Sparse Bayesian Methods for Low-Rank Matrix Estimation

S. Derin Babacan, *Member, IEEE*, Martin Luessi, *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. 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

Recently, there has been 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 Radiology, Martinos Center for Biomedical Imaging, Massachusetts General Hospital, Harvard Medical School, Boston, MA, USA e-mail: mluessi@nmr.mgh.harvard.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

This work has been supported in part by the Beckman Institute postdoctoral fellowship, ”Ministerio de Ciencia e Innovacion” under contract TIN2010-15137, the Spanish research program Consolider Ingenio 2010: MIPRCV (CSD2007-00018), and a grant from the Department of Energy (DE-NA0000457).

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 non-zero with (arbitrarily) large magnitudes.

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

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

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 $\mathbf{X}$. Under some conditions the solutions of these two problems coincide and recovery guarantees exist (see, for example, [1], [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 [29]–[34]: a nonparametric approach for symmetric positive definite matrices is proposed in [29], and a variational Bayes method is developed for collaborative filtering in [31]. The method in [32] is based on beta-Bernoulli processes for modeling and Gibbs sampling for inference.

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.* [35] proposed accelerated proximal gradient (APG) methods which are faster and generally more accurate. The augmented Lagrange Multiplier Method (ALM) [36] 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 [37] addresses this issue by simultaneously estimating the necessary parameters along with the unknowns, but the resulting algorithm uses sampling for inference and has high computational complexity.

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 deterministic 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 [38]. This property is not present in most of the existing deterministic 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-

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

and application-dependent supervision. Finally, empirical results demonstrate that the proposed methods provide very good reconstruction performance compared to existing methods while accurately estimating the unknown effective rank.

This work is closely related to some probabilistic formulations in collaborative filtering [31], [39] and nonnegative matrix factorization [40], regarding the modeling of the unknown low-rank components. The works [31], [39] are variational Bayesian approaches to matrix completion, where the low-rank matrix is modeled via two factors and two sets of independent hyperparameters. On the other hand, [40] proposed to relate these factors by a single set of hyperparameters, and a maximum *a posteriori* approach is used for inference. One of the main contributions of this work is to combine the modeling approaches in these works to obtain an intuitive modeling structure, and to provide a variational algorithm using this modeling which renders heuristic measures unnecessary and enables fully-automated estimation without free parameters. Another contribution of this work is to extend the application of this low-rank modeling to the robust PCA problem by including additional hierarchical modeling for the sparse errors with arbitrarily large coefficients. The relation with prior art is discussed in more detail later in this paper (Section IV-A).

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 $\operatorname{rank}(\mathbf{X}) = r \leq \min(m,n)$. 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 $m \times r$ and $n \times r$ 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 [41] and matrix completion [42], which generally aim to find solutions to

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

The equivalence of this optimization problem to (3) can be shown (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 precisions (inverse variances) γ_i, that is,

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

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

where $\mathbf{I}_m$ denotes the $m \times m$ identity matrix. Thus, the columns of $\mathbf{A}$ and $\mathbf{B}$ have the same sparsity profile enforced by the common precisions γ_i. As shown later, many of the precisions γ_i will assume very large 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) [38], [43] successfully utilized for compressive sensing reconstruction, where sparsity-inducing Gaussian priors are employed on each of the coefficients of the unknown vector.

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

$$\mathrm{p}(\gamma_i) = \mathrm{Gamma}(a, \frac{1}{b}) \propto \gamma_i^{a-1} \exp\left(-b\,\gamma_i\right) \,. \tag{10}$$

The parameters a and b are treated as deterministic whose values are set to small values (e.g., 10^{-5}) to obtain broad hyperpriors.

B. Observation and Noise Models

In this work, the prior structure in (8), (9) and (10) 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

$$Y_{ij} = X_{ij} + N_{ij}, \quad (i,j) \in \Omega, \tag{11}$$

or in a more compact form as

$$\mathbf{Y} = \mathcal{P}_{\Omega}\left(\mathbf{X} + \mathbf{N}\right), \tag{12}$$

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

$$\mathrm{p}(\mathbf{Y}|\mathbf{A},\mathbf{B},\beta) = \prod_{(i,j)\in\Omega} \mathcal{N}\left(Y_{ij}|X_{ij}, \beta^{-1}\right) \,, \tag{13}$$

with $\beta = 1/\epsilon$ the noise precision. The noise precision β is assigned the noninformative Jeffrey's prior

$$\mathrm{p}(\beta) = \beta^{-1}. \tag{14}$$

The joint distribution, therefore, is expressed as

$$\mathrm{p}(\mathbf{Y},\mathbf{A},\mathbf{B},\boldsymbol{\gamma},\beta) = \mathrm{p}(\mathbf{Y}|\mathbf{A},\mathbf{B},\beta)\,\mathrm{p}(\mathbf{A}|\boldsymbol{\gamma})\,\mathrm{p}(\mathbf{B}|\boldsymbol{\gamma})\mathrm{p}(\boldsymbol{\gamma})\,\mathrm{p}(\beta)\,. \tag{15}$$

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

$$\begin{aligned} \mathrm{p}(\mathbf{Y}|\mathbf{A},\mathbf{B},\mathbf{E},\beta) &= \mathcal{N}\left(\mathbf{Y}|\mathbf{A}\mathbf{B}^T + \mathbf{E}, \beta^{-1}\mathbf{I}_{mn}\right) \\ &\propto \exp\left(\frac{\beta}{2} \parallel \mathbf{Y} - \mathbf{A}\mathbf{B}^T - \mathbf{E} \parallel_{\mathrm{F}}^2\right) . \end{aligned} \tag{16}$$

As in the matrix completion case, we assign the Jeffrey's prior in (14) 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,

$$\mathrm{p}(\mathbf{E}|\boldsymbol{\alpha}) = \prod_{i=1}^{m}\prod_{j=1}^{n} \mathcal{N}\left(E_{ij}|0, \alpha_{ij}^{-1}\right) , \tag{17}$$

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 Jeffrey's priors on α_{ij}

$$\mathrm{p}(\alpha_{ij}) = \alpha_{ij}^{-1}, \ \ \forall i, j \, . \tag{18}$$

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

$$\begin{aligned} \mathrm{p}(\mathbf{Y},\mathbf{A},\mathbf{B},\mathbf{E},\boldsymbol{\gamma},\boldsymbol{\alpha},\beta) &= \mathrm{p}(\mathbf{Y}|\mathbf{A},\mathbf{B},\mathbf{E},\beta)\,\mathrm{p}(\mathbf{A}|\boldsymbol{\gamma})\,\mathrm{p}(\mathbf{B}|\boldsymbol{\gamma}) \\ &\quad \times \mathrm{p}(\mathbf{E}|\boldsymbol{\alpha})\mathrm{p}(\boldsymbol{\gamma})\,\mathrm{p}(\boldsymbol{\alpha})\,\mathrm{p}(\beta). \end{aligned} \tag{19}$$

III. Approximate Bayesian Inference

As is widely known, exact full-Bayesian inference using joint distributions such as (15) and (19) is intractable, since $\mathrm{p}(\mathbf{y})$ cannot be computed by marginalizing all latent variables. Therefore, approximation methods must be utilized. Common approximations include maximum *a posteriori* (MAP) estimation, evidence-based analysis and variational Bayes. Although all of these methods can only provide local minima, Bayesian inference (where at least one variable is integrated out) is generally more effective in avoiding undesired local minima compared to deterministic methods such as MAP. This is mainly due to the fact that Bayesian methods approximate the full posterior distributions instead of providing point-estimates of its modes. Although in theory sampling methods can provide the optimal approximation to the posteriors, the computational complexity is significantly higher for high-dimensional data than that of other methods, and convergence is generally hard to assess.

