Variational Bayesian Sparse Additive Matrix Factorization

Shinichi Nakajima
Nikon Corporation, Japan
nakajima.s@nikon.co.jp

Masashi Sugiyama
Tokyo Institute of Technology, Japan.
sugi@cs.titech.ac.jp
http://sugiyama-www.cs.titech.ac.jp/˜sugi

S. Derin Babacan
University of Illinois at Urbana-Champaign, USA.
dbabacan@illinois.edu

Abstract

Principal component analysis (PCA) approximates a data matrix with a low-rank one by imposing sparsity on its singular values. Its robust variant can cope with spiky noise by introducing an element-wise sparse term. In this paper, we extend such sparse matrix learning methods, and propose a novel framework called *sparse additive matrix factorization* (SAMF). SAMF systematically induces various types of sparsity by a Bayesian regularization effect, called *model-induced regularization*. Although group LASSO also allows us to design arbitrary types of sparsity on a matrix, SAMF, which is based on the Bayesian framework, provides inference without any requirement for manual parameter tuning. We propose an efficient iterative algorithm called the *mean update* (MU) for the variational Bayesian approximation to SAMF, which gives the global optimal solution for a large subset of parameters in each step. We demonstrate the usefulness of our method on benchmark datasets and a foreground/background video separation problem.

Keywords

variational Bayes, robust PCA, matrix factorization, sparsity, model-induced regularization.

1 Introduction

Principal component analysis (PCA) (Hotelling, 1933) is a classical method for obtaining low-dimensional expression of data. PCA can be regarded as approximating a data matrix with a low-rank one by imposing sparsity on its singular values. A robust variant of PCA further copes with sparse spiky noise included in observations (Candès et al., 2011; Ding et al., 2011; Babacan et al., 2012).

In this paper, we extend the idea of robust PCA, and propose a more general framework called *sparse additive matrix factorization* (SAMF). The proposed SAMF can handle various types of sparse noise such as row-wise and column-wise sparsity, in addition to element-wise sparsity (spiky noise) and low-rank sparsity (low-dimensional expression); furthermore, their arbitrary additive combination is also allowed. In the context of robust PCA, row-wise and column-wise sparsity can capture noise observed when some sensors are broken and their outputs are always unreliable, or some accident disturbs all sensor outputs at a time.

Flexibility of SAMF in sparsity design allows us to incorporate side information more efficiently. We show such an example in foreground/background video separation, where sparsity is induced based on image segmentation. Although group LASSO (Yuan and Lin, 2006; Raman et al., 2009) also allows arbitrary sparsity design on matrix entries, SAMF, which is based on the Bayesian framework, enables us to estimate all unknowns from observations, and allows us to enjoy inference without manual parameter tuning.

Technically, our approach induces sparsity by the so-called *model-induced regularization* (MIR) (Nakajima and Sugiyama, 2011). MIR is an implicit regularization property of the Bayesian approach, which is based on one-to-many (i.e., redundant) mapping of parameters and outcomes (Watanabe, 2009). In the case of matrix factorization, an observed matrix is decomposed into two redundant matrices, which was shown to induce sparsity in the singular values under the variational Bayesian approximation (Nakajima and Sugiyama, 2011).

We show that MIR in SAMF can be interpreted as *automatic relevance determination* (ARD) (Neal, 1996), which is a popular Bayesian approach to inducing sparsity. Nevertheless, we argue that the MIR formulation is more preferable since it allows us to derive a practically useful algorithm called the *mean update* (MU) from a recent theoretical result (Nakajima et al., 2013): The MU algorithm is based on the variational Bayesian approximation, and gives the global optimal solution for a large subset of parameters in each step. Through experiments, we show that the MU algorithm compares favorably with a standard iterative algorithm for variational Bayesian inference.

2 Formulation

In this section, we formulate the sparse additive matrix factorization (SAMF) model.

2.1 Examples of Factorization

In standard MF, an observed matrix $V \in \mathbb{R}^{L \times M}$ is modeled by a low rank target matrix $U \in \mathbb{R}^{L \times M}$ contaminated with a random noise matrix $\mathcal{E} \in \mathbb{R}^{L \times M}$.

$$V = U + \mathcal{E}.$$

Then the target matrix U is decomposed into the product of two matrices $A \in \mathbb{R}^{M \times H}$ and $B \in \mathbb{R}^{L \times H}$:

$$U^{\text{low-rank}} = BA^\top = \sum_{h=1}^{H} \boldsymbol{b}_h \boldsymbol{a}_h^\top, \tag{1}$$

where $\top$ denotes the transpose of a matrix or vector. Throughout the paper, we denote a column vector of a matrix by a bold small letter, and a row vector by a bold small letter with a tilde:

$$\begin{aligned} A &= (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_H) = (\widetilde{\boldsymbol{a}}_1, \ldots, \widetilde{\boldsymbol{a}}_M)^\top, \\ B &= (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_H) = (\widetilde{\boldsymbol{b}}_1, \ldots, \widetilde{\boldsymbol{b}}_L)^\top. \end{aligned}$$

The last equation in Eq.(1) implies that the *plain* matrix product (i.e., $BA^\top$) is the sum of rank-1 components. It was elucidated that this product induces an implicit regularization effect called *model-induced regularization* (MIR), and a low-rank (singular-component-wise sparse) solution is produced under the variational Bayesian approximation (Nakajima and Sugiyama, 2011).

Let us consider other types of factorization:

$$U^{\text{row}} = \mathit{\Gamma}_E D = (\gamma_1^e \widetilde{\boldsymbol{d}}_1, \ldots, \gamma_L^e \widetilde{\boldsymbol{d}}_L)^\top, \tag{2}$$

$$U^{\text{column}} = E \mathit{\Gamma}_D = (\gamma_1^d \boldsymbol{e}_1, \ldots, \gamma_M^d \boldsymbol{e}_M), \tag{3}$$

where $\mathit{\Gamma}_D = \mathrm{diag}(\gamma_1^d, \ldots, \gamma_M^d) \in \mathbb{R}^{M \times M}$ and $\mathit{\Gamma}_E = \mathrm{diag}(\gamma_1^e, \ldots, \gamma_L^e) \in \mathbb{R}^{L \times L}$ are diagonal matrices, and $D, E \in \mathbb{R}^{L \times M}$. These examples are also matrix products, but one of the factors is restricted to be diagonal. Because of this diagonal constraint, the l-th diagonal entry γ_l^e in $\mathit{\Gamma}_E$ is shared by all the entries in the l-th row of U^{row} as a common factor. Similarly, the m-th diagonal entry γ_m^d in $\mathit{\Gamma}_D$ is shared by all the entries in the m-th column of U^{column}.

Another example is the Hadamard (or element-wise) product:

$$U^{\text{element}} = E * D, \text{ where } (E * D)_{l,m} = E_{l,m} D_{l,m}. \tag{4}$$

In this factorization form, no entry in E and D is shared by more than one entry in U^{element}.

In fact, the forms (2)–(4) of factorization induce different types of sparsity, through the MIR mechanism. In Section 2.2, they will be derived as a row-wise, a column-wise, and an element-wise sparsity inducing terms, respectively, within a unified framework.

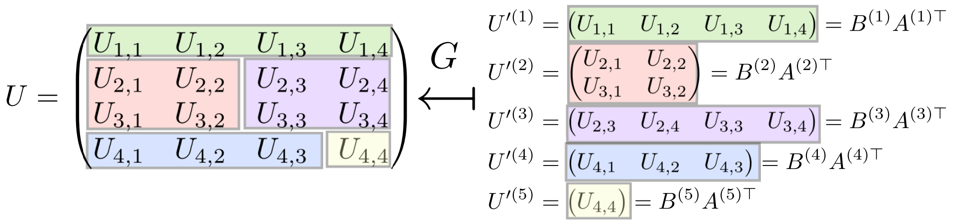

$U =$
G
$U'^{(1)} = (U_{1,1} \; U_{1,2} \; U_{1,3} \; U_{1,4}) = B^{(1)}A^{(1)\top}$
$U'^{(2)} = \begin{pmatrix} U_{2,1} & U_{2,2} \\ U_{3,1} & U_{3,2} \end{pmatrix} = B^{(2)}A^{(2)\top}$
$U'^{(3)} = (U_{2,3} \; U_{2,4} \; U_{3,3} \; U_{3,4}) = B^{(3)}A^{(3)\top}$
$U'^{(4)} = (U_{4,1} \; U_{4,2} \; U_{4,3}) = B^{(4)}A^{(4)\top}$
$U'^{(5)} = (U_{4,4}) = B^{(5)}A^{(5)\top}$

Figure 1: An example of SMF-term construction. $G(\cdot;\mathcal{X})$ with $\mathcal{X} : (k, l', m') \mapsto (l, m)$ maps the set $\{U'^{(k)}\}_{k=1}^{K}$ of the PR matrices to the target matrix U, so that $U'^{(k)}_{l',m'} = U_{\mathcal{X}(k,l',m')} = U_{l,m}$.

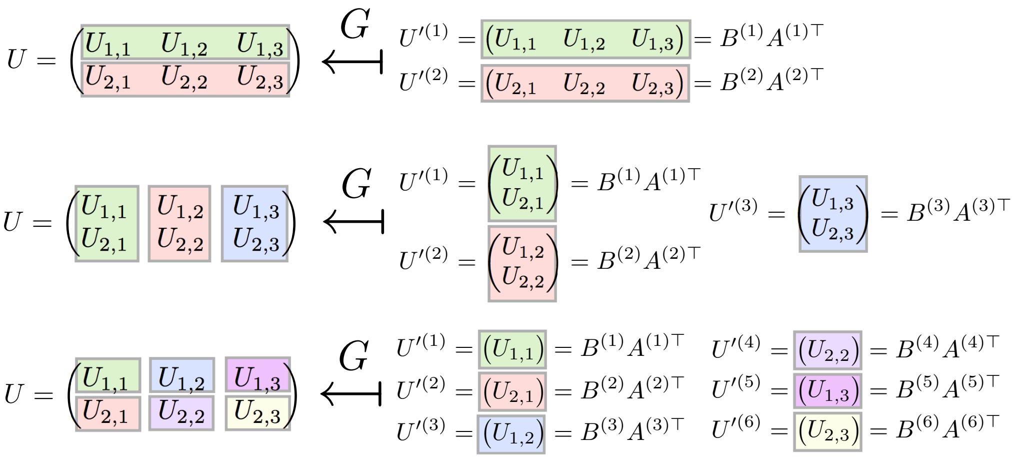

$U =$
G
$U'^{(1)} = (U_{1,1} \; U_{1,2} \; U_{1,3}) = B^{(1)}A^{(1)\top}$
$U'^{(2)} = (U_{2,1} \; U_{2,2} \; U_{2,3}) = B^{(2)}A^{(2)\top}$
$U =$
G
$U'^{(1)} = \begin{pmatrix} U_{1,1} \\ U_{2,1} \end{pmatrix} = B^{(1)}A^{(1)\top}$
$U'^{(2)} = \begin{pmatrix} U_{1,2} \\ U_{2,2} \end{pmatrix} = B^{(2)}A^{(2)\top}$
$U'^{(3)} = \begin{pmatrix} U_{1,3} \\ U_{2,3} \end{pmatrix} = B^{(3)}A^{(3)\top}$
$U =$
G
$U'^{(1)} = (U_{1,1}) = B^{(1)}A^{(1)\top}$
$U'^{(2)} = (U_{2,1}) = B^{(2)}A^{(2)\top}$
$U'^{(3)} = (U_{1,2}) = B^{(3)}A^{(3)\top}$
$U'^{(4)} = (U_{2,2}) = B^{(4)}A^{(4)\top}$
$U'^{(5)} = (U_{1,3}) = B^{(5)}A^{(5)\top}$
$U'^{(6)} = (U_{2,3}) = B^{(6)}A^{(6)\top}$

Figure 2: SMF-term construction for the row-wise (top), the column-wise (middle), and the element-wise (bottom) sparse terms.

