\cdot i}}^T \mathbf{b}_{\cdot i} \rangle}{2a + m + n} \,. \tag{35}$$

The required expectations are given by

$$\langle {\mathbf{a}_{\cdot i}}^T \mathbf{a}_{\cdot i} \rangle = \langle \mathbf{a}_{\cdot i} \rangle^T \langle \mathbf{a}_{\cdot i} \rangle + \sum_j \left(\mathbf{\Sigma}_j^a \right)_{ii} \,, \tag{36}$$

$$\langle {\mathbf{b}_{\cdot i}}^T \mathbf{b}_{\cdot i} \rangle = \langle \mathbf{b}_{\cdot i} \rangle^T \langle \mathbf{b}_{\cdot i} \rangle + \sum_j \left(\mathbf{\Sigma}_j^b \right)_{ii} \,. \tag{37}$$

3) Estimation of noise precision β: The Bayesian methodology allows for the estimation of the noise precision as well. From (22), the posterior approximation assumes a Gamma distribution with mean

$$\langle \beta \rangle = \frac{pmn}{\langle \, \| \, \mathbf{Y} - \mathcal{P}_{\Omega} \left(\mathbf{A}\mathbf{B}^T \right) \|_{\mathrm{F}}^2 \, \rangle} \,. \tag{38}$$

However, this estimation may lead to identifiability problems [40], and therefore may in some cases cause instability, especially if the number of missing values is large. In practice, we found out that the algorithm is quite robust to this parameter and setting it to a reasonable value leads to good empirical results.

In summary, the algorithm proceeds by first estimating the rows of $\mathbf{A}$ and $\mathbf{B}$ using (24) and (28), respectively, followed by the estimation of the variances γ_i using (35), and (if desired) the noise precision β using (38). By the properties of the variational Bayes methods [39], the algorithm is guaranteed to converge to a local minimum.

B. Inference for Robust PCA

1) Estimation of factors $\mathbf{A}$ *and* $\mathbf{B}$*:* The approximations to the posterior distributions of $\mathbf{A}$ and $\mathbf{B}$ take forms similar to (23) and (27) with the same factorization over the rows of $\mathbf{A}$ and $\mathbf{B}$, respectively. However, as opposed to the matrix completion case, the rows of $\mathbf{A}$ and $\mathbf{B}$ have the same covariance matrices since there are no missing values. The posterior approximation of the i^{th} row of $\mathbf{A}$ is given by

$$\mathrm{q}(\mathbf{a}_{i\cdot}) = \mathcal{N}\left(\mathbf{a}_{i\cdot}|\langle\mathbf{a}_{i\cdot}\rangle, \mathbf{\Sigma}^A\right) , \tag{39}$$

with mean and covariance

$$\langle\mathbf{a}_{i\cdot}\rangle^T = \langle\beta\rangle\, \mathbf{\Sigma}^A \langle\mathbf{B}\rangle^T \ (\mathbf{y}_{i\cdot} - \mathbf{e}_{i\cdot})^T , \tag{40}$$

$$\mathbf{\Sigma}^A = \left(\langle\beta\rangle \langle\mathbf{B}^T\mathbf{B}\rangle + \mathbf{\Gamma}\right)^{-1} . \tag{41}$$

Similarly, the posterior approximation of $\mathbf{b}_{j\cdot}$ is another multivariate normal distribution given by

$$\mathrm{q}(\mathbf{b}_{j\cdot}) = \mathcal{N}\left(\mathbf{b}_{j\cdot}|\langle\mathbf{b}_{j\cdot}\rangle, \mathbf{\Sigma}^B\right) \tag{42}$$

with parameters

$$\langle\mathbf{b}_{j\cdot}\rangle^T = \langle\beta\rangle\, \mathbf{\Sigma}^B \langle\mathbf{A}\rangle^T \ (\mathbf{y}_{\cdot j} - \mathbf{e}_{\cdot j}) , \tag{43}$$

$$\mathbf{\Sigma}^B = \left(\langle\beta\rangle \langle\mathbf{A}^T\mathbf{A}\rangle + \mathbf{\Gamma}\right)^{-1} . \tag{44}$$

The required expectations can be found as

$$\langle\mathbf{A}^T\mathbf{A}\rangle = \langle\mathbf{A}\rangle^T\langle\mathbf{A}\rangle + m\mathbf{\Sigma}^A, \tag{45}$$

$$\langle\mathbf{B}^T\mathbf{B}\rangle = \langle\mathbf{B}\rangle^T\langle\mathbf{B}\rangle + n\mathbf{\Sigma}^B . \tag{46}$$

Using these updates, the estimate of $\mathbf{X}$ is then found by $\mathbf{X} = \langle\mathbf{A}\rangle\langle\mathbf{B}\rangle^T$. Similar expressions as in (30)-(33) can be derived for (40)-(41) and (43)-(44) in the noiseless case with $\beta^{-1} \to 0$.

2) Estimation of $\mathbf{E}$*:* Using (22), the posterior distribution approximation of $\mathbf{E}$ is found to be factorized on each coefficient E_{ij} with distributions

$$\mathrm{q}(E_{ij}) = \mathcal{N}\left(E_{ij}|\langle E_{ij}\rangle, \Sigma_{ij}^E\right) , \tag{47}$$

with parameters

$$\langle E_{ij}\rangle = \langle\beta\rangle\, \Sigma_{ij}^E \left(Y_{ij} - \langle\mathbf{a}_{i\cdot}\rangle\langle\mathbf{b}_{j\cdot}\rangle^T\right), \tag{48}$$

$$\Sigma_{ij}^E = \frac{1}{\langle\beta\rangle + \langle\alpha_{ij}\rangle} . \tag{49}$$

Notice that this update corresponds to a shrinkage in coefficients E_{ij} controlled by the noise precision β and hyperparameters α_{ij}.