In this work, we present an inference procedure based on mean field variational Bayes [44], [45]. 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

$$\log \mathrm{q}(\mathbf{z}_k) = \langle \log \mathrm{p}(\mathbf{Y},\mathbf{z}) \rangle_{\mathbf{z}\backslash\mathbf{z}_k} + \mathrm{const}, \tag{20}$$

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 (15) for the matrix completion problem, and in (19) 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 (20) 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 (13), the posterior density of the i^{th} row $\mathbf{a}_{i\cdot}$ of $\mathbf{A}$ is found as

$$\mathrm{q}(\mathbf{a}_{i\cdot}) = \mathcal{N}\left(\mathbf{a}_{i\cdot}|\langle\mathbf{a}_{i\cdot}\rangle, \mathbf{\Sigma}_i^a\right) \;, \tag{21}$$

with mean and covariance

$$\langle\mathbf{a}_{i\cdot}\rangle^T = \langle\beta\rangle\,\mathbf{\Sigma}_i^a\,\langle\mathbf{B}_i\rangle^T\;\mathbf{y}_{i\cdot}{}^T\,, \tag{22}$$

$$\mathbf{\Sigma}_i^a = \left(\langle\beta\rangle\,\langle\mathbf{B}_i^T\mathbf{B}_i\rangle + \mathbf{\Gamma}\right)^{-1}\;, \tag{23}$$

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

$$\langle\mathbf{B}_i^T\mathbf{B}_i\rangle = \sum_{j:(i,j)\in\Omega}\langle\mathbf{b}_{j\cdot}{}^T\,\mathbf{b}_{j\cdot}\rangle = \sum_{j:(i,j)\in\Omega}\left(\langle\mathbf{b}_{j\cdot}{}^T\rangle\langle\mathbf{b}_{j\cdot}\rangle + \mathbf{\Sigma}_j^b\right)\;, \tag{24}$$

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 (13), the posterior density of the j^{th} row $\mathbf{b}_{j\cdot}$ of $\mathbf{B}$ is found as a normal distribution

$$\mathrm{q}(\mathbf{b}_{j\cdot}) = \mathcal{N}\left(\mathbf{b}_{j\cdot}|\langle\mathbf{b}_{j\cdot}\rangle, \mathbf{\Sigma}_j^b\right) \tag{25}$$

with parameters

$$\langle\mathbf{b}_{j\cdot}\rangle^T = \langle\beta\rangle\,\mathbf{\Sigma}_j^b\,\langle\mathbf{A}_j\rangle^T\,\mathbf{y}_{\cdot j}\,, \tag{26}$$

$$\mathbf{\Sigma}_j^b = \left(\langle\beta\rangle\,\langle\mathbf{A}_j^T\mathbf{A}_j\rangle + \mathbf{\Gamma}\right)^{-1}\;, \tag{27}$$

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 covariances $\mathbf{\Sigma}_i^b$ of the estimate of $\mathbf{B}$ are incorporated in the estimation of $\mathbf{A}$ (and vice versa).

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 a Gamma distribution

$$\mathrm{q}(\gamma_i) \propto \gamma_i^{\left(a-1+\frac{m+n}{2}\right)}\exp\left(-\gamma_i\,\frac{2b+\langle\mathbf{a}_{\cdot i}{}^T\mathbf{a}_{\cdot i}\rangle + \langle\mathbf{b}_{\cdot i}{}^T\mathbf{b}_{\cdot i}\rangle}{2}\right) \tag{28}$$

with mean

$$\langle\gamma_i\rangle = \frac{2a+m+n}{2b+\langle\mathbf{a}_{\cdot i}{}^T\mathbf{a}_{\cdot i}\rangle + \langle\mathbf{b}_{\cdot i}{}^T\mathbf{b}_{\cdot i}\rangle}\,. \tag{29}$$

The required expectations are given by

$$\langle\mathbf{a}_{\cdot i}{}^T\mathbf{a}_{\cdot i}\rangle = \langle\mathbf{a}_{\cdot i}\rangle^T\langle\mathbf{a}_{\cdot i}\rangle + \sum_j\left(\mathbf{\Sigma}_j^a\right)_{ii}\,, \tag{30}$$

$$\langle\mathbf{b}_{\cdot i}{}^T\mathbf{b}_{\cdot i}\rangle = \langle\mathbf{b}_{\cdot i}\rangle^T\langle\mathbf{b}_{\cdot i}\rangle + \sum_j\left(\mathbf{\Sigma}_j^b\right)_{ii}\;. \tag{31}$$

3) Estimation of noise precision β*:* The Bayesian methodology allows for the estimation of the noise precision as well. From (20), the posterior approximation assumes a Gamma distribution with mean

$$\langle\beta\rangle = \frac{pmn}{\langle\;\|\,\mathbf{Y}-\mathcal{P}_\Omega\left(\mathbf{A}\mathbf{B}^T\right)\|_{\mathrm{F}}^2\;\rangle}\,. \tag{32}$$

In summary, the algorithm proceeds by first estimating the rows of $\mathbf{A}$ and $\mathbf{B}$ using (22) and (26), respectively, followed by the estimation of the precisions γ_i using (29), and (if desired) the noise precision β using (32). By the properties of the variational Bayes methods, the algorithm is guaranteed to converge to a local minimum of the variational bound [45].

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 (21) and (25) with the same factorization over the rows of $\mathbf{A}$ and $\mathbf{B}$, respectively. However, as opposed to the matrix completion case, the covariances $\mathbf{\Sigma}_i^a$ of the rows of $\mathbf{A}$ are equal since there are no missing values (the same applies to $\mathbf{B}$). The posterior approximation of the i^{th} row of $\mathbf{A}$ is given by

$$\mathrm{q}(\mathbf{a}_{i\cdot}) = \mathcal{N}\left(\mathbf{a}_{i\cdot} | \langle \mathbf{a}_{i\cdot} \rangle, \mathbf{\Sigma}^A\right) \ , \tag{33}$$

with mean and covariance

$$\langle \mathbf{a}_{i\cdot} \rangle^T = \langle \beta \rangle \, \mathbf{\Sigma}^A \, \langle \mathbf{B} \rangle^T \ \left(\mathbf{y}_{i\cdot} - \mathbf{e}_{i\cdot}\right)^T \ , \tag{34}$$

$$\mathbf{\Sigma}^A = \left(\langle \beta \rangle \, \langle \mathbf{B}^T \mathbf{B} \rangle + \mathbf{\Gamma}\right)^{-1} \ . \tag{35}$$

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

$$\mathrm{q}(\mathbf{b}_{j\cdot}) = \mathcal{N}\left(\mathbf{b}_{j\cdot} | \langle \mathbf{b}_{j\cdot} \rangle, \mathbf{\Sigma}^B\right) \tag{36}$$

with parameters

$$\langle \mathbf{b}_{j\cdot} \rangle^T = \langle \beta \rangle \, \mathbf{\Sigma}^B \, \langle \mathbf{A} \rangle^T \ \left(\mathbf{y}_{\cdot j} - \mathbf{e}_{\cdot j}\right) \ , \tag{37}$$

$$\mathbf{\Sigma}^B = \left(\langle \beta \rangle \, \langle \mathbf{A}^T \mathbf{A} \rangle + \mathbf{\Gamma}\right)^{-1} \ . \tag{38}$$

The required expectations can be found as

$$\langle \mathbf{A}^T \mathbf{A} \rangle = \langle \mathbf{A} \rangle^T \langle \mathbf{A} \rangle + m \mathbf{\Sigma}^A, \tag{39}$$

$$\langle \mathbf{B}^T \mathbf{B} \rangle = \langle \mathbf{B} \rangle^T \langle \mathbf{B} \rangle + n \mathbf{\Sigma}^B \, . \tag{40}$$

Using these updates, the estimate of $\mathbf{X}$ is then found by $\mathbf{X} = \langle \mathbf{A} \rangle \langle \mathbf{B} \rangle^T$.