2.2 A General Expression of Factorization

Our general expression consists of partitioning, rearrangement, and factorization. The following is the form of a sparse matrix factorization (SMF) term:

$$U = G(\{U'^{(k)}\}_{k=1}^{K}; \mathcal{X}), \text{ where } U'^{(k)} = B^{(k)}A^{(k)\top}. \tag{5}$$

Here, $\{A^{(k)}, B^{(k)}\}_{k=1}^{K}$ are parameters to be estimated, and $G(\cdot;\mathcal{X}) : \mathbb{R}^{\prod_{k=1}^{K}(L'^{(k)} \times M'^{(k)})} \mapsto \mathbb{R}^{L \times M}$ is a designed function associated with an index mapping parameter $\mathcal{X}$, which will be explained shortly.

Figure 1 shows how to construct an SMF term. First, we partition the entries of U into K parts. Then, by rearranging the entries in each part, we form partitioned-and-rearranged (PR) matrices $U'^{(k)} \in \mathbb{R}^{L'^{(k)} \times M'^{(k)}}$ for $k = 1, \ldots, K$. Finally, each of $U'^{(k)}$ is decomposed into the product of $A^{(k)} \in \mathbb{R}^{M'^{(k)} \times H'^{(k)}}$ and $B^{(k)} \in \mathbb{R}^{L'^{(k)} \times H'^{(k)}}$, where $H'^{(k)} \leq \min(L'^{(k)}, M'^{(k)})$.

In Eq.(5), the function $G(\cdot;\mathcal{X})$ is responsible for partitioning and rearrangement: It

Table 1: Examples of SMF term. See the main text for details.

Factorization	Induced sparsity	K	$(L'^{(k)}, M'^{(k)})$	$\mathcal{X} : (k, l', m') \mapsto (l, m)$
$U = BA^\top$	low-rank	1	(L, M)	$\mathcal{X}(1, l', m') = (l', m')$
$U = \Gamma_E D$	row-wise	L	$(1, M)$	$\mathcal{X}(k, 1, m') = (k, m')$
$U = E\Gamma_D$	column-wise	M	$(L, 1)$	$\mathcal{X}(k, l', 1) = (l', k)$
$U = E * D$	element-wise	$L \times M$	$(1, 1)$	$\mathcal{X}(k, 1, 1) = \textit{vec-order}(k)$

maps the set $\{U'^{(k)}\}_{k=1}^K$ of the PR matrices to the target matrix $U \in \mathbb{R}^{L\times M}$, based on the one-to-one map $\mathcal{X} : (k, l', m') \mapsto (l, m)$ from the indices of the entries in $\{U'^{(k)}\}_{k=1}^K$ to the indices of the entries in U, such that

$$\left(G(\{U'^{(k)}\}_{k=1}^K; \mathcal{X})\right)_{l,m} = U_{l,m} = U_{\mathcal{X}(k,l',m')} = U'^{(k)}_{l',m'}. \tag{6}$$

As will be discussed in Section 4.1, the SMF-term expression (5) under the variational Bayesian approximation induces low-rank sparsity in each partition. This means that partition-wise sparsity is also induced. Accordingly, partitioning, rearrangement, and factorization should be designed in the following manner. Suppose that we are given a required sparsity structure on a matrix (examples of possible side information that suggests particular sparsity structures are given in Section 2.3). We first partition the matrix, according to the required sparsity. Some partitions can be submatrices. We rearrange each of the submatrices on which we do not want to impose low-rank sparsity into a long vector ($U'^{(3)}$ in the example in Figure 1). We leave the other submatrices which we want to be low-rank ($U'^{(2)}$), the vectors ($U'^{(1)}$ and $U'^{(4)}$) and the scalars ($U'^{(5)}$) as they are. Finally, we factorize each of the PR matrices to induce sparsity through the MIR mechanism.

Let us, for example, assume that row-wise sparsity is required. We first make the row-wise partition, i.e., separate $U \in \mathbb{R}^{L\times M}$ into L pieces of M-dimensional row vectors $U'^{(l)} = \widetilde{\boldsymbol{u}}_l^\top \in \mathbb{R}^{1\times M}$. Then, we factorize each partition as $U'^{(l)} = B^{(l)} A^{(l)\top}$ (see the top illustration in Figure 2). Thus, we obtain the row-wise sparse term (2). Here, $\mathcal{X}(k, 1, m') = (k, m')$ makes the following connection between Eqs.(2) and (5): $\gamma_l^e = B^{(k)} \in \mathbb{R}, \widetilde{\boldsymbol{d}}_l = A^{(k)} \in \mathbb{R}^{M\times 1}$ for $k = l$. Similarly, requiring column-wise and element-wise sparsity leads to Eqs.(3) and (4), respectively (see the bottom two illustrations in Figure 2). Table 1 summarizes how to design these SMF terms, where $\textit{vec-order}(k) = (1 + ((k-1) \bmod L), \lceil k/L \rceil)$ goes along the columns one after another in the same way as the *vec* operator forming a vector by stacking the columns of a matrix (in other words, $(U'^{(1)}, \ldots, U'^{(K)})^\top = \textit{vec}(U)$).

2.3 Sparse Additive Matrix Factorization

We define a sparse additive matrix factorization (SAMF) model as the sum of SMF terms (5):

$$V = \sum_{s=1}^{S} U^{(s)} + \mathcal{E}, \tag{7}$$

$$\text{where } U^{(s)} = G(\{B^{(k,s)} A^{(k,s)\top}\}_{k=1}^{K^{(s)}}; \mathcal{X}^{(s)}). \tag{8}$$

In practice, SMF terms should be designed based on side information. Suppose that $V \in \mathbb{R}^{L \times M}$ consists of M samples of L-dimensional sensor outputs. In robust PCA (Candès et al., 2011; Ding et al., 2011; Babacan et al., 2012), an element-wise sparse term is added to the low-rank term, which is expected to be the clean signal, when sensor outputs are expected to contain spiky noise:

$$V = U^{\text{low-rank}} + U^{\text{element}} + \mathcal{E}. \tag{9}$$

Here, it can be said that the "expectation of spiky noise" is used as side information.

Similarly, if we suspect that some sensors are broken, and their outputs are unreliable over all M samples, we should prepare the row-wise sparse term to capture the expected row-wise noise, and try to keep the estimated clean signal $U^{\text{low-rank}}$ uncontaminated with the row-wise noise:

$$V = U^{\text{low-rank}} + U^{\text{row}} + \mathcal{E}.$$

If we know that some accidental disturbances occurred during the observation, but do not know their exact locations (i.e., which samples are affected), the column-wise sparse term can effectively capture these disturbances.

The SMF expression (5) enables us to use side information in a more flexible way. In Section 5.4, we show that our method can be applied to a foreground/background video separation problem, where *moving* objects (such as a person in Figure 3) are considered to belong to the foreground. Previous approaches (Candès et al., 2011; Ding et al., 2011; Babacan et al., 2012) constructed the observation matrix V by stacking all pixels in each frame into each column (Figure 4), and fitted it by the model (9). Here, the low-rank term and the element-wise sparse term are expected to capture the static background and the moving foreground, respectively. However, we can also rely on a natural assumption that a pixel segment having similar intensity values in an image tends to belong to the same object. Based on this side information, we adopt a segment-wise sparse term, where the PR matrix is constructed using a precomputed over-segmented image (Figure 5). We will show in Section 5.4 that the segment-wise sparse term captures the foreground more accurately than the element-wise sparse term.

Let us summarize the parameters of the SAMF model (7) as follows:

$$\Theta = \{\Theta_{\mathrm{A}}^{(s)}, \Theta_{\mathrm{B}}^{(s)}\}_{s=1}^{S}, \quad \text{where} \quad \Theta_{\mathrm{A}}^{(s)} = \{A^{(k,s)}\}_{k=1}^{K^{(s)}}, \quad \Theta_{\mathrm{B}}^{(s)} = \{B^{(k,s)}\}_{k=1}^{K^{(s)}}.$$

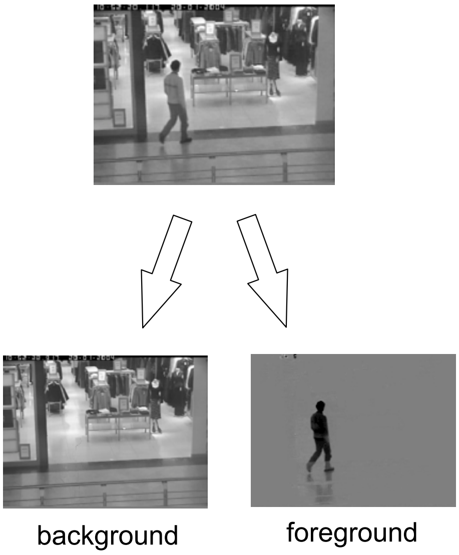


Figure 3: Foreground/background video separation task.

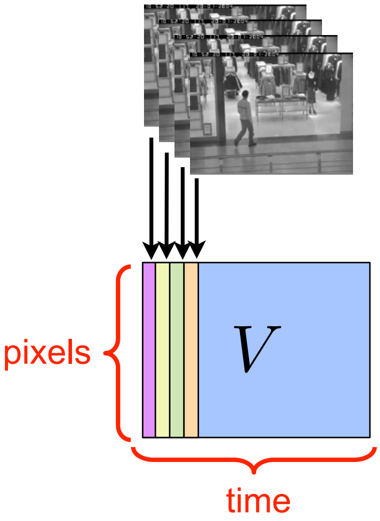


Figure 4: The observation matrix V is constructed by stacking all pixels in each frame into each column.

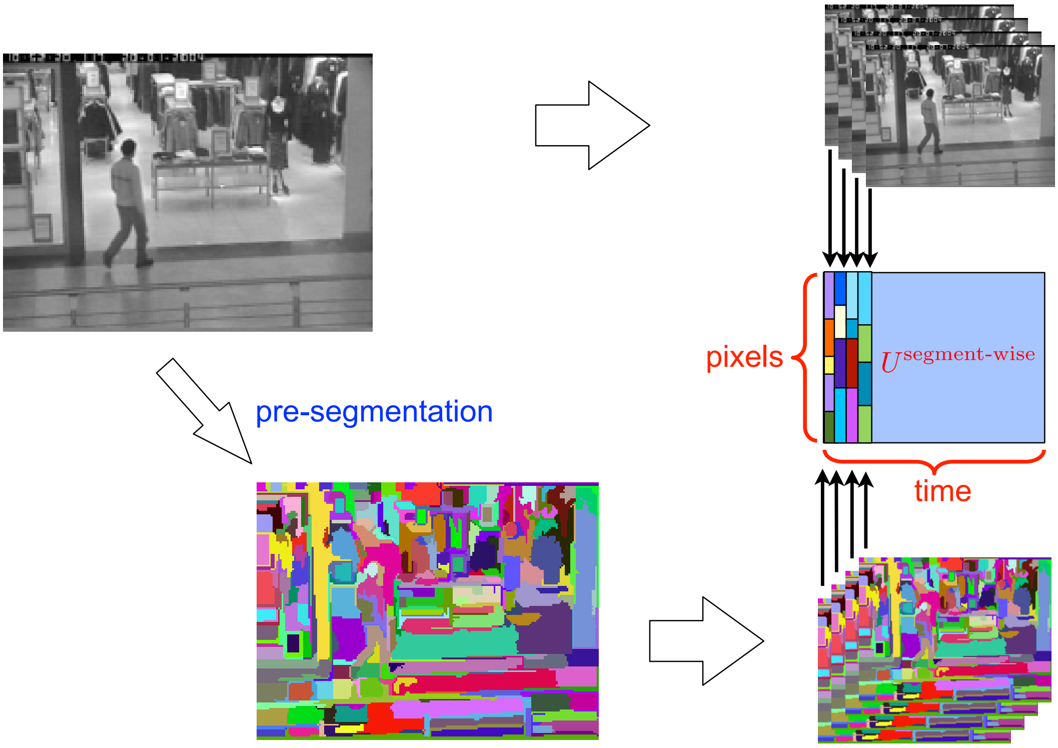


Figure 5: Construction of a segment-wise sparse term. The original frame is pre-segmented and the sparseness is induced in a segment-wise manner. Details are described in Section 5.4.