3) Estimation of hyperparameters $\boldsymbol{\gamma}$*:* Similarly to the above, the posterior density of $\boldsymbol{\gamma}$ is found as an inverse Gamma distribution with mean given in (35). The only difference is in the calculation of the expectations, which are given by

$$\langle\mathbf{a}_{\cdot i}{}^T\mathbf{a}_{\cdot i}\rangle = \langle\mathbf{a}_{\cdot i}\rangle^T\langle\mathbf{a}_{\cdot i}\rangle + m\left(\mathbf{\Sigma}^A\right)_{ii}, \tag{50}$$

$$\langle\mathbf{b}_{\cdot i}{}^T\mathbf{b}_{\cdot i}\rangle = \langle\mathbf{b}_{\cdot i}\rangle^T\langle\mathbf{b}_{\cdot i}\rangle + n\left(\mathbf{\Sigma}^B\right)_{ii} . \tag{51}$$

4) Estimation of hyperparameters $\boldsymbol{\alpha}$*:* The posterior density of hyperarameters α_{ij} is found as a Gamma distribution with mean

$$\langle\alpha_{ij}\rangle = \frac{1}{\langle E_{ij}\rangle^2 + \Sigma_{ij}^E} . \tag{52}$$

5) Estimation of noise precision β: Finally, the posterior approximation of the noise precision assumes a Gamma distribution with mean

$$\langle \beta \rangle = \frac{mn}{\langle \parallel \mathbf{Y} - \mathbf{A}\mathbf{B}^T - \mathbf{E} \parallel_{\mathrm{F}}^2 \rangle}, \tag{53}$$

where

$$\begin{aligned} \langle \parallel \mathbf{Y} - \mathbf{A}\mathbf{B}^T - \mathbf{E} \parallel_{\mathrm{F}}^2 \rangle = & \parallel \mathbf{Y} - \langle \mathbf{A} \rangle \langle \mathbf{B} \rangle^T - \langle \mathbf{E} \rangle \parallel_{\mathrm{F}}^2 + \mathrm{trace}\left(n \langle \mathbf{A} \rangle^T \langle \mathbf{A} \rangle \mathbf{\Sigma}^B \right) \\ & + \mathrm{trace}\left(m \langle \mathbf{B} \rangle^T \langle \mathbf{B} \rangle \mathbf{\Sigma}^A \right) + \mathrm{trace}\left(mn \, \mathbf{\Sigma}^A \mathbf{\Sigma}^B \right) + \sum_{i=1}^{m} \sum_{j=1}^{n} \Sigma_{ij}^E . \end{aligned} \tag{54}$$

It should be noted that in theory, the identifiability problem in the matrix completion case still exists with the estimation of β. However, as opposed to the matrix completion case, we did not observe practical problems with the incorporation of this estimation.

In summary, the proposed algorithm estimates the low rank component $\mathbf{X}$ by estimating its factors using (40) and (43), followed by the estimation of the sparse matrix $\mathbf{E}$ using (48), and finally the estimation of all hyperparameters using (35), (52) and (53), until convergence.

IV. Discussion

A. Related Prior Art

The methodology presented in this work resembles some methods developed for collaborative filtering, probabilistic principal component analysis (PCA) and (nonnegative) matrix factorization. In the collaborative filtering method proposed in [30], independent Gaussian priors are placed on the columns of $\mathbf{A}$ and $\mathbf{B}$ with separate sets of variances, and a variational Bayesian analysis is employed for inference. Although the modeling is similar, the columns of $\mathbf{A}$ and $\mathbf{B}$ are not coupled through the use of common variances as in our work. Employing common parameters is of crucial importance in removing redundant components from the estimated matrix and determining the effective rank. In theory, the modeling in (8) and (9) with common variances is used to represent the correlation between the columns of $\mathbf{A}$ and $\mathbf{B}$, and it also removes possible scale problems due to the use of separate sets of variances. To cope with scalability issues, [30] uses fixed, heuristically selected values for one set, and estimates the other hyperparameter set. Finally, in contrast to our work, [30] has reported that no sparsity in the variances occurs during the application of their algorithm.

The Bayesian PCA methods [41]–[43] also have some similarity with our approach (with a different prior structure); these methods can be seen as marginalizing the matrix $\mathbf{B}$ out from the joint distribution and estimating $\mathbf{A}$ only. Although a similar approach can be developed in our formulation, i.e., marginalize one matrix factor to estimate the other, estimation of the common variances γ_i becomes problematic since $\mathbf{A}$ and $\mathbf{B}$ cannot be integrated out together from the joint distribution. Another work similar to our approach is [44], which aims at solving the nonnegative matrix factorization problem with a similar prior structure on $\mathbf{A}$ and $\mathbf{B}$. This work, however, employs nonnegative priors on $\mathbf{A}$ and $\mathbf{B}$, and resorts to a multiplicative MAP based estimation procedure for the sake of maintaining nonnegativity in $\mathbf{A}$ and $\mathbf{B}$. Note also that these methods have not been developed to handle the missing values as in the matrix completion problem, or the large sparse errors as in the robust PCA problem. Some statistical approaches have been developed in [16], [17] using heavy-tailed distributions for robust estimation against outliers, but these do not include explicit modeling of sparse errors and hence cannot separate these from dense errors.