2) Estimation of $\mathbf{E}$*:* Using (20), the posterior distribution approximation of $\mathbf{E}$ is found to be factorized on each coefficient E_{ij} with distributions

$$\mathrm{q}(E_{ij}) = \mathcal{N}\left(E_{ij} | \langle E_{ij} \rangle, \Sigma_{ij}^E\right) \ , \tag{41}$$

with parameters

$$\langle E_{ij} \rangle = \langle \beta \rangle \ \Sigma_{ij}^E \ \left(Y_{ij} - \langle \mathbf{a}_{i\cdot} \rangle \langle \mathbf{b}_{j\cdot} \rangle^T\right) \ , \tag{42}$$

$$\Sigma_{ij}^E = \frac{1}{\langle \beta \rangle + \langle \alpha_{ij} \rangle} \, . \tag{43}$$

Notice that (42) can be rewritten as

$$\langle E_{ij} \rangle = \frac{\langle \beta \rangle}{\langle \beta \rangle + \langle \alpha_{ij} \rangle} \ \left(Y_{ij} - \langle \mathbf{a}_{i\cdot} \rangle \langle \mathbf{b}_{j\cdot} \rangle^T\right) \ , \tag{44}$$

where the first term is at most 1, and hence this estimation of E_{ij} corresponds to a shrinkage of the difference between the observations and the low-rank estimates controlled by the noise precision β and the hyperparameters α_{ij}. When a specific hyperparameter goes to infinity, that is, $\langle \alpha_{ij} \rangle^{-1} \to 0$, the mean and variance of the corresponding coefficient E_{ij} become zero, resulting in sparse estimates of $\mathbf{E}$.

3) Estimation of hyperparameters $\boldsymbol{\gamma}$*:* Similarly to the above, the posterior density of $\boldsymbol{\gamma}$ is found as a Gamma distribution with mean given in (29). The only difference is in the calculation of the expectations, which are given by

$$\langle \mathbf{a}_{\cdot i}{}^T \mathbf{a}_{\cdot i} \rangle = \langle \mathbf{a}_{\cdot i} \rangle^T \langle \mathbf{a}_{\cdot i} \rangle + m \left(\mathbf{\Sigma}^A\right)_{ii} , \tag{45}$$

$$\langle \mathbf{b}_{\cdot i}{}^T \mathbf{b}_{\cdot i} \rangle = \langle \mathbf{b}_{\cdot i} \rangle^T \langle \mathbf{b}_{\cdot i} \rangle + n \left(\mathbf{\Sigma}^B\right)_{ii} \, . \tag{46}$$

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

$$\langle \alpha_{ij} \rangle = \frac{1}{\langle E_{ij} \rangle^2 + \Sigma_{ij}^E} \, . \tag{47}$$

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

$$\langle \beta \rangle = \frac{mn}{\langle \parallel \mathbf{Y} - \mathbf{A}\mathbf{B}^T - \mathbf{E} \parallel_{\mathrm{F}}^2 \rangle}\,, \tag{48}$$

where

$$\begin{aligned} &\langle \parallel \mathbf{Y} - \mathbf{A}\mathbf{B}^T - \mathbf{E} \parallel_{\mathrm{F}}^2 \rangle = \parallel \mathbf{Y} - \langle \mathbf{A} \rangle \langle \mathbf{B} \rangle^T - \langle \mathbf{E} \rangle \parallel_{\mathrm{F}}^2 \\ &+ \mathrm{Tr}\left(n \, \langle \mathbf{A} \rangle^T \langle \mathbf{A} \rangle \boldsymbol{\Sigma}^B \right) + \mathrm{Tr}\left(m \, \langle \mathbf{B} \rangle^T \langle \mathbf{B} \rangle \boldsymbol{\Sigma}^A \right) \\ &+ \mathrm{Tr}\left(mn \, \boldsymbol{\Sigma}^A \boldsymbol{\Sigma}^B \right) + \sum_{i=1}^{m} \sum_{j=1}^{n} \Sigma_{ij}^E \,. \end{aligned} \tag{49}$$

In summary, the proposed algorithm estimates the low rank component $\mathbf{X}$ by estimating its factors using (34) and (37), followed by the estimation of the sparse matrix $\mathbf{E}$ using (42), and finally the estimation of all hyperparameters using (29), (47) and (48), until convergence.

IV. Discussion

A. Related Prior Art

The methodology presented in this work is closely related to some methods developed for collaborative filtering, probabilistic principal component analysis (PCA) and (nonnegative) matrix factorization. In collaborative filtering methods proposed in [31], [39], 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 these models are similar to our approach, the columns of $\mathbf{A}$ and $\mathbf{B}$ are not coupled through the use of common precisions 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 precisions 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 precisions. To cope with scalability issues, [31] uses fixed, heuristically selected values for one set, and estimates the other hyperparameter set. Finally, in contrast to our work, [31] has reported that no sparsity in the precisions occurs during the application of their algorithm.

The idea of coupling the columns of $\mathbf{A}$ and $\mathbf{B}$ is also used in [40], which aims at solving the nonnegative matrix factorization problem. 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. Note also that this method has 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 [16], [17], [46] use 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.

The Bayesian PCA methods [47]–[49] 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 (or vice versa). Although a similar approach can be developed in our formulation, i.e., marginalize one matrix factor to estimate the other, estimation of the common precisions γ_i becomes problematic since $\mathbf{A}$ and $\mathbf{B}$ cannot be integrated out together from the joint distribution.

Finally, another Bayesian modeling and inference strategy is proposed in [37] for the robust PCA. The work uses four distinct factors for the low-rank component, two of which are modeled using Gaussian priors similar to this work, and the remaining two is used to model the sparse eigenvalues of the low-rank matrix. Sparseness is explicitly imposed using a beta-Bernoulli hierarchical prior such that irrelevant components can be removed. This is in contrast to our work and the approaches presented above, where the model does not lead to exact pruning, but rather to a "soft" pruning (by driving components to values numerically indistinguishable from machine precision). The sparse component is modeled in a similar fashion in [37] with a combination of a beta-Bernoulli and normal-Gamma prior hierarchies. Due to the complex modeling, the posterior distributions can only be inferred using sampling strategies.