As in the probabilistic MF (Salakhutdinov and Mnih, 2008), we assume independent Gaussian noise and priors. Thus, the likelihood and the priors are written as

$$p(V|\Theta) \propto \exp\left(-\frac{1}{2\sigma^2}\left\|V - \sum_{s=1}^{S} U^{(s)}\right\|_{\mathrm{Fro}}^2\right), \tag{10}$$

$$p(\{\Theta_{\mathrm{A}}^{(s)}\}_{s=1}^{S}) \propto \exp\left(-\frac{1}{2}\sum_{s=1}^{S}\sum_{k=1}^{K^{(s)}} \mathrm{tr}\left(A^{(k,s)}C_A^{(k,s)-1}A^{(k,s)\top}\right)\right), \tag{11}$$

$$p(\{\Theta_{\mathrm{B}}^{(s)}\}_{s=1}^{S}) \propto \exp\left(-\frac{1}{2}\sum_{s=1}^{S}\sum_{k=1}^{K^{(s)}} \mathrm{tr}\left(B^{(k,s)}C_B^{(k,s)-1}B^{(k,s)\top}\right)\right), \tag{12}$$

where $\|\cdot\|_{\mathrm{Fro}}$ and $\mathrm{tr}(\cdot)$ denote the Frobenius norm and the trace of a matrix, respectively. We assume that the prior covariances of $A^{(k,s)}$ and $B^{(k,s)}$ are diagonal and positive-definite:

$$C_A^{(k,s)} = \mathrm{diag}(c_{a_1}^{(k,s)2}, \ldots, c_{a_H}^{(k,s)2}),$$
$$C_B^{(k,s)} = \mathrm{diag}(c_{b_1}^{(k,s)2}, \ldots, c_{b_H}^{(k,s)2}).$$

Without loss of generality, we assume that the diagonal entries of $C_A^{(k,s)}C_B^{(k,s)}$ are arranged in the non-increasing order, i.e., $c_{a_h}^{(k,s)}c_{b_h}^{(k,s)} \geq c_{a_{h'}}^{(k,s)}c_{b_{h'}}^{(k,s)}$ for any pair $h < h'$.

2.4 Variational Bayesian Approximation

The Bayes posterior is written as

$$p(\Theta|V) = \frac{p(V|\Theta)p(\Theta)}{p(V)}, \tag{13}$$

where $p(V) = \langle p(V|\Theta)\rangle_{p(\Theta)}$ is the marginal likelihood. Here, $\langle\cdot\rangle_p$ denotes the expectation over the distribution p. Since the Bayes posterior (13) for matrix factorization is computationally intractable, the variational Bayesian (VB) approximation was proposed (Bishop, 1999; Lim and Teh, 2007; Ilin and Raiko, 2010; Babacan et al., 2012).

Let $r(\Theta)$, or r for short, be a trial distribution. The following functional with respect to r is called the free energy:

$$F(r|V) = \left\langle \log \frac{r(\Theta)}{p(V|\Theta)p(\Theta)} \right\rangle_{r(\Theta)} = \left\langle \log \frac{r(\Theta)}{p(\Theta|V)} \right\rangle_{r(\Theta)} - \log p(V). \tag{14}$$

The first term is the Kullback-Leibler (KL) distance from the trial distribution to the Bayes posterior, and the second term is a constant. Therefore, minimizing the free energy (14) amounts to finding a distribution closest to the Bayes posterior in the sense of the KL distance. In the VB approximation, the free energy (14) is minimized over some restricted function space.

Following the standard VB procedure (Bishop, 1999; Lim and Teh, 2007; Babacan et al., 2012), we impose the following decomposability constraint on the posterior:

$$r(\Theta) = \prod_{s=1}^{S} r_{\mathrm{A}}^{(s)}(\Theta_{\mathrm{A}}^{(s)}) r_{\mathrm{B}}^{(s)}(\Theta_{\mathrm{B}}^{(s)}). \tag{15}$$

Under this constraint, it is easy to show that the VB posterior minimizing the free energy (14) is written as

$$\begin{aligned} r(\Theta) = \prod_{s=1}^{S} \prod_{k=1}^{K^{(s)}} \Bigg(& \prod_{m'=1}^{M'^{(k,s)}} \mathcal{N}_{H'^{(k,s)}}(\widetilde{\boldsymbol{a}}_{m'}^{(k,s)}; \widehat{\widetilde{\boldsymbol{a}}}_{m'}^{(k,s)}, \Sigma_{A}^{(k,s)}) \\ & \cdot \prod_{l'=1}^{L'^{(k,s)}} \mathcal{N}_{H'^{(k,s)}}(\widetilde{\boldsymbol{b}}_{l'}^{(k,s)}; \widehat{\widetilde{\boldsymbol{b}}}_{l'}^{(k,s)}, \Sigma_{B}^{(k,s)}) \Bigg), \end{aligned} \tag{16}$$

where $\mathcal{N}_d(\cdot; \boldsymbol{\mu}, \Sigma)$ denotes the d-dimensional Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance Σ.

3 Algorithm for SAMF

In this section, we first present a theorem that reduces a partial SAMF problem to the standard MF problem, which can be solved *analytically.* Then we derive an algorithm for the entire SAMF problem.

3.1 Key Theorem

Let us denote the mean of $U^{(s)}$, defined in Eq.(8), over the VB posterior by

$$\begin{aligned} \widehat{U}^{(s)} &= \langle U^{(s)} \rangle_{r_{\mathrm{A}}^{(s)}(\Theta_{A}^{(s)}) r_{\mathrm{B}}^{(s)}(\Theta_{B}^{(s)})} \\ &= G(\{\widehat{B}^{(k,s)} \widehat{A}^{(k,s)\top}\}_{k=1}^{K^{(s)}}; \mathcal{X}^{(s)}). \end{aligned} \tag{17}$$

Then we obtain the following theorem (its proof is given in Appendix A):

Theorem 1 *Given $\{\widehat{U}^{(s')}\}_{s' \neq s}$ and the noise variance σ^2, the VB posterior of $(\Theta_{A}^{(s)}, \Theta_{B}^{(s)}) = \{A^{(k,s)}, B^{(k,s)}\}_{k=1}^{K^{(s)}}$ coincides with the VB posterior of the following MF model:*

$$p(Z'^{(k,s)} | A^{(k,s)}, B^{(k,s)}) \propto \exp\left(-\frac{1}{2\sigma^2} \left\| Z'^{(k,s)} - B^{(k,s)} A^{(k,s)\top} \right\|_{Fro}^{2}\right), \tag{18}$$

$$p(A^{(k,s)}) \propto \exp\left(-\frac{1}{2} tr\left(A^{(k,s)} C_{A}^{(k,s)-1} A^{(k,s)\top}\right)\right), \tag{19}$$

$$p(B^{(k,s)}) \propto \exp\left(-\frac{1}{2} tr\left(B^{(k,s)} C_{B}^{(k,s)-1} B^{(k,s)\top}\right)\right), \tag{20}$$

for each $k = 1, \ldots, K^{(s)}$*. Here,* $Z'^{(k,s)} \in \mathbb{R}^{L'^{(k,s)} \times M'^{(k,s)}}$ *is defined as*

$$Z'^{(k,s)}_{l',m'} = Z^{(s)}_{\mathcal{X}^{(s)}(k,l',m')}, \text{ where } Z^{(s)} = V - \sum_{s' \neq s} \widehat{U}^{(s)}. \tag{21}$$

The left formula in Eq.(21) relates the entries of $Z^{(s)} \in \mathbb{R}^{L \times M}$ to the entries of $\{Z'^{(k,s)} \in \mathbb{R}^{L'^{(k,s)} \times M'^{(k,s)}}\}_{k=1}^{K^{(s)}}$ by using the map $\mathcal{X}^{(s)} : (k, l', m') \mapsto (l, m)$ (see Eq.(6) and Figure 1).

Theorem 1 states that a partial problem of SAMF—finding the posterior of $(A^{(k,s)}, B^{(k,s)})$ for each $k = 1, \ldots, K(s)$, given $\{\widehat{U}^{(s')}\}_{s' \neq s}$ and σ^2— can be solved in the same way as in the standard VBMF, to which the global analytic solution is available (Nakajima et al., 2013). Based on this theorem, we will propose a useful algorithm in the following subsections.

The noise variance σ^2 is also unknown in many applications. To estimate σ^2, we can use the following lemma (its proof is also included in Appendix A):

Lemma 1 *Given the VB posterior for* $\{\Theta_A^{(s)}, \Theta_B^{(s)}\}_{s=1}^S$*, the noise variance* σ^2 *minimizing the free energy* (14) *is given by*

$$\begin{aligned}\sigma^2 = \frac{1}{LM}\Bigg\{ \|V\|^2_{Fro} - 2\sum_{s=1}^{S} tr\left(\widehat{U}^{(s)\top}\left(V - \sum_{s'=s+1}^{S} \widehat{U}^{(s')}\right)\right) \\ + \sum_{s=1}^{S}\sum_{k=1}^{K^{(s)}} tr\Big((\widehat{A}^{(k,s)\top}\widehat{A}^{(k,s)} + M'^{(k,s)}\Sigma_A^{(k,s)}) \\ \cdot (\widehat{B}^{(k,s)\top}\widehat{B}^{(k,s)} + L'^{(k,s)}\Sigma_B^{(k,s)})\Big)\Bigg\}.\end{aligned} \tag{22}$$

3.2 Partial Analytic Solution

Theorem 1 allows us to use the results given in Nakajima et al. (2013), which provide the global analytic solution for VBMF. Although the free energy of VBMF is also non-convex, Nakajima et al. (2013) showed that the minimizers can be written as a reweighted singular value decomposition. This allows one to solve the minimization problem separately for each singular component, which facilitated the analysis. By finding all stationary points and calculating the free energy on them, they successfully obtained an analytic-form of the global VBMF solution.

Combining Theorem 1 above and Theorems 3–5 in Nakajima et al. (2013), we obtain the following corollaries:

Corollary 1 *Assume that* $L'^{(k,s)} \leq M'^{(k,s)}$ *for all* (k, s)*, and that* $\{\widehat{U}^{(s')}\}_{s' \neq s}$ *and the noise variance* σ^2 *are given. Let* $\gamma_h^{(k,s)}$ (≥ 0) *be the* h*-th largest singular value of* $Z'^{(k,s)}$*, and let* $\boldsymbol{\omega}_{a_h}^{(k,s)}$ *and* $\boldsymbol{\omega}_{b_h}^{(k,s)}$ *be the associated right and left singular vectors:*

$$Z'^{(k,s)} = \sum_{h=1}^{L'^{(k,s)}} \gamma_h^{(k,s)} \boldsymbol{\omega}_{b_h}^{(k,s)} \boldsymbol{\omega}_{a_h}^{(k,s)\top}. \tag{23}$$

Let $\widehat{\gamma}_h^{(k,s)\mathrm{second}}$ *be the* second *largest real solution of the following* quartic *equation with respect to* t*:*

$$f_h(t) := t^4 + \xi_3^{(k,s)} t^3 + \xi_2^{(k,s)} t^2 + \xi_1^{(k,s)} t + \xi_0^{(k,s)} = 0, \tag{24}$$

where the coefficients are defined by

$$\begin{aligned}
\xi_3^{(k,s)} &= \frac{(L'^{(k,s)} - M'^{(k,s)})^2 \gamma_h^{(k,s)}}{L'^{(k,s)} M'^{(k,s)}}, \\
\xi_2^{(k,s)} &= -\left(\xi_3 \gamma_h^{(k,s)} + \frac{(L'^{(k,s)2} + M'^{(k,s)2}) \eta_h^{(k,s)2}}{L'^{(k,s)} M'^{(k,s)}} + \frac{2\sigma^4}{c_{a_h}^{(k,s)2} c_{b_h}^{(k,s)2}} \right), \\
\xi_1^{(k,s)} &= \xi_3^{(k,s)} \sqrt{\xi_0^{(k,s)}}, \\
\xi_0^{(k,s)} &= \left(\eta_h^{(k,s)2} - \frac{\sigma^4}{c_{a_h}^{(k,s)2} c_{b_h}^{(k,s)2}} \right)^2, \\
\eta_h^{(k,s)2} &= \left(1 - \frac{\sigma^2 L'^{(k,s)}}{\gamma_h^{(k,s)2}} \right) \left(1 - \frac{\sigma^2 M'^{(k,s)}}{\gamma_h^{(k,s)2}} \right) \gamma_h^{(k,s)2}.
\end{aligned}$$