B. Estimating the effective rank

The proposed algorithm enforces low-rank solutions by enforcing column sparsity in $\mathbf{A}$ and $\mathbf{B}$. During inference, most of the hyperparameters γ_i are driven to zero, which will force the posterior means of the columns to go to zero as well. In our implementation, columns of $\mathbf{A}$ and $\mathbf{B}$ were declared irrelevant if the corresponding γ_i is less than a threshold. We typically use $\gamma_i < 10^{-10}$ for $a = 10^5$ and $b = 10^{-5}$ or $\gamma_i < 10^{-12}$ for $a = 10^6$ and $b = 10^{-6}$. Such a selection of a and b strongly enforces column sparsity in $\mathbf{A}$ and $\mathbf{B}$, as the variances of the priors in (12) are close to zero. This is in contrast to classical sparse Bayesian learning methods [34], [37] where generally broad priors (with a and b small) are employed. In our experiments, we observed that such selections of a and b (including zero values) resulted in similar reconstruction errors but slower convergence speeds. In all cases, the threshold value of γ_i should be chosen according to the minimum value possible in (35) (i.e., when $\langle \mathbf{a}_{\cdot i}{}^T \mathbf{a}_{\cdot i} \rangle \approx 0$ and $\langle \mathbf{b}_{\cdot i}{}^T \mathbf{b}_{\cdot i} \rangle \approx 0$).

C. Sparsity of the estimate of $\mathbf{E}$

As discussed in Section III-B2, the update procedure (48) of the coefficients E_{ij} is in fact a shrinkage procedure, where the amount of shrinkage is controlled by the estimates of both the noise precision β and the hyperparameters α_{ij}. This resembles closely the automatic relevance determination in the original work of relevance vector machines [34]. During the iterative procedure, many of the estimated precisions α_{ij} will approach very high values, which makes the corresponding posteriors in (47) very sharply peaked at zero. In the limit of $\alpha_{ij} \to \infty$, the posterior is infinitely peaked at zero, leading to a zero estimate of $\langle E_{ij} \rangle$ in (48). In our implementation, we prune the coefficients E_{ij} with large corresponding α_{ij} values (e.g., 10^{16}) via thresholding, leading to a sparse estimate of $\mathbf{E}$. We have also observed empirically that instead of (52), using the fixed-point updates

$$\langle \alpha_{ij} \rangle^{\text{new}} = \frac{1 - \langle \alpha_{ij} \rangle^{\text{old}} \Sigma_{ij}^{E}}{\langle E_{ij} \rangle^2} \tag{55}$$

lead to much faster convergence and enhanced sparsity. This is also in agreement with the original formulation of sparse Bayesian learning in [34], [40].

D. Computational Complexity

While the proposed algorithms have demonstrated good empirical performance for a variety of matrix completion and robust PCA problems, care must be taken when applied to large scale problems. In matrix completion, the computation of the inverse matrices in (25) and (29) can be quite expensive; their computation is $\mathrm{O}(k^3)$, where k is the number of columns in each $\mathbf{A}_j$ matrix (or the number of columns in each $\mathbf{B}_i$ matrix). k is also equal to the estimated rank at each iteration. However, by construction, many rows of $\mathbf{A}$ ($\mathbf{B}$) are removed to obtain $\mathbf{A}_j$ ($\mathbf{B}_i$), such that $\mathbf{A}_j$ ($\mathbf{B}_i$) might possibly have fewer rows than columns. Each $\mathbf{A}_j$ has on the average pm rows and k columns (recall p is the fraction of observed entries to the matrix size with $p < 1$). If $pm < k$, we can utilize the Woodbury identity [45] to obtain a different form for $\mathbf{\Sigma}_j^b$, given by

$$\mathbf{\Sigma}_j^b = \mathbf{\Gamma}^{-1} - \mathbf{\Gamma}^{-1} \langle \mathbf{A}_j \rangle^T \left(\langle \beta \rangle^{-1} \mathbf{I} + \langle \mathbf{A}_j \rangle \mathbf{\Gamma}^{-1} \langle \mathbf{A}_j \rangle^T \right)^{-1} \langle \mathbf{A}_j \rangle \mathbf{\Gamma}^{-1}, \tag{56}$$

which has the average-case complexity $\mathrm{O}(p^3\, m^3)$. In practice, we compare the number of columns and rows in $\mathbf{A}_j$ and $\mathbf{B}_i$ at each iteration to automatically choose the least complexity update. Overall, the complexity of the algorithm is $\mathrm{O}(m \cdot \min(p^3 n^3, k^3) + n \cdot \min(p^3 m^3, k^3))$. Empirically, however, we observed that convergence is rapid; most of the variances become negligible in the very first iterations and can be removed from the model by simple thresholding (similarly to [34]). Other optimizations can also be implemented such as using conjugate gradients to solve for posterior means in (24) and (28), and avoiding the computation of the off-diagonal terms of $\mathbf{\Sigma}_i^a$ and $\mathbf{\Sigma}_j^b$. These optimizations will lead to decreased computational complexity at the expense of recovery performance.

In the robust PCA case, an analysis similar to the above (using similar identities as (56)) gives an overall computational complexity of $\mathrm{O}(\min(n^3, k^3) + \min(m^3, k^3))$ per iteration. However, as in the matrix completion case, the effective rank is generally reduced rapidly in the first few iterations, therefore resulting in a very efficient inference scheme.