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 very large values, which will force the posterior means of the columns to go to zero, effectively removing them from the model and reducing the rank. In our implementation, columns of $\mathbf{A}$ and $\mathbf{B}$ were declared irrelevant at convergence if the corresponding γ_i^{-1} assumes a very small value (e.g., 10^{-16}).

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

As discussed in Section III-B2, the update procedure (42) 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 [38]. During the iterative procedure, many of the estimated precisions α_{ij} will approach very high values, which makes the corresponding posteriors in (41) 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 (42). 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 $\mathbf{E}$. In addition, we can find another update rule from (47) as

$$\langle \alpha_{ij} \rangle \left(\langle E_{ij} \rangle^2 + \Sigma_{ij}^E \right) = 1 \tag{50}$$

$$\langle \alpha_{ij} \rangle^{\text{new}} \langle E_{ij} \rangle^2 + \langle \alpha_{ij} \rangle^{\text{old}} \Sigma_{ij}^E = 1 \tag{51}$$

$$\langle \alpha_{ij} \rangle^{\text{new}} = \frac{1 - \langle \alpha_{ij} \rangle^{\text{old}} \Sigma_{ij}^E}{\langle E_{ij} \rangle^2}, \tag{52}$$

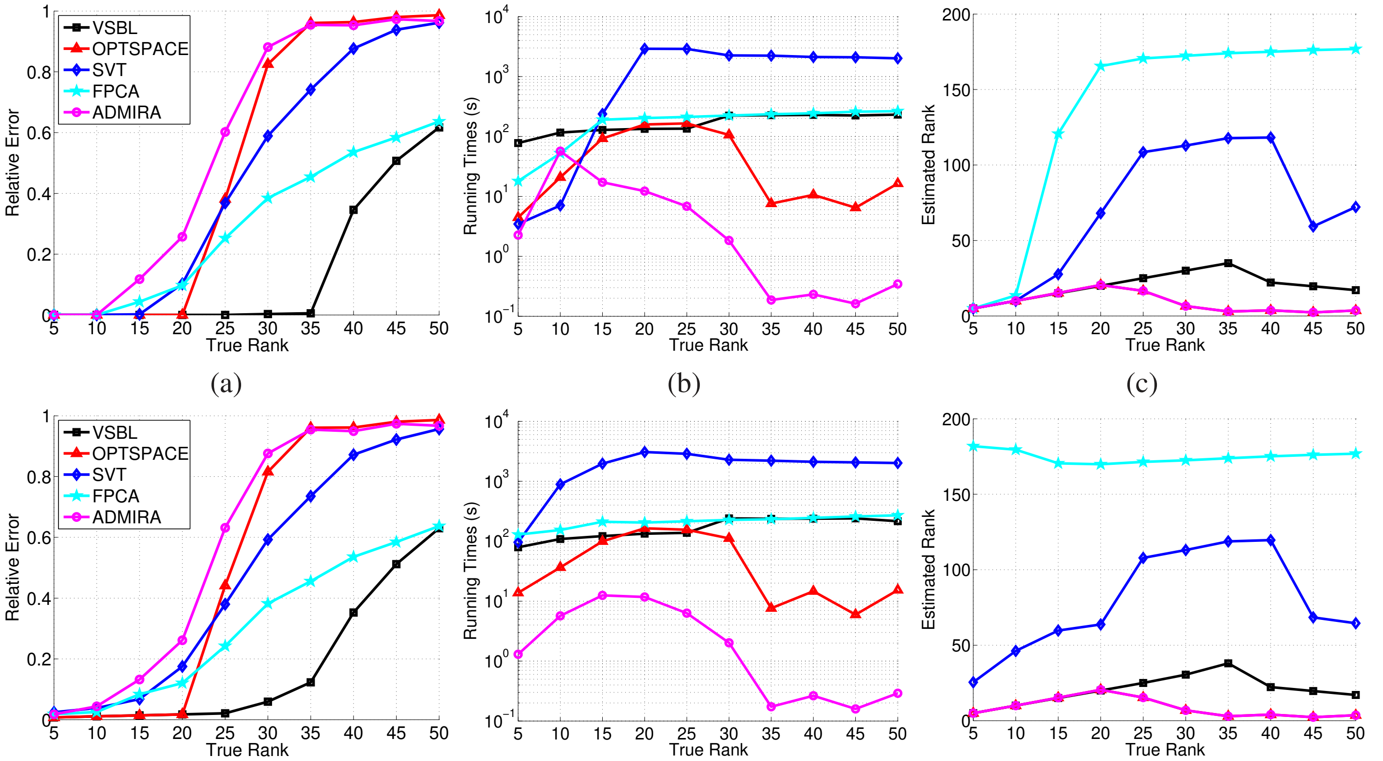
which is a fixed-point update for $\langle \alpha_{ij} \rangle$. We have also observed empirically that using these updates instead of (47) leads to much faster convergence and enhanced sparsity, although no theoretical convergence guarantees exist. Note that this update is also used in the original formulation of sparse Bayesian learning in [38].

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 (23) and (27) 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 [50] to obtain a different form for $\mathbf{\Sigma}_j^b$, given by

$$\mathbf{\Sigma}_j^b = \mathbf{\Gamma}^{-1} - \mathbf{\Gamma}^{-1} \langle \mathbf{A}_j \rangle^T \left(\langle \beta \rangle^{-1} \mathbf{I}_k + \langle \mathbf{A}_j \rangle \mathbf{\Gamma}^{-1} \langle \mathbf{A}_j \rangle^T \right)^{-1} \langle \mathbf{A}_j \rangle \mathbf{\Gamma}^{-1}, \tag{53}$$

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 precisions assume very large values in the very first iterations, and the norms of the corresponding columns become numerically equal to zero, so that they can be removed from the model (similarly to [38]). Other optimizations can also be implemented such as using the conjugate gradient method to solve for posterior means in (22) and (26), 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 (53)) 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.



(a) (b) (c)

(d) (e) (f)

Fig. 1. Estimation results with matrices of size 500×500 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. The legend is common to all figures.

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.

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. The source code developed to obtain the results shown in this section can be found at `https://netfiles.uiuc.edu/dbabac`

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 500×500 of ranks $r = 5, \ldots, 50$ by randomly sampling $500 \times r$ 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]. All of these are deterministic methods with different optimization strategies: OPTSPACE is based on optimization over the Grasmann manifold, SVT uses nuclear-norm minimization with singular-value thresholding, FPCA is a fixed-point Bregman iterative method, and ADMIRA is an efficient greedy method which iteratively adds components during reconstruction.

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. Notice also that in the proposed method VSBL, all required parameters (including the noise variance) are estimated in an automated fashion. In VSBL, we use $\|\hat{\mathbf{X}}^{i} - \hat{\mathbf{X}}^{i-1}\|_{\mathrm{F}}/\|\hat{\mathbf{X}}^{i-1}\|_{\mathrm{F}} < 10^{-5}$ as the convergence criteria, where $\hat{\mathbf{X}}^{i}$ and $\hat{\mathbf{X}}^{i-1}$ are estimates of $\mathbf{X}$ in the i^{th} and $(i-1)^{\mathrm{th}}$ iterations, respectively.