Let

$$\widetilde{\gamma}_h^{(k,s)} = \sqrt{\tau + \sqrt{\tau^2 - L'^{(k,s)} M'^{(k,s)} \sigma^4}}, \tag{25}$$

where

$$\tau = \frac{(L'^{(k,s)} + M'^{(k,s)})\sigma^2}{2} + \frac{\sigma^4}{2 c_{a_h}^{(k,s)2} c_{b_h}^{(k,s)2}}.$$

Then, the **global** *VB solution can be expressed as*

$$\begin{aligned}
\widehat{U}'^{(k,s)\mathrm{VB}} &= (\widehat{B}^{(k,s)} \widehat{A}^{(k,s)\top})^{\mathrm{VB}} = \sum_{h=1}^{H'^{(k,s)}} \widehat{\gamma}_h^{(k,s)\mathrm{VB}} \boldsymbol{\omega}_{b_h}^{(k,s)} \boldsymbol{\omega}_{a_h}^{(k,s)\top}, \\
&\text{where} \quad \widehat{\gamma}_h^{(k,s)\mathrm{VB}} = \begin{cases} \widehat{\gamma}_h^{(k,s)\mathrm{second}} & \text{if } \gamma_h^{(k,s)} > \widetilde{\gamma}_h^{(k,s)}, \\ 0 & \text{otherwise.} \end{cases}
\end{aligned} \tag{26}$$

Corollary 2 *Assume that* $L'^{(k,s)} \leq M'^{(k,s)}$ *for all* (k,s)*. Given* $\{\widehat{U}^{(s')}\}_{s' \neq s}$ *and the noise variance* σ^2*, the* **global empirical** *VB solution (where the hyperparameters* $\{C_A^{(k,s)}, C_B^{(k,s)}\}$ *are also estimated from observation) is given by*

$$\begin{aligned}
\widehat{U}'^{(k,s)\mathrm{EVB}} &= \sum_{h=1}^{H'^{(k,s)}} \widehat{\gamma}_h^{(k,s)\mathrm{EVB}} \boldsymbol{\omega}_{b_h}^{(k,s)} \boldsymbol{\omega}_{a_h}^{(k,s)\top}, \\
&\text{where} \quad \widehat{\gamma}_h^{(k,s)\mathrm{EVB}} = \begin{cases} \breve{\gamma}_h^{(k,s)\mathrm{VB}} & \text{if } \gamma_h^{(k,s)} > \underline{\gamma}_h^{(k,s)} \text{ and } \varDelta_h^{(k,s)} \leq 0, \\ 0 & \text{otherwise.} \end{cases}
\end{aligned} \tag{27}$$

Here,

$$\underline{\gamma}_h^{(k,s)} = (\sqrt{L'^{(k,s)}} + \sqrt{M'^{(k,s)}})\sigma, \tag{28}$$

$$\begin{aligned}\breve{c}_h^{(k,s)2} &= \frac{1}{2L'^{(k,s)}M'^{(k,s)}} \left(\gamma_h^{(k,s)2} - (L'^{(k,s)} + M'^{(k,s)})\sigma^2 \right.\\ &\left. \quad + \sqrt{\left(\gamma_h^{(k,s)2} - (L'^{(k,s)} + M'^{(k,s)})\sigma^2\right)^2 - 4L'^{(k,s)}M'^{(k,s)}\sigma^4}\right),\end{aligned} \tag{29}$$

$$\begin{aligned}\varDelta_h^{(k,s)} &= M'^{(k,s)} \log\left(\frac{\gamma_h^{(k,s)}}{M'^{(k,s)}\sigma^2}\breve{\gamma}_h^{(k,s)\mathrm{VB}} + 1\right)\\ &\qquad + L'^{(k,s)} \log\left(\frac{\gamma_h^{(k,s)}}{L'^{(k,s)}\sigma^2}\breve{\gamma}_h^{(k,s)\mathrm{VB}} + 1\right)\\ &\qquad + \frac{1}{\sigma^2}\left(-2\gamma_h^{(k,s)}\breve{\gamma}_h^{(k,s)\mathrm{VB}} + L'^{(k,s)}M'^{(k,s)}\breve{c}_h^{(k,s)2}\right),\end{aligned} \tag{30}$$

and $\breve{\gamma}_h^{(k,s)\mathrm{VB}}$ *is the VB solution for* $c_{a_h}^{(k,s)}c_{b_h}^{(k,s)} = \breve{c}_h^{(k,s)}$*.*

Corollary 3 *Assume that* $L'^{(k,s)} \leq M'^{(k,s)}$ *for all* (k,s)*. Given* $\{\widehat{U}^{(s')}\}_{s' \neq s}$ *and the noise variance* σ^2*, the VB posteriors are given by*

$$r_{\mathrm{A}^{(k,s)}}^{\mathrm{VB}}(A^{(k,s)}) = \prod_{h=1}^{H'^{(k,s)}} \mathcal{N}_{M'^{(k,s)}}(\boldsymbol{a}_h^{(k,s)}; \widehat{\boldsymbol{a}}_h^{(k,s)}, \sigma_{a_h}^{(k,s)2} I_{M'^{(k,s)}}),$$

$$r_{\mathrm{B}^{(k,s)}}^{\mathrm{VB}}(B^{(k,s)}) = \prod_{h=1}^{H'^{(k,s)}} \mathcal{N}_{L'^{(k,s)}}(\boldsymbol{b}_h^{(k,s)}; \widehat{\boldsymbol{b}}_h^{(k,s)}, \sigma_{b_h}^{(k,s)2} I_{L'^{(k,s)}}),$$

where, for $\widehat{\gamma}_h^{(k,s)\mathrm{VB}}$ *being the solution given by Corollary 1,*

$$\widehat{\boldsymbol{a}}_h^{(k,s)} = \pm\sqrt{\widehat{\gamma}_h^{(k,s)\mathrm{VB}}\widehat{\delta}_h^{(k,s)}} \cdot \boldsymbol{\omega}_{a_h}^{(k,s)}, \quad \widehat{\boldsymbol{b}}_h^{(k,s)} = \pm\sqrt{\widehat{\gamma}_h^{(k,s)\mathrm{VB}}\widehat{\delta}_h^{(k,s)-1}} \cdot \boldsymbol{\omega}_{b_h}^{(k,s)},$$

$$\begin{aligned}\sigma_{a_h}^{(k,s)2} &= \frac{1}{2M'^{(k,s)}(\widehat{\gamma}_h^{(k,s)\mathrm{VB}}\widehat{\delta}_h^{(k,s)-1} + \sigma^2 c_{a_h}^{(k,s)-2})}\\ &\quad \cdot \left\{ -\left(\widehat{\eta}_h^{(k,s)2} - \sigma^2(M'^{(k,s)} - L'^{(k,s)})\right)\right.\\ &\qquad\left. + \sqrt{(\widehat{\eta}_h^{(k,s)2} - \sigma^2(M'^{(k,s)} - L'^{(k,s)}))^2 + 4M'^{(k,s)}\sigma^2\widehat{\eta}_h^{(k,s)2}}\right\},\end{aligned}$$

$$\begin{aligned}\sigma_{b_h}^{(k,s)2} &= \frac{1}{2L'^{(k,s)}(\widehat{\gamma}_h^{(k,s)\mathrm{VB}}\widehat{\delta}_h^{(k,s)} + \sigma^2 c_{b_h}^{(k,s)-2})}\\ &\quad \cdot \left\{ -\left(\widehat{\eta}_h^{(k,s)2} + \sigma^2(M'^{(k,s)} - L'^{(k,s)})\right)\right.\end{aligned}$$

Algorithm 1 Mean update (MU) algorithm for (empirical) VB SAMF.

```
1: Initialization: $\widehat{U}^{(s)} \leftarrow 0_{(L,M)}$ for $s = 1, \ldots, S$, $\sigma^2 \leftarrow \|V\|^2_{\mathrm{Fro}}/(LM)$.
2: for $s = 1$ to $S$ do
3:     The (empirical) VB solution of $U'^{(k,s)} = B^{(k,s)}A^{(k,s)\top}$ for each $k = 1, \ldots, K^{(s)}$, given
       $\{\widehat{U}^{(s')}\}_{s' \neq s}$, is computed by Corollary 1 (Corollary 2).
4:     $\widehat{U}^{(s)} \leftarrow G(\{\widehat{B}^{(k,s)}\widehat{A}^{(k,s)\top}\}_{k=1}^{K^{(s)}}; \mathcal{X}^{(s)})$.
5: end for
6: $\sigma^2$ is estimated by Lemma 1, given the VB posterior on $\{\Theta_A^{(s)}, \Theta_B^{(s)}\}_{s=1}^{S}$ (computed by
   Corollary 3).
7: Repeat 2 to 6 until convergence.
```

$$+\sqrt{(\widehat{\eta}_h^{(k,s)2}+\sigma^2(M'^{(k,s)}-L'^{(k,s)}))^2+4L'^{(k,s)}\sigma^2\widehat{\eta}_h^{(k,s)2}}\Bigg\},$$

$$\widehat{\delta}_h^{(k,s)} = \frac{1}{2\sigma^2 M'^{(k,s)} c_{a_h}^{(k,s)-2}}\Bigg\{(M'^{(k,s)}-L'^{(k,s)})(\gamma_h^{(k,s)}-\widehat{\gamma}_h^{(k,s)\mathrm{VB}})$$

$$+\sqrt{(M'^{(k,s)}-L'^{(k,s)})^2(\gamma_h^{(k,s)}-\widehat{\gamma}_h^{(k,s)\mathrm{VB}})^2+\frac{4\sigma^4 L'^{(k,s)}M'^{(k,s)}}{c_{a_h}^{(k,s)2}c_{b_h}^{(k,s)2}}}\Bigg\},$$

$$\widehat{\eta}_h^{(k,s)2} = \begin{cases} \eta_h^{(k,s)2} & \textit{if } \gamma_h^{(k,s)} > \widetilde{\gamma}_h^{(k,s)}, \\ \frac{\sigma^4}{c_{a_h}^{(k,s)2}c_{b_h}^{(k,s)2}} & \textit{otherwise.} \end{cases}$$

Note that the corollaries above assume that $L'^{(k,s)} \leq M'^{(k,s)}$ for all (k,s). However, we can easily obtain the result for the case when $L'^{(k,s)} > M'^{(k,s)}$ by considering the transpose $\widehat{U}'^{(k,s)\top}$ of the solution. Also, we can always take the mapping $\mathcal{X}^{(s)}$ so that $L'^{(k,s)} \leq M'^{(k,s)}$ holds for all (k,s) without any practical restriction. This eases the implementation of the algorithm.

When σ^2 is known, Corollary 1 and Corollary 2 provide the global analytic solution of the *partial* problem, where the variables on which $\{\widehat{U}^{(s')}\}_{s' \neq s}$ depends are fixed. Note that they give the global analytic solution for single-term ($S = 1$) SAMF.

3.3 Mean Update Algorithm

Using Corollaries 1–3 and Lemma 1, we propose an algorithm for SAMF, which we call the *mean update* (MU) algorithm. We describe its pseudo-code in Algorithm 1, where $0_{(d_1,d_2)}$ denotes the $d_1 \times d_2$ matrix with all entries equal to zero. Note that, under the empirical Bayesian framework, all unknown parameters are estimated from observation, which allows inference without manual parameter tuning.