E. Initialization

Although randomly initializing the matrices $\mathbf{A}$ and $\mathbf{B}$ generally provided satisfactory results, faster convergence and better reconstruction performance can be achieved by more carefully selecting the initial values. In our implementations, we calculate the SVD of the matrix $\mathbf{Y} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ and set $\mathbf{A} = \mathbf{U}\mathbf{S}^{\frac{1}{2}}$ and $\mathbf{B}^T = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^T$. With this choice, the algorithm is initialized with a (near) full-rank matrix $\mathbf{Y}$. On the other hand, one can initialize the algorithm with a lower rank estimate by removing columns of $\mathbf{A}$ and $\mathbf{B}$ which correspond to small eigenvalues of $\mathbf{Y}$. Empirical results show negligible difference in performance if a reasonable initial rank (larger than the true rank) is chosen, whereas the computational complexity can be significantly reduced. Moreover, independently of the initial rank, the algorithm successfully removes irrelevant components from the estimate and estimates the effective rank accurately.

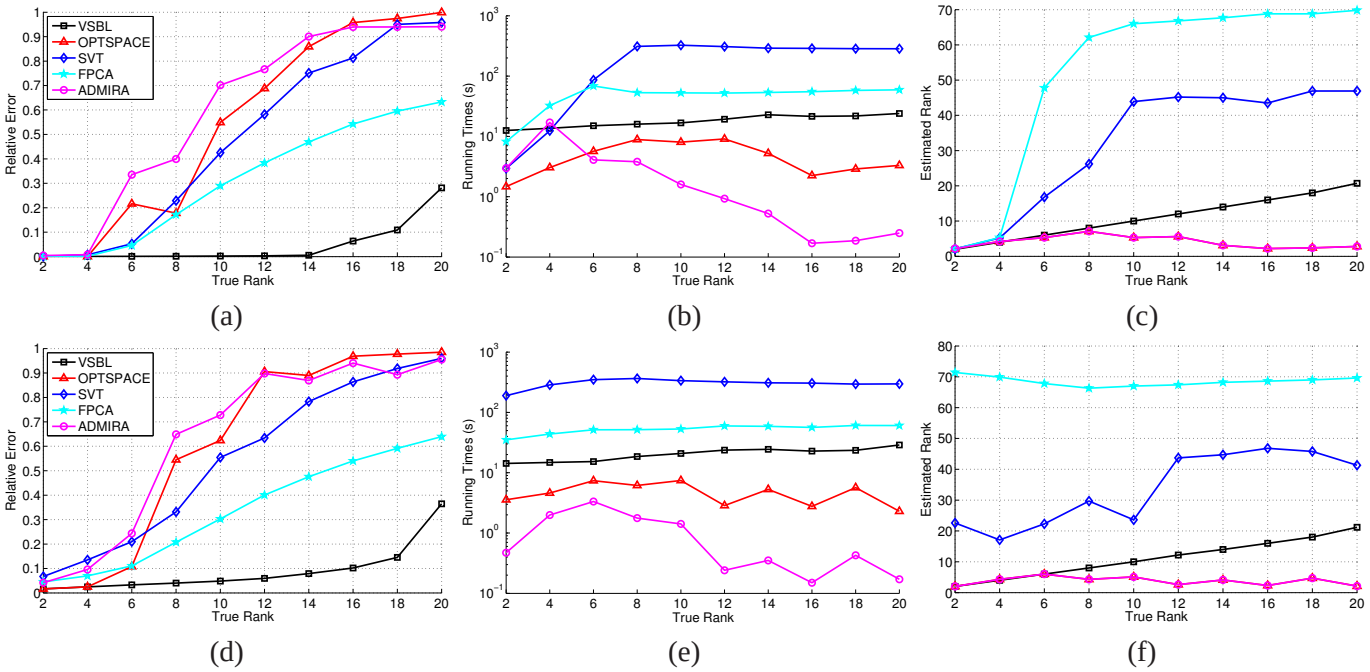


Fig. 1. Estimation results with matrices of size 200×200 with varying ranks when 20% of the entries are observed. *Top:* No observation noise, *Bottom:* with observation noise. (a,d) Relative recovery error, (b,e) running times, and (c,f) estimated ranks.

V. Empirical Results

In this section, we provide experimental results for the matrix completion and robust PCA problems with both synthetically generated and real data sets. To examine the empirical performance of the proposed method, we performed experiments commonly used in the literature and compared the proposed methods to some existing algorithms.

A. Matrix Completion

Our first example illustrates the effectiveness of the proposed approach on determining the correct rank. We generated test matrices $\mathbf{X}$ of size 200×200 of ranks $r = 2, \ldots, 20$ by randomly sampling $200 \times r$ matrices $\mathbf{A}$ and $\mathbf{B}$ from a standard normal distribution $\mathcal{N}(0, 1)$ and setting $\mathbf{X} = \mathbf{A}\mathbf{B}^T$. The fraction of observed entries p is 0.2, and they are sampled uniformly at random. For each experiment, the relative recovery error is measured as $\| \hat{\mathbf{X}} - \mathbf{X} \|_{\mathrm{F}} \,/\, \| \mathbf{X} \|_{\mathrm{F}}$, where $\hat{\mathbf{X}}$ is the estimate of $\mathbf{X}$.