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 20 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 all cases where the rank $r \leq 35$. As expected, errors in both the recovery and the estimated rank increase as the original rank increases. OPTSPACE and ADMIRA generally underestimate the rank, whereas FPCA and SVT consistently 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 500×500 matrices of fixed rank of 10 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 $r(m+n-r)$ degrees of freedom (df), and the oversampling degrees of freedom (osdf) is defined as pmn/df with the number of measurements pmn [23]. Experimental results for $\mathrm{osdf} = 2, 3, \ldots, 7$ are depicted in Figure 2 for the same noise conditions as above. The corresponding sampling ratios are $p \approx 0.08, 0.12, 0.16, 0.20, 0.24, 0.28$. It is evident that VSBL provides very accurate reconstructions and estimates the correct rank even with very low number of observations. 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.

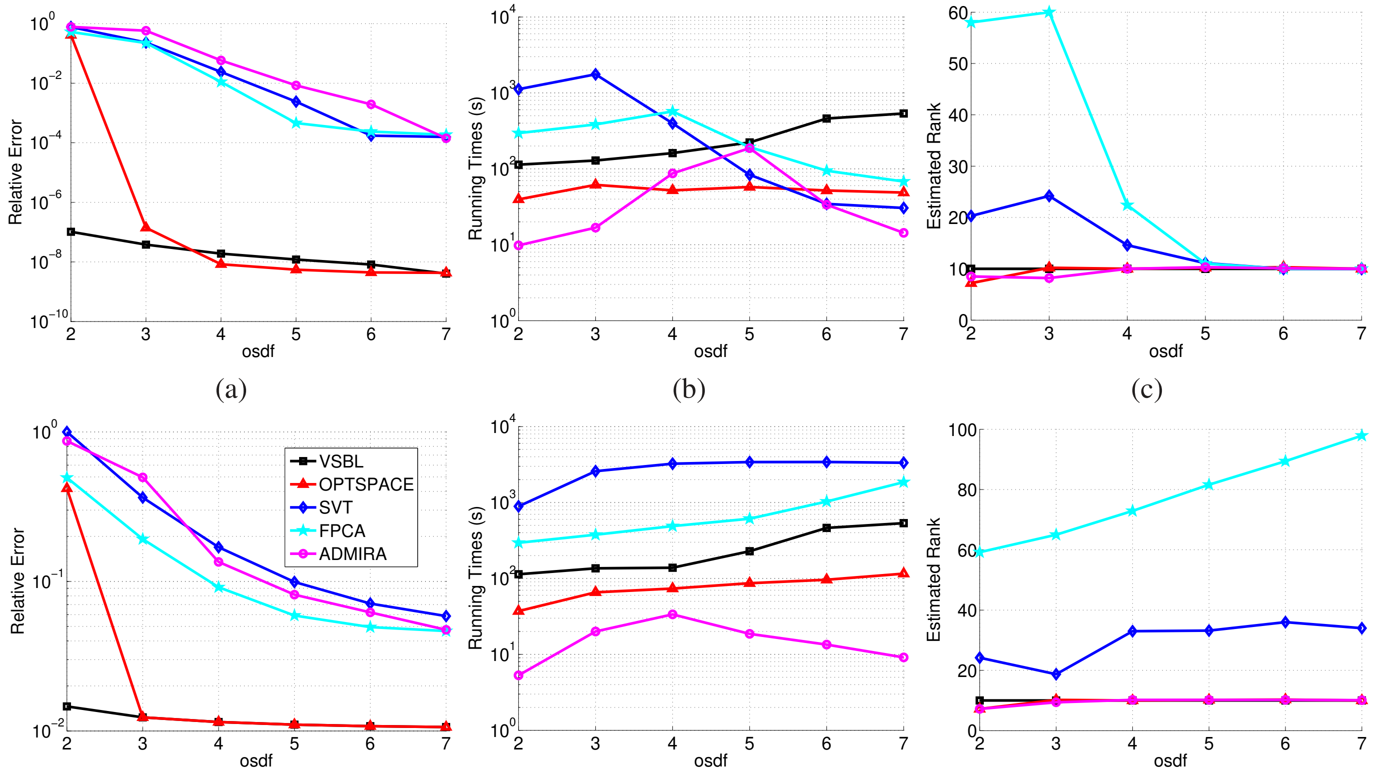
We next illustrate a real-world application of low-rank matrix completion methods on rating prediction from existing ratings. We use the Jester joke[2] and MovieLens[3] datasets, which are commonly used for testing recommendation systems. The Jester joke dataset contains user ratings on jokes where the ratings range from -10 to 10 with 200 quantization levels. The MovieLens dataset consists of user ratings on movies with integer ratings ranging from 1 to 5. Most of the entries are not available in these datasets, and the goal is predicting the missing entries by modeling the dataset as low-rank.

In the Jester joke data set, we generated a full rating matrix 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 real data sets, Jester data set is not low rank or even approximately low rank. To account for this in the proposed algorithm, we used a fixed, high value for the noise variance ($\beta^{-1} = 20$) to encourage low-rank estimates (other values > 5 provided very similar results). 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.

In the MovieLens data set, we experimented with the 100k dataset with 100,000 ratings from 1000 users on 1700 movies, and the 1M dataset with 1,000,209 ratings from 6040 users on 3900 movies. In both datasets, we randomly generated subsets of the rating matrices by sampling $p = 0.1$ and $p = 0.5$ of the available ratings for each user. Note that the rating matrices are very sparse: The 100k dataset contains only about 5% of the entries,

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

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



(a) (b) (c)

(d) (e) (f)

Fig. 2. Estimation results with matrices of size 500 × 500 of rank 10 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. The legend is common to all figures.

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
FPCA	0.2026	0.2046	0.2052	0.2096	0.2060	0.2051

and the 1M dataset about 4%. Therefore, these datasets are extremely challenging for matrix completion methods which do not take any other information into account (such as user and genre information). The NMAE results (average of 10 realizations) are shown in Table II. It can be observed that VSBL provides lower prediction errors than other methods.

B. Robust PCA

1) Comparison with state-of-the-art: 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 $[-10, 10]$. The number of non-zero entries is set equal to $0.05mn$. 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 $\parallel \hat{\mathbf{X}}^i - \mathbf{X} \parallel_{\text{F}} / \parallel \mathbf{X} \parallel_{\text{F}}$ and $\parallel \hat{\mathbf{E}}^i - \mathbf{E} \parallel_{\text{F}} / \parallel \mathbf{E} \parallel_{\text{F}}$ and the convergence criterion is $\|\hat{\mathbf{X}}^i - \hat{\mathbf{X}}^{i-1}\|_{\text{F}}/\|\hat{\mathbf{X}}^{i-1}\|_{\text{F}} < 10^{-5}$, where $\hat{\mathbf{X}}^i$ and $\hat{\mathbf{E}}^i$ represent the estimates in the i^{th} iteration.

TABLE II
NMAE VALUES ON THE MOVIELENS DATA SETS

Dataset	100k		1M	
	p = 0.1	p=0.5	p = 0.1	p=0.5
VSBL	0.2045	0.1350	0.1840	0.1460
OPT	0.2110	0.1780	0.1875	0.1755
FPCA	0.3175	0.2633	0.1990	0.1862

We present comparisons with the Bayesian method proposed in [37] (denoted by BRPCA) and the optimization-based method in [36] (denoted by ALM). As mentioned before, BRPCA is based on factorizations of both the low-rank and the sparse components, and low-rank and sparsity constraints are imposed using a combination of beta-Bernoulli priors on each component. The inference is performed using a Markov chain Monte Carlo (MCMC) sampling scheme. On the other hand, ALM is based on soft-thresholding the singular values of the low-rank component and elements of the sparse component. The inference is deterministic and is based on the augmented Lagrange multiplier method. We use the exact inference method in [36] and manually tuned its parameters to report its best performance in terms of recovery error. The proposed method, developed in Section III-B, is denoted as VBRPCA.