The MU algorithm is similar in spirit to the backfitting algorithm (Hastie and Tibshirani, 1986; D'Souza et al., 2004), where each additive term is updated to fit a dummy target. In the MU algorithm, $Z^{(s)}$ defined in Eq.(21) corresponds to the dummy target in the backfitting algorithm. Although each of the corollaries and the lemma above guarantee the global optimality for each step, the MU algorithm does not generally guarantee the simultaneous global optimality over the entire parameter space. Nevertheless, experimental results in Section 5 show that the MU algorithm performs very well in practice.

When Corollary 1 or Corollary 2 is applied in Step 3 of Algorithm 1, a singular value decomposition (23) of $Z'^{(k,s)}$, defined in Eq.(21), is required. However, for many practical SMF terms, including the row-wise, the column-wise, and the element-wise terms as well as the segment-wise term (which will be defined in Section 5.4), $Z'^{(k,s)} \in \mathbb{R}^{L'^{(k,s)} \times M'^{(k,s)}}$ is a vector or scalar, i.e., $L'^{(k,s)} = 1$ or $M'^{(k,s)} = 1$. In such cases, the singular value and the singular vectors are given simply by

$$\begin{aligned} \gamma_1^{(k.s)} &= \|Z'^{(k,s)}\|, \quad \boldsymbol{\omega}_{a_1}^{(k.s)} = Z'^{(k,s)}/\|Z'^{(k,s)}\|, \quad \boldsymbol{\omega}_{b_1}^{(k.s)} = 1 & \text{if } L'^{(k,s)} = 1, \\ \gamma_1^{(k.s)} &= \|Z'^{(k,s)}\|, \quad \boldsymbol{\omega}_{a_1}^{(k.s)} = 1, \quad \boldsymbol{\omega}_{b_1}^{(k.s)} = Z'^{(k,s)}/\|Z'^{(k,s)}\| & \text{if } M'^{(k,s)} = 1. \end{aligned}$$

4 Discussion

In this section, we first relate MIR to ARD. Then, we introduce the standard VB iteration for SAMF, which is used as a baseline in the experiments. After that, we discuss related previous work, and the limitation of the current work.

4.1 Relation between MIR and ARD

The MIR effect (Nakajima and Sugiyama, 2011) induced by *factorization* actually has a close connection to the *automatic relevance determination* (ARD) effect (Neal, 1996). Assume $C_A = I_H$, where I_d denotes the d-dimensional identity matrix, in the *plain* MF model (18)–(20) (here we omit the suffixes k and s for brevity), and consider the following transformation: $BA^\top \mapsto U \in \mathbb{R}^{L \times M}$. Then, the likelihood (18) and the prior (19) on A are rewritten as

$$p(Z'|U) \propto \exp\left(-\frac{1}{2\sigma^2}\|Z' - U\|_{\mathrm{Fro}}^2\right), \tag{31}$$

$$p(U|B) \propto \exp\left(-\frac{1}{2}\mathrm{tr}\left(U^\top (BB^\top)^\dagger U\right)\right), \tag{32}$$

where † denotes the Moore-Penrose generalized inverse of a matrix. The prior (20) on B is kept unchanged. $p(U|B)$ in Eq.(32) is so-called the ARD prior with the covariance hyperparameter $BB^\top \in \mathbb{R}^{L \times L}$. It is known that this induces the ARD effect, i.e., the *empirical* Bayesian procedure where the prior covariance hyperparameter $BB^\top$ is also estimated from observation induces strong regularization and sparsity (Neal, 1996); see also Efron and Morris (1973) for a simple Gaussian case.

In the current context, Eq.(32) induces low-rank sparsity on U if no restriction on $BB^\top$ is imposed. Similarly, we can show that $(\gamma_l^e)^2$ in Eq.(2), $(\gamma_m^d)^2$ in Eq.(3), and $E_{l,m}^2$ in Eq.(4) act as prior variances shared by the entries in $\widetilde{\boldsymbol{u}}_l \in \mathbb{R}^M$, $\boldsymbol{u}_m \in \mathbb{R}^L$, and $U_{l,m} \in \mathbb{R}$, respectively. This explains the mechanism how the factorization forms in Eqs.(2)–(4) induce row-wise, column-wise, and element-wise sparsity, respectively.

When we employ the SMF-term expression (5), MIR occurs in each partition. Therefore, partition-wise sparsity and low-rank sparsity in each partition is observed. Corollaries 1 and 2 theoretically support this fact: Small singular values are discarded by thresholding in Eqs.(26) and (27).

4.2 Standard VB Iteration

Following the standard procedure for the VB approximation (Bishop, 1999; Lim and Teh, 2007; Babacan et al., 2012), we can derive the following algorithm, which we call the *standard VB iteration*:

$$\widehat{A}^{(k,s)} = \sigma^{-2} Z'^{(k,s)\top} \widehat{B}^{(k,s)} \Sigma_A^{(k,s)}, \tag{33}$$

$$\Sigma_A^{(k,s)} = \sigma^2 \left(\widehat{B}^{(k,s)\top} \widehat{B}^{(k,s)} + L'^{(k,s)} \Sigma_B^{(k,s)} + \sigma^2 C_A^{(k,s)-1} \right)^{-1}, \tag{34}$$

$$\widehat{B}^{(k,s)} = \sigma^{-2} Z'^{(k,s)} \widehat{A}^{(k,s)} \Sigma_B^{(k,s)}, \tag{35}$$

$$\Sigma_B^{(k,s)} = \sigma^2 \left(\widehat{A}^{(k,s)\top} \widehat{A}^{(k,s)} + M'^{(k,s)} \Sigma_A^{(k,s)} + \sigma^2 C_B^{(k,s)-1} \right)^{-1}. \tag{36}$$

Iterating Eqs.(33)–(36) for each (k, s) in turn until convergence gives a local minimum of the free energy (14).

In the empirical Bayesian scenario, the hyperparameters $\{C_A^{(k,s)}, C_B^{(k,s)}\}_{k=1,s=1}^{K^{(s)}\,S}$ are also estimated from observations. The following update rules give a local minimum of the free energy:

$$c_{a_h}^{(k,s)2} = \|\widehat{\boldsymbol{a}}_h^{(k,s)}\|^2 / M'^{(k,s)} + (\Sigma_A^{(k,s)})_{hh}, \tag{37}$$

$$c_{b_h}^{(k,s)2} = \|\widehat{\boldsymbol{b}}_h^{(k,s)}\|^2 / L'^{(k,s)} + (\Sigma_B^{(k,s)})_{hh}. \tag{38}$$

When the noise variance σ^2 is unknown, it is estimated by Eq.(22) in each iteration.

The standard VB iteration is computationally efficient since only a single parameter in $\{\widehat{A}^{(k,s)}, \Sigma_A^{(k,s)}, \widehat{B}^{(k,s)}, \Sigma_B^{(k,s)}, c_{a_h}^{(k,s)2}, c_{b_h}^{(k,s)2}\}_{k=1,s=1}^{K^{(s)}\,S}$ is updated in each step. However, it is known that the standard VB iteration is prone to suffer from the local minima problem (Nakajima et al., 2013). On the other hand, although the MU algorithm also does not guarantee the global optimality as a whole, it simultaneously gives the global optimal solution for the set $\{\widehat{A}^{(k,s)}, \Sigma_A^{(k,s)}, \widehat{B}^{(k,s)}, \Sigma_B^{(k,s)}, c_{a_h}^{(k,s)2}, c_{b_h}^{(k,s)2}\}_{k=1}^{K^{(s)}}$ for each s in each step. In Section 5, we will experimentally show that the MU algorithm tends to give a better solution (i.e., with a smaller free energy) than the standard VB iteration.

4.3 Related Work

As widely known, traditional PCA is sensitive to outliers in data and generally fails in their presence. Robust PCA (Candès et al., 2011) was developed to cope with large outliers that are not modeled within the traditional PCA. Unlike methods based on robust statistics (Huber and Ronchetti, 2009; Fischler and Bolles, 1981; Torre and Black, 2003; Ke and Kanade, 2005; Gao, 2008; Luttinen et al., 2009; Lakshminarayanan et al., 2011), Candès et al. (2011) explicitly modeled the spiky noise with an additional element-wise sparse term (see Eq.(9)). This model can also be applied to applications where the task is to estimate the element-wise sparse term itself (as opposed to discarding it as noise). A typical such application is foreground/background video separation (Figure 3).

The original formulation of robust PCA is non-Bayesian, and the sparsity is induced by the ℓ_1-norm regularization. Although its solution can be efficiently obtained via the augmented Lagrange multiplier (ALM) method (Lin et al., 2009), there are unknown algorithmic parameters that should be carefully tuned to obtain its best performance. Employing a Bayesian formulation addresses this issue: A sampling-based method (Ding et al., 2011) and a VB method (Babacan et al., 2012) were proposed, where all unknown parameters are estimated from the observation.

Babacan et al. (2012) conducted an extensive experimental comparison between their VB method, called a VB robust PCA, and other methods. They reported that the ALM method (Lin et al., 2009) requires careful tuning of its algorithmic parameters, and the Bayesian sampling method (Ding et al., 2011) has high computational complexity that can be prohibitive in large-scale applications. Compared to these methods, the VB robust PCA is favorable both in terms of computational complexity and estimation performance.

Our SAMF framework contains the robust PCA model as a special case where the observed matrix is modeled as the sum of a low-rank and an element-wise sparse terms. The VB algorithm used in Babacan et al. (2012) is the same as the standard VB iteration introduced in Section 4.2, except a slight difference in the hyperprior setting. Accordingly, our proposal in this paper is an extension of the VB robust PCA in two ways—more variation in sparsity with different types of factorization and higher accuracy with the MU algorithm. In Section 5, we experimentally show advantages of these extensions. In our experiment, we use a SAMF counterpart of the VB robust PCA, named 'LE'-SAMF in Section 5.1, with the standard VB iteration as a baseline method for comparison.

Group LASSO (Yuan and Lin, 2006) also provides a framework for arbitrary sparsity design, where the sparsity is induced by the ℓ_1-regularization. Although the convexity of the group LASSO problem is attractive, it typically requires careful tuning of regularization parameters, as the ALM method for robust PCA. On the other hand, group-sparsity is induced by *model-induced regularization* in SAMF, and all unknown parameters can be estimated, based on the Bayesian framework.

Another typical application of MF is collaborative filtering, where the observed matrix has missing entries. Fitting the observed entries with a low-rank matrix enables us to predict the missing entries. Convex optimization methods with the *trace-norm* penalty (i.e., singular values are regularized by the ℓ_1-penalty) have been extensively studied (Srebro et al., 2005; Rennie and Srebro, 2005; Cai et al., 2010; Ji and Ye, 2009; Tomioka et al., 2010).

Bayesian approaches to MF have also been actively explored. A *maximum a posteriori* (MAP) estimation, which computes the mode of the posterior distributions, was shown to be equivalent to the ℓ_1-MF when Gaussian priors are imposed on factorized matrices (Srebro et al., 2005). Salakhutdinov and Mnih (2008) applied the Markov chain Monte Carlo method to MF for the fully-Bayesian treatment. The VB approximation (Attias, 1999; Bishop, 2006) has also been applied to MF (Bishop, 1999; Lim and Teh, 2007; Ilin and Raiko, 2010), and it was shown to perform well in experiments. Its theoretical properties, including the *model-induced regularization*, have been investigated in Nakajima and Sugiyama (2011).

4.4 Limitations of SAMF and MU Algorithm

Here, we note the limitations of SAMF and the MU algorithm. First, in the current formulation, each SMF term is not allowed to have overlapping groups. This excludes important applications, e.g., simultaneous feature and sample selection problems (Jacob et al., 2009). Second, the MU algorithm cannot be applied when the observed matrix has missing entries, although SAMF itself still works with the standard VB iteration. This is because the global analytic solution, on which the MU algorithm relies, holds only for the fully-observed case. Third, we assume the Gaussian distribution for the dense noise ($\mathcal{E}$ in Eq.(7)), which may not be appropriate for, e.g., binary observations. Variational techniques for non-conjugate likelihoods, such as the one used in Seeger and Bouchard (2012), are required to extend SAMF to more general noise distributions. Fourth, we rely on the VB inference so far, and have not known if the fully-Bayesian treatment with additional hyperpriors can improve the performance. Overcoming some of the limitations described above is a promising future work.