We present comparisons with the following algorithms: OPTSPACE [22], SVT [25], FPCA [27] and ADMIRA [28]. Our method, developed in Section III-A, is denoted by VSBL. We used the procedure proposed in [22] to estimate the initial target rank required by ADMIRA and OPTSPACE. On the other hand, other methods automatically estimate the rank of the unknown matrix. We consider two test cases, one with noiseless observations, and one where observed entries are corrupted by zero-mean white Gaussian noise with standard deviation 0.05. Each simulation result is obtained by averaging 10 random instances. Figure 1 shows the relative reconstruction error, running times (on a 3GHz Core2 Duo CPU) and estimated ranks for each algorithm for both test cases. Among all algorithms, VSBL provides the highest recovery performance for all ranks, and also estimates the correct rank in each case. As expected, errors in both the recovery and the estimated rank increase as the original rank increases. OPTSPACE and ADMIRA consistently underestimate the rank, whereas FPCA and SVT overestimate it. A similar behavior is observed in the presence of observation noise: although the recovery performance of all algorithms decreases, VSBL still exhibits a better ability to recover the original matrix and the correct rank than other methods.

We next consider another set of experimental conditions where 200×200 matrices of fixed rank of 5 are generated, and the number of observed entries is varied according to different oversampling degrees of freedom. Note that a matrix of size $m \times n$ of rank r depends upon $\mathrm{df} = r(m + n - r)$ degrees of freedom, and the oversampling degrees of freedom (osdf) is defined as pmn/df [23]. Experimental results for $\mathrm{osdf} = 2, 3, \ldots, 10$

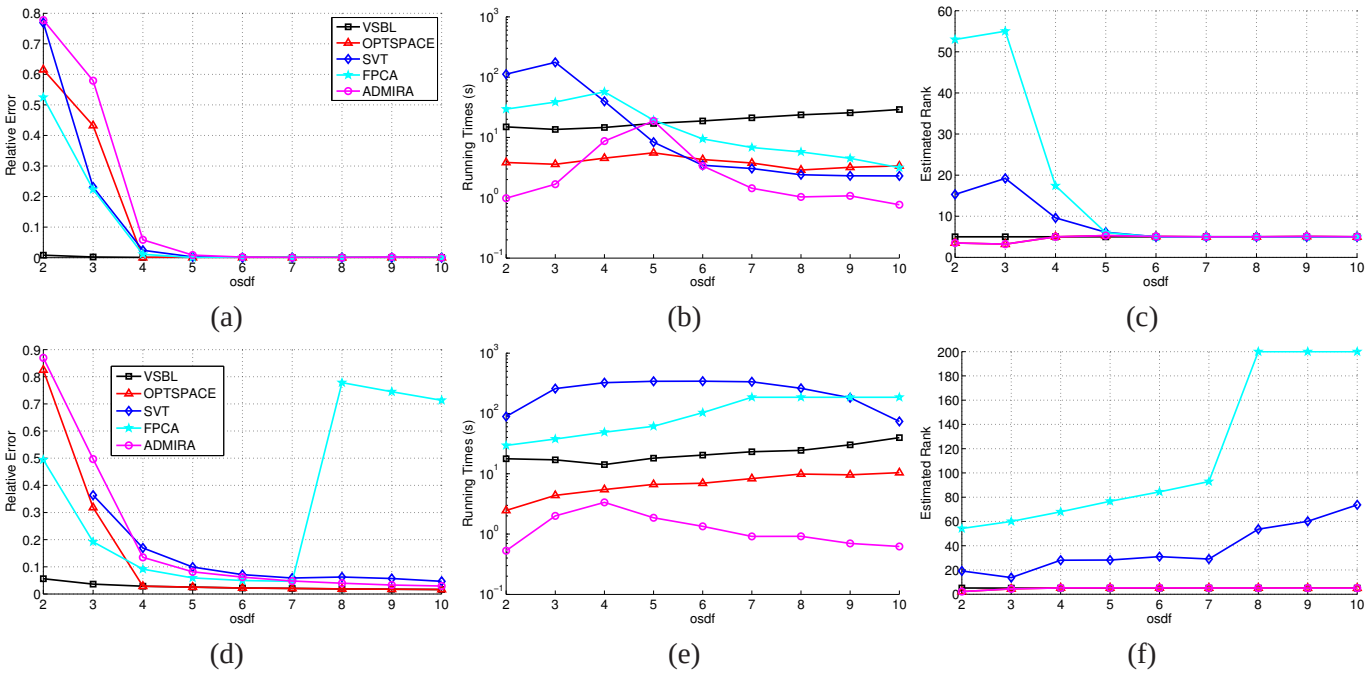


Fig. 2. Estimation results with matrices of size 200×200 of rank 5 with varying oversampling degrees of freedom. *Top:* No observation noise, *Bottom:* with observation noise. (a,d) Relative recovery error, (b,e) running times, and (c,f) estimated ranks. Error rates of SVT for $\mathrm{osdf} = 2$ and of FPC for $\mathrm{osdf} = 8, 9, 10$ with noisy observations are very high due to convergence failures.