Table III 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 high accuracy in all test cases. While the running times of ALM and VBRPCA are very similar, the proposed method generally showed faster convergence rate, especially in large matrix sizes. 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 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. We empirically found out that ALM is very sensitive to this parameter, and generally requires careful tuning (see [37] for a related discussion). A comparison of ALM and VBRPCA is shown in Fig. 3, where matrices of size $m \times n$ with $m = n = 500, 1000, 2000, 3000, 4000$ are generated with the rank of the low-rank component equal to $0.05m$ and the number of non-zeros in the sparse component equal to $0.05mn$. Results are shown both with noiseless and noisy observations, where in the latter case noise variance is set equal to 10^{-3}. It is evident that while ALM provides very low reconstruction errors in the noiseless case, its performance is significantly decreased and the rank is consistently overestimated when dense noise is present. On the other hand, the performance of VBRPCA is comparable to ALM in the noiseless case and better in the noisy case. In addition, VBRPCA estimates the rank correctly in both cases and requires lower computation times than ALM. It should be emphasized that ALM required careful manual tuning of its convergence parameter in all experiments, while VBRPCA automatically estimates all algorithmic parameters including the dense noise level. BRPCA has a similar mechanism for automatic noise estimation through a Bayesian formulation, but its results are generally inferior to the proposed method and its computational complexity is significantly higher.

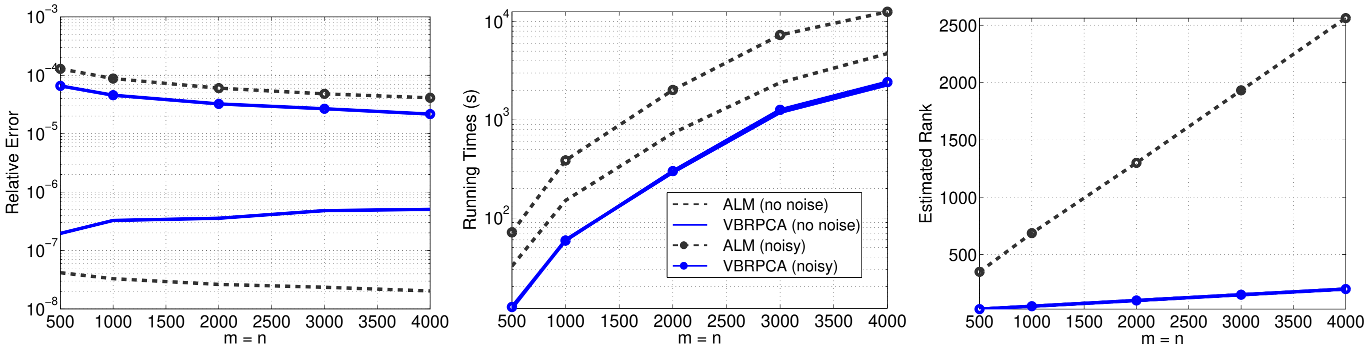
Our second example illustrates a real-world application of robust PCA methods. We consider the foreground/background separation problem in video as in [37]. Each column of the data matrix $\mathbf{Y}$ is generated by concatenating pixels of one video frame into a vector. In this application, the low-rank 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[4] consisting of 158 frames of size 192 × 144. Example results obtained by the algorithms in one video frame are shown in Fig. 4. 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

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

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

Method	σ	m=n	rank($\mathbf{X}$)	$\parallel \mathbf{E} \parallel_0$	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	2000	5	3.5×10^{-8}	1.3×10^{-7}	1.71
ALM	0	200	10	2000	10	1.5×10^{-8}	0.7×10^{-7}	2.53
ALM	0	500	25	12500	25	0.6×10^{-8}	0.5×10^{-7}	15.3
ALM	0	1000	50	50000	50	4.4×10^{-8}	6.0×10^{-7}	83.94
BRPCA	0	200	5	2000	4	3.5×10^{-5}	1.3×10^{-5}	41.40
BRPCA	0	200	10	2000	10	5.2×10^{-4}	1.2×10^{-4}	358.20
BRPCA	0	500	25	12500	25	1.1×10^{-4}	7.2×10^{-4}	3116.33
BRPCA	0	1000	50	50000	50	9.5×10^{-5}	1.9×10^{-4}	27160.10
VBRPCA	0	200	5	2000	5	0.5×10^{-6}	4.0×10^{-7}	0.81
VBRPCA	0	200	10	2000	10	1.8×10^{-6}	2.0×10^{-7}	1.02
VBRPCA	0	500	25	12500	25	2.1×10^{-6}	4.8×10^{-7}	7.78
VBRPCA	0	1000	50	50000	50	3.5×10^{-6}	1.7×10^{-7}	38.47
ALM	10^{-3}	200	5	2000	140	2.6×10^{-4}	5.5×10^{-4}	4.39
ALM	10^{-3}	200	10	2000	140	2.0×10^{-4}	6.0×10^{-4}	4.63
ALM	10^{-3}	500	25	12500	349	1.3×10^{-4}	6.0×10^{-4}	40.12
ALM	10^{-3}	1000	50	50000	663	0.9×10^{-4}	5.0×10^{-4}	229.98
BRPCA	10^{-3}	200	5	2000	5	4.1×10^{-3}	1.2×10^{-3}	45.39
BRPCA	10^{-3}	200	10	2000	10	4.2×10^{-3}	1.7×10^{-3}	370.21
BRPCA	10^{-3}	500	25	12500	25	7.2×10^{-3}	6.7×10^{-3}	3360.10
BRPCA	10^{-3}	1000	50	50000	50	8.2×10^{-3}	5.4×10^{-3}	27412.21
VBRPCA	10^{-3}	200	5	2000	5	1.0×10^{-5}	1.7×10^{-4}	0.90
VBRPCA	10^{-3}	200	10	2000	10	9.0×10^{-5}	1.9×10^{-4}	1.19
VBRPCA	10^{-3}	500	25	12500	25	6.4×10^{-5}	1.8×10^{-4}	7.83
VBRPCA	10^{-3}	1000	50	50000	50	3.7×10^{-5}	1.8×10^{-4}	39.98