5 Experimental Results

In this section, we first experimentally compare the performance of the MU algorithm and the standard VB iteration. Then, we test the model selection ability of SAMF, based on the free energy comparison. After that, we demonstrate the usefulness of the flexibility of SAMF on benchmark datasets and in a real-world application.

5.1 Mean Update vs. Standard VB

We compare the algorithms under the following model:

$$V = U^{\mathrm{LRCE}} + \mathcal{E},$$

where

$$U^{\mathrm{LRCE}} = \sum_{s=1}^{4} U^{(s)} = U^{\text{low-rank}} + U^{\text{row}} + U^{\text{column}} + U^{\text{element}}. \tag{39}$$

Here, 'LRCE' stands for the sum of the Low-rank, Row-wise, Column-wise, and Element-wise terms, each of which is defined in Eqs.(1)–(4). We call this model 'LRCE'-SAMF. As explained in Section 2.3, 'LRCE' -SAMF may be used to separate the clean signal $U^{\text{low-rank}}$ from a possible row-wise sparse component (constantly broken sensors), a column-wise sparse component (accidental disturbances affecting all sensors), and an element-wise sparse component (randomly distributed spiky noise). We also evaluate 'LCE'-SAMF, 'LRE'-SAMF, and 'LE'-SAMF, which can be regarded as generalizations of robust PCA (Candès et al., 2011; Ding et al., 2011; Babacan et al., 2012). Note that 'LE'-SAMF corresponds to an SAMF counterpart of robust PCA.

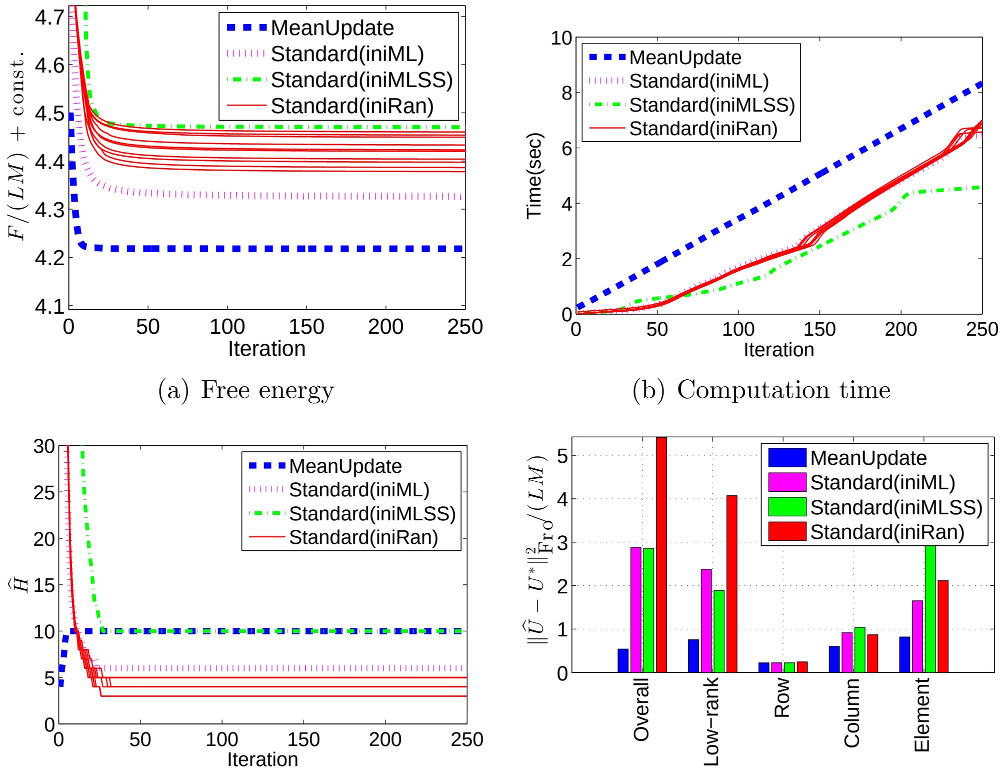
First, we conducted an experiment with artificial data. We assume the empirical VB scenario with unknown noise variance, i.e., the hyperparameters $\{C_A^{(k,s)}, C_B^{(k,s)}\}_{k=1,s=1}^{K^{(s)}\,S}$ and the noise variance σ^2 are also estimated from observations. We use the full-rank model ($H = \min(L, M)$) for the low-rank term $U^{\text{low-rank}}$, and expect the MIR effect to find the true rank of $U^{\text{low-rank}}$, as well as the non-zero entries in $U^{\text{row}}, U^{\text{column}}$, and U^{element}.

We created an artificial dataset with the data matrix size $L = 40$ and $M = 100$, and the rank $H^* = 10$ for a *true* low-rank matrix $U^{\text{low-rank}*} = B^* A^{*\top}$. Each entry in $A^* \in \mathbb{R}^{M \times H^*}$ and $B^* \in \mathbb{R}^{L \times H^*}$ was drawn from $\mathcal{N}_1(0, 1)$. A *true* row-wise (column-wise) part $U^{\text{row}*}$ ($U^{\text{column}*}$) was created by first randomly selecting ρL rows (ρM columns) for $\rho = 0.05$, and then adding a noise subject to $\mathcal{N}_M(\mathbf{0}, \zeta I_M)$ ($\mathcal{N}_L(\mathbf{0}, \zeta I_L)$) for $\zeta = 100$ to each of the selected rows (columns). A *true* element-wise part $U^{\text{element}*}$ was similarly created by first selecting ρLM entries, and then adding a noise subject to $\mathcal{N}_1(0, \zeta)$ to each of the selected entries. Finally, an observed matrix V was created by adding a noise subject to $\mathcal{N}_1(0, 1)$ to each entry of the sum $U^{\text{LRCE}*}$ of the above four *true* matrices.

It is known that the standard VB iteration (reviewed in Section 4.2) is known to be sensitive to initialization (Nakajima et al., 2013). We set the initial values in the following way: The mean parameters $\{\widehat{A}^{(k,s)}, \widehat{B}^{(k,s)}\}_{k=1,s=1}^{K^{(s)}\,S}$ were randomly created so that each entry follows $\mathcal{N}_1(0, 1)$. The covariances $\{\Sigma_A^{(k,s)}, \Sigma_B^{(k,s)}\}_{k=1,s=1}^{K^{(s)}\,S}$ and the hyperparameters $\{C_A^{(k,s)}, C_B^{(k,s)}\}_{k=1,s=1}^{K^{(s)}\,S}$ were set to be identity. The initial noise variance was set to $\sigma^2 = 1$. Note that we rescaled V so that $\|V\|^2_{\text{Fro}}/(LM) = 1$, before starting iteration. We ran the standard VB algorithm 10 times, starting from different initial points, and each trial is plotted by a solid line (labeled as 'Standard(iniRan)') in Figure 6.

Initialization for the MU algorithm (described in Algorithm 1) is simple: We just set $\widehat{U}^{(s)} = 0_{(L,M)}$ for $s = 1, \ldots, S$, and $\sigma^2 = 1$. Initialization of all other variables is not needed. Furthermore, we empirically observed that the initial value for σ^2 does not affect the result much, unless it is too small. Note that, in the MU algorithm, initializing σ^2 to a large value is not harmful, because it is set to an adequate value after the first iteration with the mean parameters kept $\widehat{U}^{(s)} = 0_{(L,M)}$. The result with the MU algorithm is plotted by the dashed line in Figure 6.

Figures 6(a)–6(c) show the free energy, the computation time, and the estimated rank, respectively, over iterations, and Figure 6(d) shows the reconstruction errors after 250 iterations. The reconstruction errors consist of the *overall* error $\|\widehat{U}^{\text{LRCE}} - U^{\text{LRCE}*}\|_{\text{Fro}}/(LM)$, and the four component-wise errors $\|\widehat{U}^{(s)} - U^{(s)*}\|_{\text{Fro}}/(LM)$. The graphs show that the MU algorithm, whose iteration is computationally slightly more expensive than the standard VB iteration, immediately converges to a local minimum with the free energy substan-



(a) Free energy

(b) Computation time

(c) Estimated rank

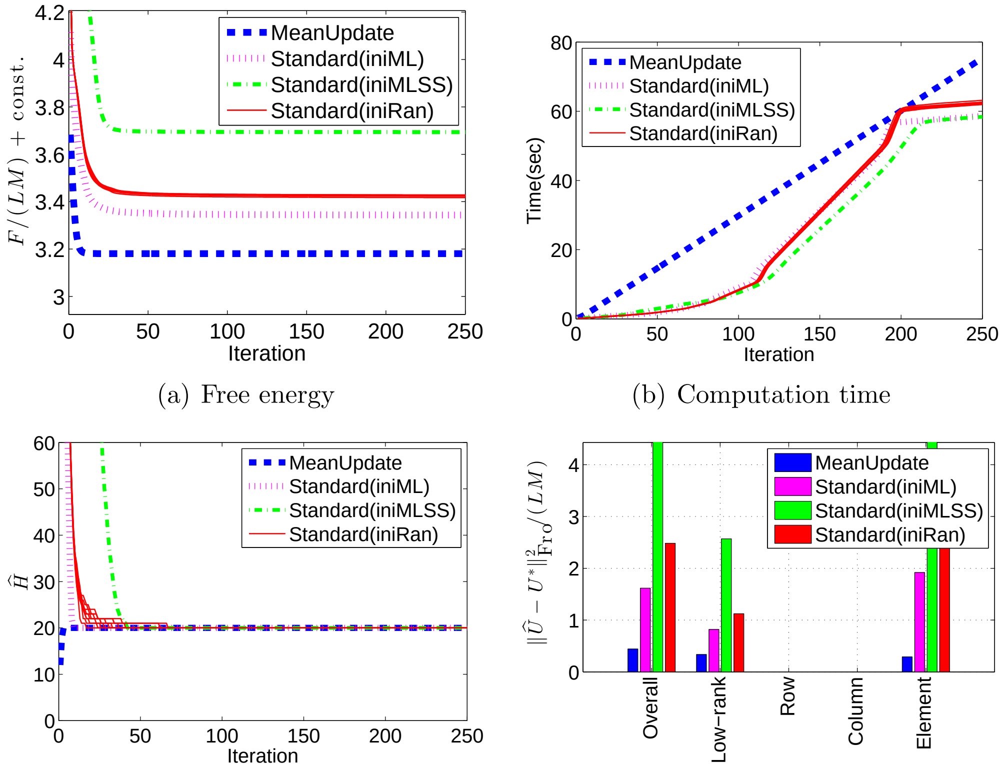
(d) Reconstruction error

Figure 6: Experimental results with 'LRCE'-SAMF for an artificial dataset ($L = 40, M = 100, H^* = 10, \rho = 0.05$).

tially lower than the standard VB iteration. The estimated rank agrees with the true rank $\widehat{H} = H^* = 10$, while all 10 trials of the standard VB iteration failed to estimate the true rank. It is also observed that the MU algorithm well reconstructs each of the four terms.