TABLE I
NMAE VALUES ON THE JESTER JOKE DATA SET

	p = 0.1			p = 0.5		
# of users	100	300	1000	100	300	1000
VSBL	0.1625	0.1594	0.1584	0.1720	0.1669	0.1626
ADMIRA	0.1698	0.1705	0.1618	0.1775	0.1737	0.1710
OPT	0.1685	0.1700	0.1610	0.1744	0.1715	0.1694
SVT	0.1804	0.1682	0.1621	0.1943	0.1824	0.1743
FPC	0.2026	0.2046	0.2052	0.2096	0.2060	0.2051

are depicted in Figure 2 for the same noise conditions as above. The corresponding sampling ratios are $p \approx 0.1, 0.14, 0.20, 0.24, 0.30, 0.34, 0.40, 0.44, 0.50$. It is clear that VSBL provides very accurate reconstructions even with very low number of observations, for which other algorithms fail to provide meaningful results. In terms of computation time, ADMIRA provided the best performance in most of the simulations, whereas execution times for VSBL were stable throughout the testing conditions and were comparable to those of the other methods.

Our final example illustrates a real-world application of low-rank matrix completion methods. We generated a full rating matrix from the Jester joke data set[2] by removing all users containing missing entries, and applied the algorithms to randomly generated subsets of this matrix with different number of users and fraction of observed ratings p. The number of jokes is fixed to 100. As the performance measure we use the normalized mean absolute error (NMAE), which, for this dataset, is defined as $\frac{\sum_{(i,j)\in T} |X_{ij} - \hat{X}_{ij}|}{20\,|T|}$ [27], with $\hat{X}_{ij}$ the estimated missing components, T the set of missing entries, and $|T| = p$. It is known that as with most of the real-data sets, Jester data set is not low rank or even approximately low rank. To account for this in the proposed algorithm, β^{-1} is set equal to a high value (≈ 20) to encourage shrinkage. Numerical results (average of 10 realizations) are shown in Table I for two p values and three different number of users. It can be observed that VSBL achieves a better prediction error than other algorithms in all test cases.

[2]Available at http://eigentaste.berkeley.edu/

B. Robust PCA

In our first experiment, we demonstrate the performance of the proposed method using synthetic data in comparison with existing approaches. The low-rank component $\mathbf{X}$ is generated as in Section V-A. The non-zero entries of the sparse matrix $\mathbf{E} \in \mathbb{R}^{m \times n}$ are located uniformly at random and are drawn from a uniform distribution in the range $[-10, 10]$. We consider both a noise-free and a noisy case where white Gaussian noise with variance 10^{-3} is added to the original data. As before, the relative recovery error is measured as $\| \hat{\mathbf{X}} - \mathbf{X} \|_{\mathrm{F}} \; / \; \| \mathbf{X} \|_{\mathrm{F}}$ and $\| \hat{\mathbf{E}} - \mathbf{E} \|_{\mathrm{F}} \; / \; \| \mathbf{E} \|_{\mathrm{F}}$, where $\hat{\mathbf{X}}$ and $\hat{\mathbf{E}}$ represent the estimates.

We present comparisons with the Bayesian MCMC method proposed in [33] (denoted by BRPCA) and the optimization-based method in [32] (denoted by ALM). We use the exact inference method in [32] to report its best performance in terms of recovery error. The proposed method, developed in Section III-B, is denoted as VBRPCA.

Table II shows the relative reconstruction error, running times (on a 3GHz Core2 Duo CPU) and estimated ranks/sparsity levels for each algorithm for both noiseless and noisy cases. The average of 10 random instances is reported in each experiment.

It is clear that all methods provide very good reconstructions with both noiseless and noisy observations; both the low-rank and sparse components are recovered with excellent accuracy in all test cases. The running times of ALM and VBRPCA are very similar; although ALM has a lower theoretical computational complexity, the proposed method has comparable (and even lower) running times due to its fast convergence. BRPCA, on the other hand, has a very high computational complexity and therefore has longer running times in all test cases.

Although ALM is a very attractive method (especially for large scale problems) due to its recovery performance and fast convergence, it does not provide means to estimate the dense noise level. Therefore, its convergence threshold should be adapted to the noise variance to achieve the optimal performance, which requires user supervision. The setting for the convergence threshold used for the noiseless case causes overfitting in the low-rank component and longer convergence times in the presence of dense noise, as illustrated in Table II. When the threshold is adjusted to handle the dense (i.e., full-rank) noise component, results similar to the noiseless case can be obtained (see [33] for a related discussion). However, ALM is very sensitive to this parameter, and it generally requires careful tuning for accurate estimation of the rank.

On the other hand, the proposed method VBRPCA automatically estimates the dense noise level and provides high accuracy results comparable to the noiseless case. BRPCA has a similar mechanism for automatic noise estimation through Bayesian formulation, but its results are generally inferior to the proposed method and its computational complexity is significantly higher.

Our second example illustrates a real-world application of robust PCA methods. We consider the foreground/background separation problem in video as in [33]. Each column of the data matrix $\mathbf{Y}$ is generated by concatenating pixels of one video frame into a vector. In this application, the low-rank component corresponds to the background of the scene, and the sparse component is used to model the moving objects in the foreground. It is clear that for a completely static background, the ideal estimate of the rank of the background is 1, but in the case of dynamic backgrounds (e.g., due to illumination changes), the rank can be higher.

All algorithms are applied to the video data[3] consisting of 158 frames of size 192×144. Example results obtained by the algorithms in one video frame are shown in Fig. 3. Due to the slow motion of the people, they can be incorporated by mistake into the low-rank component (i.e., the background), which is the case with the ALM algorithm. This is due to overfitting in the low-rank component, which was also observed in the synthetic experiments with the ALM method in the presence of dense noise. The BRPCA algorithm provides a better result, but parts of the foreground are mistakingly classified as background. The proposed algorithm results in a much cleaner separation, mainly due to the fact that a lower-rank estimate for the background is enforced compared to the other methods. This helps to avoid misclassification of foreground and background pixels. In this dataset, the running times of the algorithms were around 10 mins for ALM, 60 mins for BRPCA, and 11 mins for the proposed method.