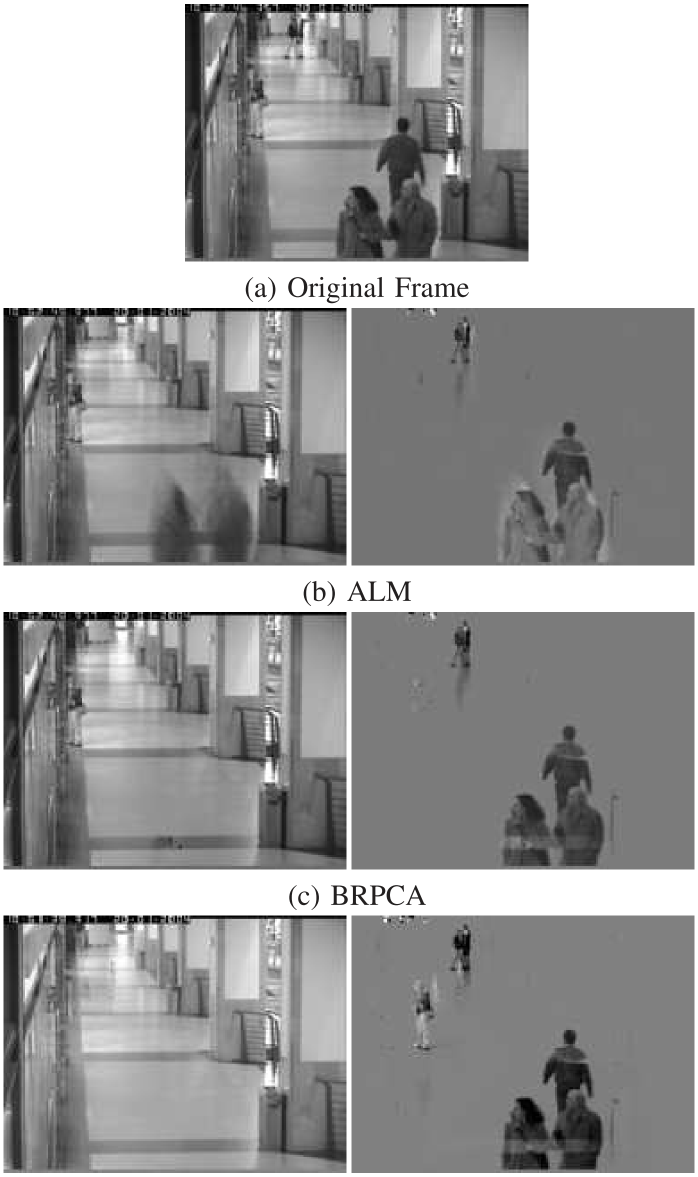


(a) (b) (c)

Fig. 3. Comparison of ALM and VBRPCA with varying matrix sizes for robust PCA. Matrices are of size $m \times n$ with $m = n$, the rank of the low-rank component is set equal to $0.05m$, and the number of non-zeros in the sparse component is set equal to $0.05mn$. (a) Relative reconstruction errors, (b) running times, and (c) estimated ranks. The legend is common to all figures.

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 (the estimated rank in this case is 1). 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.

2) Comparison of inference methods: Although in this work we developed the algorithms based on variational Bayesian inference, other inference methods can be employed as well based on the same Bayesian modeling shown in Sec. II. Here we compare VBRPCA with two other inference schemes, namely, maximum *a posteriori* (MAP)



(a) Original Frame

(b) ALM

(c) BRPCA

(d) VBRPCA

Fig. 4. Video background/foreground separation. (a) Original video frame, the reconstructions by (b) ALM, (c) BRPCA, and (d) VBRPCA. *Left:* background reconstruction, *right:* foreground reconstruction.

estimation and Gibbs sampling. This comparison will provide some insight both on the effectiveness and accuracy of the variational Bayesian approach. Moreover, it can be used to assess the accuracy of the variational approximation.

For both methods, using the observation model (16) and the priors given in (8), (9), (10), (14), (17), (18), we form the conditional posterior distributions as

$$\mathrm{p}(\mathbf{a}_{i\cdot}|\mathbf{Y},\mathbf{z}\backslash\mathbf{a}_{i\cdot}) = \mathcal{N}\left(\mathbf{a}_{i\cdot}|\tilde{\mathbf{a}}_{i\cdot},\tilde{\mathbf{\Sigma}}^{A}\right) \ , \tag{54}$$

$$\mathrm{p}(\mathbf{b}_{i\cdot}|\mathbf{Y},\mathbf{z}\backslash\mathbf{b}_{i\cdot}) = \mathcal{N}\left(\mathbf{b}_{i\cdot}|\tilde{\mathbf{b}}_{i\cdot},\tilde{\mathbf{\Sigma}}^{B}\right) \ , \tag{55}$$

$$\mathrm{p}(E_{ij}|\mathbf{Y},\mathbf{z}\backslash E_{ij}) = \mathcal{N}\left(E_{ij}|\tilde{E}_{ij},\tilde{\Sigma}^{E}_{ij}\right) \ , \tag{56}$$

$$\mathrm{p}(\gamma_i|\mathbf{Y},\mathbf{z}\backslash\gamma_i) = \mathrm{Gamma}\left(\frac{a+m+n}{2},\frac{2}{2b+\tilde{\mathbf{a}}_{\cdot i}^{T}\tilde{\mathbf{a}}_{\cdot i}+\tilde{\mathbf{b}}_{\cdot i}^{T}\tilde{\mathbf{b}}_{\cdot i}}\right) \ , \tag{57}$$

$$\mathrm{p}(\alpha_{ij}|\mathbf{Y},\mathbf{z}\backslash\alpha_{ij}) = \mathrm{Gamma}\left(\frac{1}{2},\frac{2}{\tilde{E}^{2}_{ij}}\right) \ , \tag{58}$$

$$\mathrm{p}(\beta|\mathbf{Y},\mathbf{z}\backslash\beta) = \mathrm{Gamma}\left(\frac{mn}{2},\frac{2}{\parallel \mathbf{Y}-\tilde{\mathbf{A}}\tilde{\mathbf{B}}^{T}-\tilde{\mathbf{E}}\parallel_{\mathrm{F}}^{2}}\right) \ , \tag{59}$$

where $\mathbf{z}$ is the set of all latent variables as before, $\mathrm{Gamma}(k,\theta)$ is the Gamma distribution with shape parameter k and scale parameter θ (see (10)). The parameters of the distributions above are given by

$$\tilde{\mathbf{a}}_{i\cdot} = \tilde{\beta}\,\tilde{\mathbf{\Sigma}}^{A}\,\tilde{\mathbf{B}}^{T}\ \left(\mathbf{y}_{i\cdot}-\tilde{\mathbf{e}}_{i\cdot}\right)^{T} \ , \tag{60}$$

$$\tilde{\mathbf{\Sigma}}^{A} = \left(\tilde{\beta}\,\tilde{\mathbf{B}}^{T}\tilde{\mathbf{B}}+\tilde{\mathbf{\Gamma}}\right)^{-1} \ , \tag{61}$$

$$\tilde{\mathbf{b}}_{j\cdot}^{T} = \tilde{\beta}\,\tilde{\mathbf{\Sigma}}^{B}\,\tilde{\mathbf{A}}^{T}\ \left(\mathbf{y}_{\cdot j}-\tilde{\mathbf{e}}_{\cdot j}\right) \ , \tag{62}$$

$$\tilde{\mathbf{\Sigma}}^{B} = \left(\tilde{\beta}\,\tilde{\mathbf{A}}^{T}\tilde{\mathbf{A}}+\tilde{\mathbf{\Gamma}}\right)^{-1} \ , \tag{63}$$

$$\tilde{\mathbf{\Gamma}} = \mathrm{diag}\left(\tilde{\gamma}_i\right) \ , \tag{64}$$

$$\tilde{E}_{ij} = \tilde{\beta}\ \tilde{\Sigma}^{E}_{ij}\ \left(Y_{ij}-\tilde{\mathbf{a}}_{i\cdot}\tilde{\mathbf{b}}_{j\cdot}^{T}\right) \ , \tag{65}$$

$$\tilde{\Sigma}^{E}_{ij} = \frac{1}{\tilde{\beta}+\tilde{\alpha}_{ij}} \, . \tag{66}$$

The MAP estimates are found as the mode of these distributions, whereas in Gibbs sampling we sample from these distributions in an alternating fashion and collect the sampled values. In both cases, the estimated values are denoted with a tilde (˜).