We can slightly improve the performance of the standard VB iteration by adopting different initialization schemes. The line labeled as 'Standard(iniML)' in Figure 6 indicates the maximum likelihood (ML) initialization, i.e, $(\widehat{\boldsymbol{a}}_h^{(k,s)}, \widehat{\boldsymbol{b}}_h^{(k,s)}) = (\gamma_h^{(k,s)1/2}\boldsymbol{\omega}_{a_h}^{(k,s)}, \gamma_h^{(k,s)1/2}\boldsymbol{\omega}_{b_h}^{(k,s)})$. Here, $\gamma_h^{(k,s)}$ is the h-th largest singular value of the (k,s)-th PR matrix $V'^{(k,s)}$ of V (such that $V'^{(k,s)}_{l',m'} = V_{\mathcal{X}^{(s)}(k,l',m')}$), and $\boldsymbol{\omega}_{a_h}^{(k,s)}$ and $\boldsymbol{\omega}_{b_h}^{(k,s)}$ are the associated right and left singular vectors. Also, we empirically found that starting from small σ^2 alleviates the local minima problem. The line labeled as 'Standard(iniMLSS)' indicates the ML initialization with $\sigma^2 = 0.0001$. We can see that this scheme successfully recovered the true rank. However, the free energy and the reconstruction error are still substantially worse than the MU algorithm.

Figure 7 shows results with 'LE'-SAMF when $L = 100$, $M = 300$, $H^* = 20$, and $\rho = 0.1$. We see that the MU algorithm compares favorably with the standard VB



(a) Free energy

(b) Computation time

(c) Estimated rank

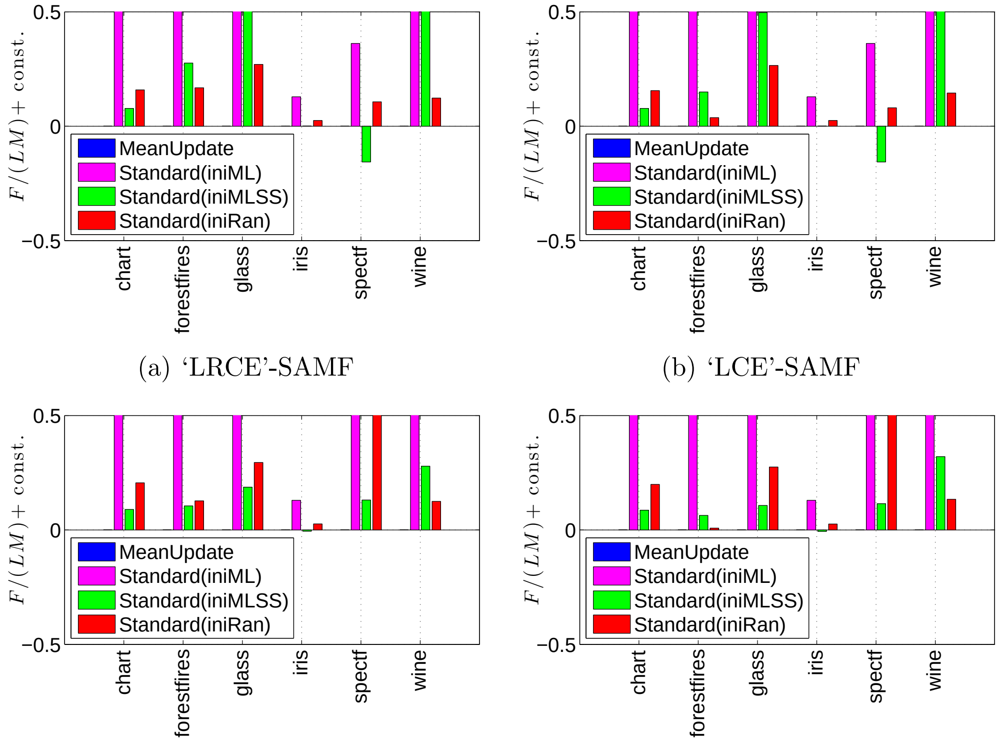
(d) Reconstruction error

Figure 7: Experimental results with 'LE'-SAMF for an artificial dataset $(L = 100, M = 300, H^* = 20, \rho = 0.1)$.

iteration. We have also tested various SAMF models including 'LCE'-SAMF, 'LRE'-SAMF, and 'LE'-SAMF under different settings for L, M, H^*, and ρ, and empirically found that the MU algorithm generally gives a better solution with lower free energy and smaller reconstruction errors than the standard VB iteration.

Next, we conducted experiments with benchmark data. Since we do not know the *true* model of these data, we only focus on the achieved free energy, which directly indicates the approximation accuracy to the Bayes posterior (see Section 2.4). To this end, we simply fitted SAMF models to benchmark datasets by the MU algorithm and the standard VB iteration, and plotted the obtained free energy.

Figure 8 shows the free energy after convergence in 'LRCE'-SAMF, 'LCE'-SAMF, 'LRE'-SAMF, and 'LE'-SAMF on several datasets from the *UCI repository* (Asuncion and Newman, 2007). For better comparison, a constant is added to the obtained free energy, so that the value of the MU algorithm is zero. We can see a clear advantage of the MU algorithm over the standard VB iteration.



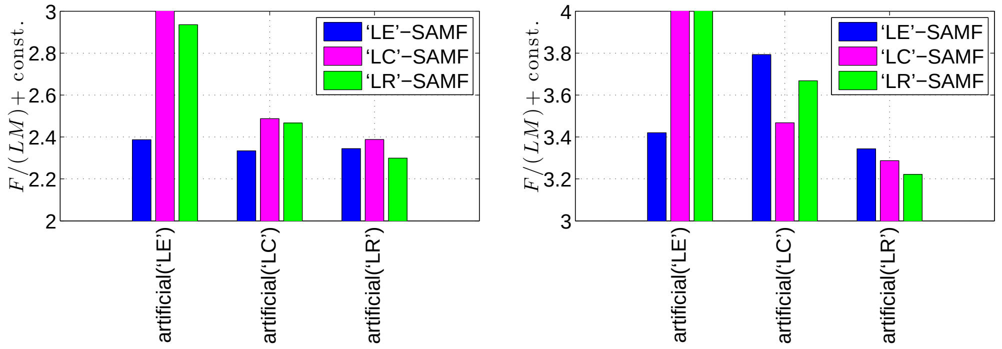
(a) 'LRCE'-SAMF

(b) 'LCE'-SAMF

(c) 'LRE'-SAMF

(d) 'LE'-SAMF

Figure 8: Free energy on benchmark data.



(a) $\zeta = 100$

(b) $\zeta = 100 \cdot LM$

Figure 9: Free energy comparison for model selection.

5.2 Model Selection Ability

We tested the model selection ability of SAMF, by checking if the 'LE'-SAMF, 'LC'-SAMF, and 'LR'-SAMF models give the lowest free energy for the artificial data created from the corresponding models, respectively.

We created artificial('LE') data from a *true* 'LE'-SAMF with $L = 150$, $M = 200$, $H^* = 20$, $\rho = 0.1$, and $\zeta = 100$. Likewise, we created artificial('LC') and artificial('LR') data from *true* 'LC'-SAMF and 'LR'-SAMF models, respectively. Then, we applied 'LE'-SAMF, 'LC'-SAMF, and 'LR'-SAMF models to those artificial data, and plotted the obtained free energies in Figure 9(a). If the model selection is successful, 'LE'-SAMF should result in the lowest free energy than the other two models on the artificial('LE') data. The same applies to 'LC'-SAMF and 'LR'-SAMF, respectively.

We see in Figure 9(a) that the model selection is successful on the artificial('LE') data and the artificial('LR') data, but not on the artificial('LC') data. In our investigation, we observed that SAMF sometimes mixes up the column-wise term with the low-rank term. We expect that a column-wise noise should be captured by the column-wise term. However, it can also be captured by either of the low-rank and the element-wise terms at the expense of a small loss of degrees of freedom (i.e., the problem is nearly ill-posed). The Bayesian regularization should choose the column-wise term, because it uses the lowest degrees of freedom to capture the column-wise noise. We suspect that the difference in regularization between the low-rank and the column-wise terms is too small for stable model selection against the disturbance by random noise and the existence of local minima. We also conducted the same experiment, of which the result is shown in Figure 9(b), with stronger sparse noise with $\zeta = 100LM$. In this case, the model selection is successful on all three artificial data. Further investigation on handling these nearly ill-posed cases is left as future work.

5.3 Robustness against Simulated Sparse Noise

Here, we experimentally show the usefulness of the SAMF extension beyond the robust PCA with simulated sparse noise.

Datasets from the UCI repository consist of M samples with L dimensions. We simulated sparse noise that contaminates a small number of measurements over the samples. We also simulated some accidents that cause simultaneous contamination in all the measurements of a small number of samples. As explained in Section 2.3, the SAMF model can capture the former type of noise by the element-wise sparse term, and the latter type of noise by the column-wise sparse term.

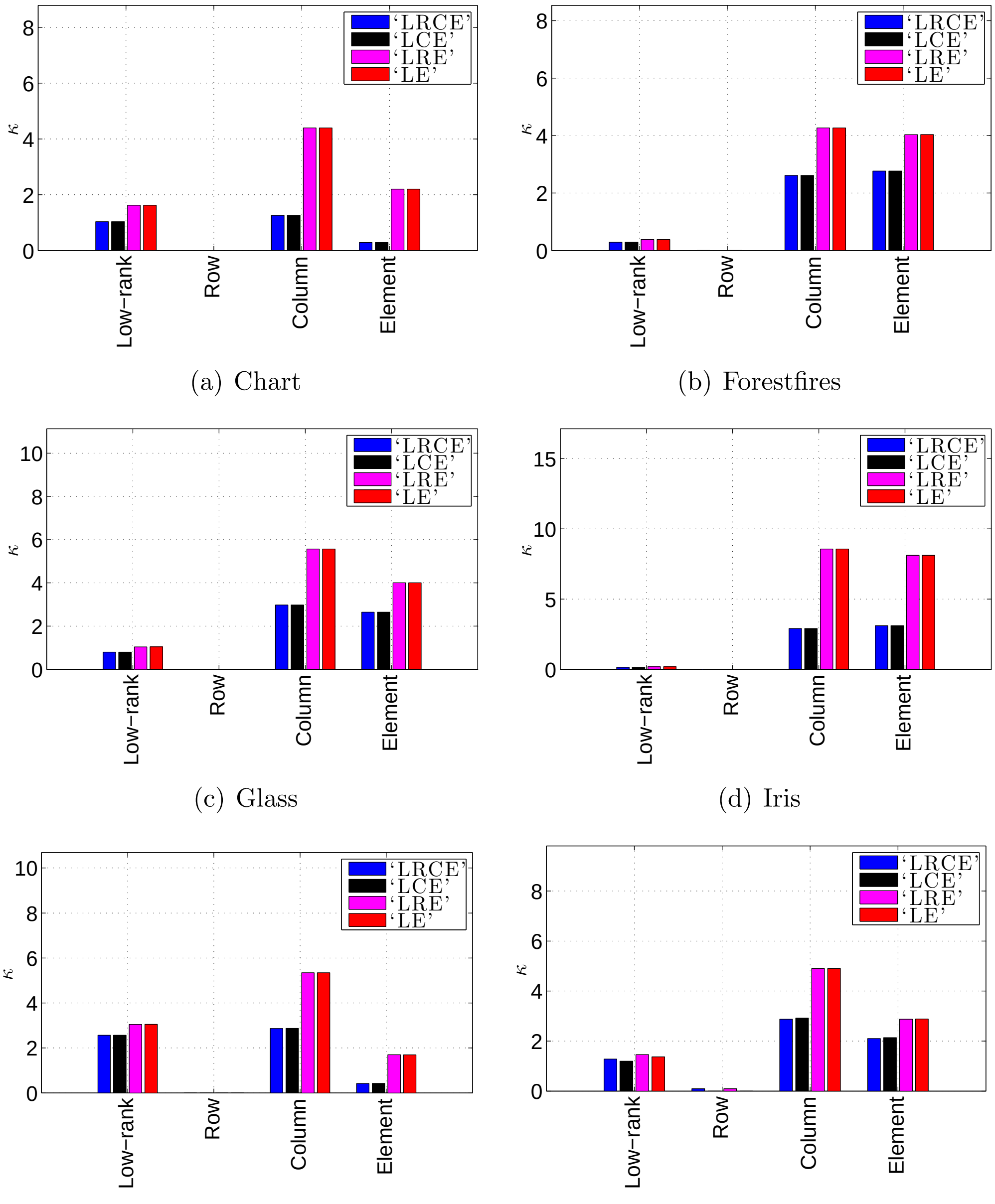
We created semi-artificial data in the following procedure. We first rescaled the benchmark data V^{org} so that $\|V^{\mathrm{org}}\|_{\mathrm{Fro}}^2/(LM) = 1$. Then, artificial *true* sparse noise components, $U^{\mathrm{column}*}$ and $U^{\mathrm{element}*}$, were created in the same way as in Section 5.1 with $\rho = 0.05$, and added to V^{org}, i.e.,

$$V^{\mathrm{sim}} = V^{\mathrm{org}} + U^{\mathrm{column}*} + U^{\mathrm{element}*}.$$

Since we do not know the *true* model of the original benchmark data, we focus on robustness against the simulated sparse noise. For the column-wise sparse term, we evaluate the following value:

$$\kappa^{\mathrm{column}} = \|(\widehat{U}^{\mathrm{column}+} - \widehat{U}^{\mathrm{column}-}) - U^{\mathrm{column}*}\|_{\mathrm{Fro}}^2/(LM),$$

where $\widehat{U}^{\mathrm{column}+}$ is the column-wise sparse term estimated from the simulated data V^{sim}, and $\widehat{U}^{\mathrm{column}-}$ is the column-wise sparse term estimated from the original data V^{org}. If a



(a) Chart

(b) Forestfires

(c) Glass

(d) Iris

(e) Spectf

(f) Wine

Figure 10: Robustness against simulated sparse noise on benchmark data. Smaller κ indicates better performance.