VI. Conclusions

In this paper, we have applied sparse Bayesian learning principles to the low-rank matrix estimation in matrix completion and robust principal component analysis. We introduced a formulation where the low-rank constraint

[3]The data can be found in http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/.

TABLE II
RELATIVE RECONSTRUCTION ERRORS, ESTIMATED RANKS AND COMPUTATION TIMES FOR ROBUST PCA

Method	σ	m=n	rank($\mathbf{X}$)	$\parallel E \parallel_0$	rank($\hat{\mathbf{X}}$)	$\frac{\|\hat{\mathbf{X}}-\mathbf{X}\|_F}{\|\mathbf{X}\|_F}$	$\frac{\|\hat{\mathbf{E}}-\mathbf{E}\|_F}{\|\mathbf{E}\|_F}$	time(s)
ALM	0	200	5	400	5	2.1×10^{-8}	4.4×10^{-7}	1.16
ALM	0	200	10	400	10	2.0×10^{-8}	6.0×10^{-7}	1.58
ALM	0	400	20	1600	20	1.0×10^{-8}	4.4×10^{-7}	5.1
ALM	0	800	40	6400	40	2.2×10^{-8}	2.6×10^{-7}	50.04
BRPCA	0	200	5	400	4	1.5×10^{-5}	1.4×10^{-6}	22.80
BRPCA	0	200	10	400	10	5.2×10^{-4}	1.2×10^{-7}	240.40
BRPCA	0	400	20	1600	20	3.7×10^{-4}	9.1×10^{-7}	2276.33
BRPCA	0	800	40	6400	40	2.4×10^{-4}	1.6×10^{-7}	23340.66
VBRPCA	0	200	5	400	5	2.8×10^{-15}	6.1×10^{-15}	0.56
VBRPCA	0	200	10	400	10	4.7×10^{-15}	1.1×10^{-14}	0.68
VBRPCA	0	400	20	1600	20	3.3×10^{-15}	1.4×10^{-14}	4.56
VBRPCA	0	800	40	6400	40	4.2×10^{-15}	3.7×10^{-14}	40.78
ALM	10^{-3}	200	5	400	140	6.2×10^{-4}	1.4×10^{-3}	5.57
ALM	10^{-3}	200	10	400	140	3.6×10^{-4}	1.7×10^{-3}	5.97
ALM	10^{-3}	400	20	1600	276	1.8×10^{-4}	1.6×10^{-3}	39.00
ALM	10^{-3}	800	40	6400	549	0.9×10^{-4}	1.6×10^{-3}	287.00
BRPCA	10^{-3}	200	5	400	5	4.5×10^{-4}	1.0×10^{-3}	28.42
BRPCA	10^{-3}	200	10	400	10	4.4×10^{-4}	1.5×10^{-3}	240.10
BRPCA	10^{-3}	400	20	1600	20	1.1×10^{-3}	7.4×10^{-3}	2270.43
BRPCA	10^{-3}	800	40	6400	40	0.9×10^{-3}	3.7×10^{-3}	23166.05
VBRPCA	10^{-3}	200	5	400	5	2.8×10^{-4}	3.0×10^{-3}	0.56
VBRPCA	10^{-3}	200	10	400	10	2.6×10^{-4}	3.3×10^{-3}	0.73
VBRPCA	10^{-3}	400	20	1600	20	1.4×10^{-4}	3.2×10^{-3}	5.34
VBRPCA	10^{-3}	800	40	6400	40	0.7×10^{-4}	3.2×10^{-3}	46.52

is imposed on the estimate by using its sparse representation; starting from the factorized form of the unknown matrix, we enforce a common sparsity profile on its underlying components using a probabilistic formulation. We modeled the remaining unknown variables and observations in the hierarchical Bayesian framework and developed inference methods based on mean-field variational Bayes approximating the posteriors of interest. Empirical results suggest that the proposed algorithms are very effective in pruning irrelevant dimensions and recover the correct number of effective components in the matrix estimate, and they provide competitive, and even higher, performance than current state-of-the-art approaches in terms of reconstruction performance.

Concluding, we believe that the formulation based on the sparsity concepts is a powerful approach for low-rank matrix estimation problems, and many advanced methods in the well-developed Bayesian sparse approximation field can be applied with ease for potentially extending the range of applications. These and further theoretical analysis remain as future research directions.

REFERENCES

[1] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. of Comput. Math.*, vol. 9, pp. 717–772, 2008.

[2] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Trans. Inform. Theory*, vol. 56, no. 5, pp. 2053–2080, 2009.

[3] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *CoRR*, vol. abs/0912.3599, 2009.

[4] Z. Zhou, X. Li, J. Wright, E. Cands, and Y. Ma, "Stable principal component pursuit," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2010.

[5] C. R. Johnson, "Matrix completion problems: a survey," in *Proceed. of Symposia in Applied Mathematics*, vol. 40, 1990, pp. 171–198.