For empirical comparison, we create synthetic datasets similar to Section V-B1 where the low-rank component $\mathbf{X}$ is 400×400 with coefficients drawn from a $\mathcal{N}(0,1)$ distribution, the sparse matrix $\mathbf{E}$ has 8000 non-zero entries (sparsity level 5%) drawn from a uniform distribution $[-10, 10]$ located uniformly at random. We consider both noiseless and noisy settings where in the latter case white Gaussian noise with variance 10^{-3} is added to the observations. The rank of $\mathbf{X}$ is varied from 10 to 60 in steps of 10.

The relative reconstruction errors and rank estimates of $\mathbf{X}$, along with average running times are shown in Fig. 5. Reconstruction errors in the estimates of the sparse component $\mathbf{E}$ are similar to those of $\mathbf{X}$ and are not shown. It can be seen that both variational Bayesian inference and Gibbs sampling provide more accurate estimates than MAP. The MAP approach is very sensitive to the values of the hyperparameters a and b, and is prone to over- and under-fitting depending on their selection. Similar results are obtained even when the correct noise level is provided to MAP (data not shown), indicating that MAP is unable to avoid undesirable local minima. An important result is that the variational Bayesian inference provides results comparable to those of Gibbs sampling in terms of reconstruction error, and for $\mathrm{rank} \leq 40$, it correctly estimates the unknown rank and the sparsity level. In addition, its running times are 2-4 orders of magnitude lower than those of Gibbs sampling, although a relatively low number of iterations are used for the sampling method (10000 for burn-in and 2000 for collection, compared to 25000 burn-in and 5000 collection in BRPCA [37]). Its running times are also comparable to the MAP approach, which has lower complexity per iteration but generally requires many more iterations for convergence.

In summary, the inference procedure developed in this work based on variational Bayesian analysis provides very accurate results compared to sampling while being computationally considerably more efficient.

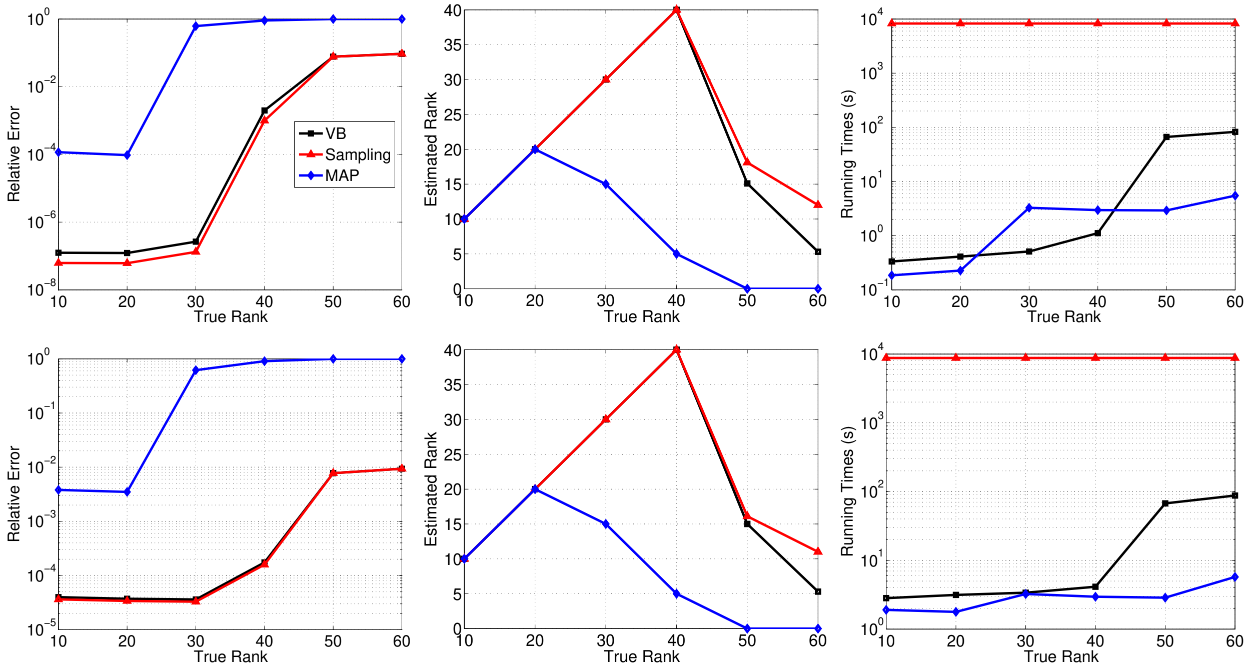


Fig. 5. Comparison of inference methods. *Top row:* noiseless case, *bottom row:* noisy case. *From left to right:* relative reconstruction errors, estimated ranks, and running times.

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 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. The sparse error component in the robust PCA problem is also modeled and effectively inferred by sparse Bayesian learning principles. We modeled the remaining unknown variables and observations within the hierarchical Bayesian framework and developed inference methods based on mean-field variational Bayes approximating the posteriors of interest. This inference scheme is shown to be advantageous both in terms of computational requirements and estimation performance compared to other inference schemes. 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.

Acknowledgments

The authors would like to thank Shinichi Nakajima, the associate editor and anonymous reviewers for their comments and suggestions that helped improve the paper.

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. Candès, 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.

[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. Candès, 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] N. Ding, Y. A. Qi, R. Xiang, I. Molloy, and N. Li, "Nonparametric Bayesian matrix factorization by Power-EP," in *AISTATS*, vol. 9, 2010, pp. 169–176.

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

[32] M. Zhou, C. Wang, M. Chen, J. Paisley, D. Dunson, and L. Carin, "Nonparametric Bayesian matrix completion," in *Proc. IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM2010)*, Oct 2010.

[33] F. Wood and T. L. Griffiths, "Particle filtering for nonparametric Bayesian matrix factorization," in *Advances in Neural Information Processing Systems 19*. Cambridge, MA: MIT Press, 2006, pp. 1513–1520.

[34] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *Proceedings of the 25th international conference on Machine learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 880–887.

[35] 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.

[36] 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.

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

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

[39] T. Raiko, A. Ilin, and J. Karhunen, "Principal component analysis for large scale problems with lots of missing values," in *Proceedings of the 18th European Conference on Machine Learning (ECML 2007)*, J. Kok, J. Koronacki, R. Mantaras, S. Matwin, D. Mladenic, and A. Skowron, Eds. Springer Berlin / Heidelberg, 2007, vol. 4701, pp. 691–698.

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

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

[42] 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.

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

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

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

[46] B. Lakshminarayanan, G. Bouchard, and C. Archambeau, “Robust Bayesian matrix factorization,” in *Proc. Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2011.

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

[48] C. M. Bishop, “Bayesian PCA,” in *NIPS*, 1999, pp. 382–388.

[49] ——, “Variational principal components,” in *Proceedings Ninth International Conference on Artificial Neural Networks, ICANN’99*, 1999, pp. 509–514.

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