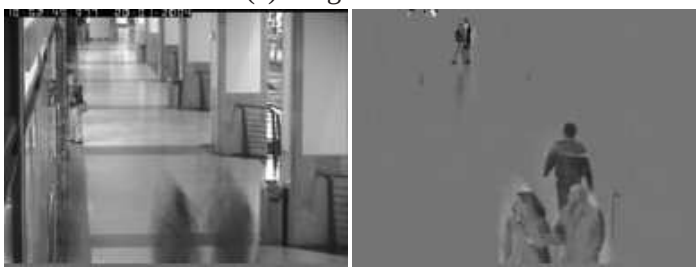
[6] Z. Liu and L. Vandenberghe, "Interior-point method for nuclear norm approximation with application to system identification," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, pp. 1235–1256, 2009.

[7] S. Oh, A. Karbasi, and A. Montanari, "Sensor Network Localization from Local Connectivity : Performance Analysis for the MDS-MAP Algorithm," in *IEEE Information Theory Workshop (ITW 2010)*, 2010.

[8] N. Srebro, "Learning with matrix factorizations," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 2004.

[9] P. Chen and D. Suter, "Recovering the missing components in a large noisy low-rank matrix: Application to SFM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, pp. 1051–1063, 2004.

(a) Original Frame



(b) ALM

(c) BRPCA

(d) VBRPCA

Fig. 3. Video background/foreground separation. (a) Original video frame, the reconstructions by (b) ALM, (c) BRPCA, and (d) VBRPCA. *Left:* background reconstruction, *right:* foreground reconstruction.

[10] H. Ji, C. Liu, Z. Shen, and Y. Xu, "Robust video denoising using low rank matrix completion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, 2010.

[11] Z.-P. Liang, "Spatiotemporal imaging with partially separable functions," in *NFSI-ICFBI 2007*, 2007, pp. 181 –182.

[12] P. J. Huber and E. M. Ronchetti, *Robust Statistics*. Wiley, 2009.

[13] F. D. la Torre and M. J. Black, "A framework for robust subspace learning," *International Journal of Computer Vision*, vol. 54, pp. 117–142, 2003.

[14] Q. Ke and T. Kanade, "Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2005.

[15] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381–385, 1981.

[16] J. Gao, "Robust l1 principal component analysis and its Bayesian variational inference," *Neural Computation*, vol. 20, pp. 555–578, 2008.

[17] J. Luttinen, A. Ilin, and J. Karhunen, "Bayesian robust PCA for incomplete data," in *International Conference on Independent Component Analysis and Signal Separation*, 2009.

[18] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, and Y. Ma, "Towards a practical face recognition system: Robust alignment and illumination by sparse representation," *submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, July 2010.

[19] C. Papadimitriou, P. Rghavan, H. Tamaki, and S. Vempala, "Latent semantic indexing, a probabilistic analysis," *Journal of Computer and System Sciences*, vol. 61, pp. 217–235, 2000.

[20] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, July 2010.

[21] 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.

[22] R. H. Keshavan, S. Oh, and A. Montanari, "Matrix completion from a few entries," *submitted to IEEE Trans. Inf. Theory, arXiv:0901.3150v2*, 2009.

[23] E. J. Candès and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925 – 936, 2009.

[24] V. Chandrasekharan, S. Sanghavi, P. Parillo, and A. Wilsky, "Rank-sparsity incoherence for matrix decomposition." *preprint*, 2009.

[25] J.-F. Cai, E. J. Candes, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2008.

[26] R. Meka, P. Jain, and I. S. Dhillon, "Guaranteed rank minimization via singular value projection," *arXiv:0909.5457*, 2009.

[27] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and Bregman iterative methods for matrix rank minimization," *arXiv:0905.1643v2*, 2009.

[28] K. Lee and Y. Bresler, "ADMiRA: Atomic decomposition for minimum rank approximation," *arXiv:0905.0044*, 2009.

[29] J. Paisley and L. Carin, "A nonparametric Bayesian model for kernel matrix completion," in *ICASSP 2010*, Dallas, USA, April 2010.

[30] Y. J. Lim and Y. W. Teh, "Variational Bayesian approach to movie rating prediction," in *Proceedings of KDD Cup and Workshop*, 2007.

[31] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," *SIAM J. Optimization (submitted)*, 2009.

[32] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," University of Illinois at Urbana-Champaign, Tech. Rep., 2010.

[33] X. Ding, L. He, and L. Carin, "Bayesian robust principal component analysis," *submitted to IEEE Trans. Image Processing*, 2010.

[34] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, pp. 211–244, 2001.

[35] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *NIPS 13*, 2001, pp. 556–562.

[36] J. Haldar and D. Hernando, "Rank-constrained solutions to linear matrix equations using power factorization," *IEEE Signal Process. Lett.*, vol. 16, no. 7, pp. 584–587, 2009.

[37] D. Wipf, J. Palmer, and B. D. Rao, "Perspectives on sparse Bayesian learning," *NIPS 16*, 2004.

[38] M. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, The Gatsby Computational Neuroscience Unit, University College London, 2003.

[39] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[40] D. Wipf and B. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3704 –3716, july 2007.

[41] M. E. Tipping, "Sparse kernel principal component analysis," in *NIPS*, 2000, pp. 633–639.

[42] C. M. Bishop, "Bayesian PCA," in *NIPS*, 1999, pp. 382–388.

[43] ——, "Variational principal components," in *Proceedings Ninth International Conference on Artificial Neural Networks, ICANN'99*, 1999, pp. 509–514.

[44] V. Y. F. Tan and C. Fvotte, "Automatic relevance determination in nonnegative matrix factorization," in *Proc. Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS'09)*, 2009.

[45] K. Mardia, J. Kent, and J. Bibby, *Multivariate analysis*. Academic Press; New York, 1979.