SAMF model perfectly captures the simulated *true* column-wise sparse noise $U^{\text{column}*}$ with its column-wise sparse term, then $\kappa^{\text{column}} = 0$ because the estimated column-wise sparse term is increased by the simulated noise, i.e., $\widehat{U}^{\text{column}+} - \widehat{U}^{\text{column}-} = U^{\text{column}*}$. Therefore, smaller κ^{column} is expected to indicate higher robustness of the model against the sparse noise. κ^{element} is calculated in the same way, and $\kappa^{\text{low-rank}}$ and $\kappa^{\text{row-wise}}$ are calculated without the simulated noise term, i.e., $U^{\text{low-rank}*} = U^{\text{row-wise}*} = 0_{(L,M)}$.

Figure 10 shows the values of κ averaged over 10 trials with randomly created sparse noise. As expected, the SAMF models (‘LRCE’-SAMF and ‘LCE’-SAMF) having the column-wise sparse term are more reliable than the others (‘LRE’-SAMF and ‘LE’-SAMF).

5.4 Real-world Application

Finally, we demonstrate the usefulness of the flexibility of SAMF in a foreground (FG)/background (BG) video separation problem (Figure 3). Candès et al. (2011) formed the observed matrix V by stacking all pixels in each frame into each column (Figure 4), and applied robust PCA (with ‘LE’-terms)—the low-rank term captures the *static* BG and the element-wise (or pixel-wise) term captures the *moving* FG, e.g., people walking through. As discussed in Section 4.3, SAMF is an extension of the VB robust PCA (Babacan et al., 2012), which is the current state-of-the-art. We use ‘LE’-SAMF,

$$V = U^{\text{low-rank}} + U^{\text{element}} + \mathcal{E},$$

which is conceptually the same as the VB robust PCA, as a baseline method for comparison.

The SAMF framework enables a fine-tuned design for the FG term. Assuming that pixels in an image segment with similar intensity values tend to share the same label (i.e., FG or BG), we formed a segment-wise sparse SMF term: $U'^{(k)}$ for each k is a column vector consisting of all pixels in each segment. We produced an over-segmented image from each frame by using the efficient graph-based segmentation (EGS) algorithm (Felzenszwalb and Huttenlocher, 2004), and substituted the segment-wise sparse term for the FG term (see Figure 5):

$$V = U^{\text{low-rank}} + U^{\text{segment}} + \mathcal{E}.$$

We call this method *segmentation-based SAMF* (sSAMF). Note that EGS is computationally very efficient: It takes less than 0.05 sec on a usual laptop to segment a 192×144 grey image. EGS has several tuning parameters, and the obtained segmentation is sensitive to some of them. However, we confirmed that sSAMF performs similarly with visually different segmentations obtained over a wide range of tuning parameters (see detailed information below on the segmentation algorithm). Therefore, careful parameter tuning of EGS is not necessary for our purpose.

We compared sSAMF with ‘LE’-SAMF on the ‘WalkByShop1front’ video from the *Caviar dataset.*[1] Thanks to the Bayesian framework, all unknown parameters (except the ones for segmentation) are estimated automatically with no manual parameter tuning. For both models (‘LE’-SAMF and sSAMF), we used the MU algorithm, which has been shown in Section 5.1 to be practically more reliable than the standard VB iteration. The original video consists of 2360 frames, each of which is a color image with 384×288 pixels. We resized each image into 192×144 pixels, averaged over the color channels, and sub-sampled every 15 frames (the frame IDs are $0, 15, 30, \ldots, 2355$). Thus, V is of the size

[1] `http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/`

of 27684 (pixels) × 158 (frames). We evaluated 'LE'-SAMF and sSAMF on this video, and found that both models perform well (although 'LE'-SAMF failed in a few frames).

To contrast the methods more clearly, we created a more *difficult* video by sub-sampling every 5 frames from 1501 to 2000 (the frame IDs are $1501, 1506, \ldots, 1996$ and V is of the size of 27684 (pixels) × 100 (frames)). Since more people walked through in this period, BG estimation is more challenging. The result is shown in Figure 11.

Figure 11(a) shows an original frame. This is a difficult snap shot, because a person stayed at a same position for a while, which confuses separation. Figures 11(c) and 11(d) show the BG and the FG terms obtained by 'LE'-SAMF, respectively. We can see that 'LE'-SAMF failed to separate the person from BG (the person is partly captured in the BG term). On the other hand, Figures 11(e) and 11(f) show the BG and the FG terms obtained by sSAMF based on the segmented image shown in Figure 11(b). We can see that sSAMF successfully separated the person from BG in this difficult frame. A careful look at the legs of the person makes us understand how segmentation helps separation—the legs form a single segment (light blue colored) in Figure 11(b), and the segment-wise sparse term (Figure 11(f)) captured all pixels on the legs, while the pixel-wise sparse term (Figure 11(d)) captured only a part of those pixels.

We observed that, in all frames of the *difficult* video, as well as the *easier* one, sSAMF gave good separation, while 'LE'-SAMF failed in several frames (see movies provided as Online Resource).

For reference, we applied the convex optimization approach (Candès et al., 2011), which solves the minimization problem

$$\min_{U,E} \|U^{\mathrm{BG}}\|_{\mathrm{Tr}} + \lambda \|U^{\mathrm{FG}}\|_1 \text{ s.t. } V = U^{\mathrm{BG}} + U^{\mathrm{FG}},$$

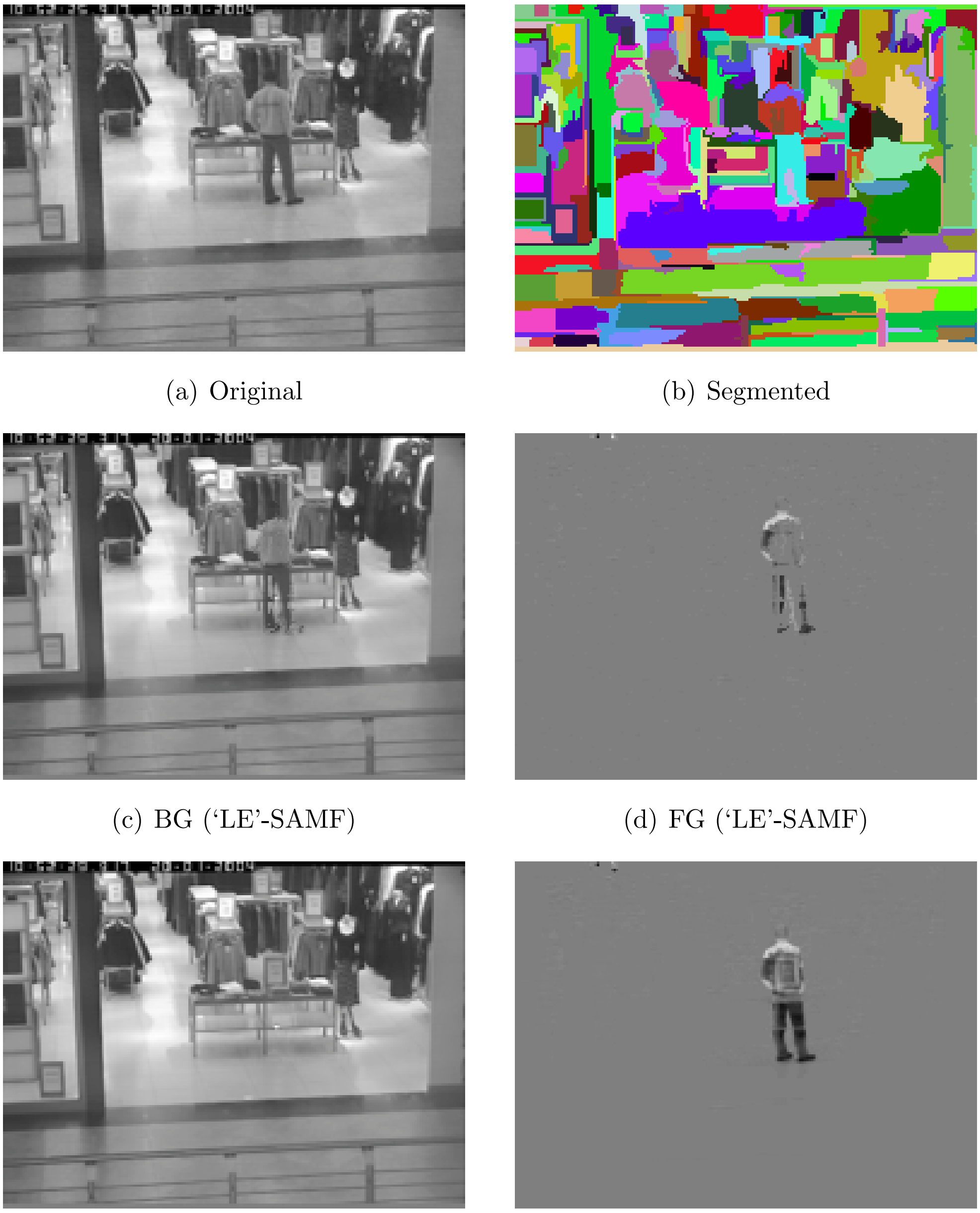
where $\|\cdot\|_{\mathrm{Tr}}$ and $\|\cdot\|_1$ denote the trace norm and the ℓ_1-norm of a matrix, respectively, by the inexact ALM algorithm (Lin et al., 2009). Figure 12 shows the obtained BG and FG terms of the same frame as in Figure 11 with $\lambda = 0.001, 0.005, 0.025$. We see that the performance strongly depends on the parameter value of λ, and that sSAMF gives an almost identical result (bottom row in Figure 11) to the best ALM result with $\lambda = 0.005$ (middle row in Figure 12) without any manual parameter tuning.

Below, we give detailed information on the segmentation algorithm, the computation time, and Online Resource.

Segmentation Algorithm

For the efficient graph-based segmentation (EGS) algorithm (Felzenszwalb and Huttenlocher, 2004), we used the code publicly available from the authors' homepage.[2] EGS has three tuning parameters: *sigma*, the smoothing parameter; k, the threshold parameter; *minc*, minimum segment size. Among them, k dominantly determines the typical size of segments (larger k leads to larger segments). To obtain over-segmented images for sSAMF in our experiment, we chose $k = 50$, and the other parameters are set to $sigma = 0.5$ and

[2] `http://www.cs.brown.edu/~pff/`



(a) Original

(b) Segmented

(c) BG (‘LE’-SAMF)

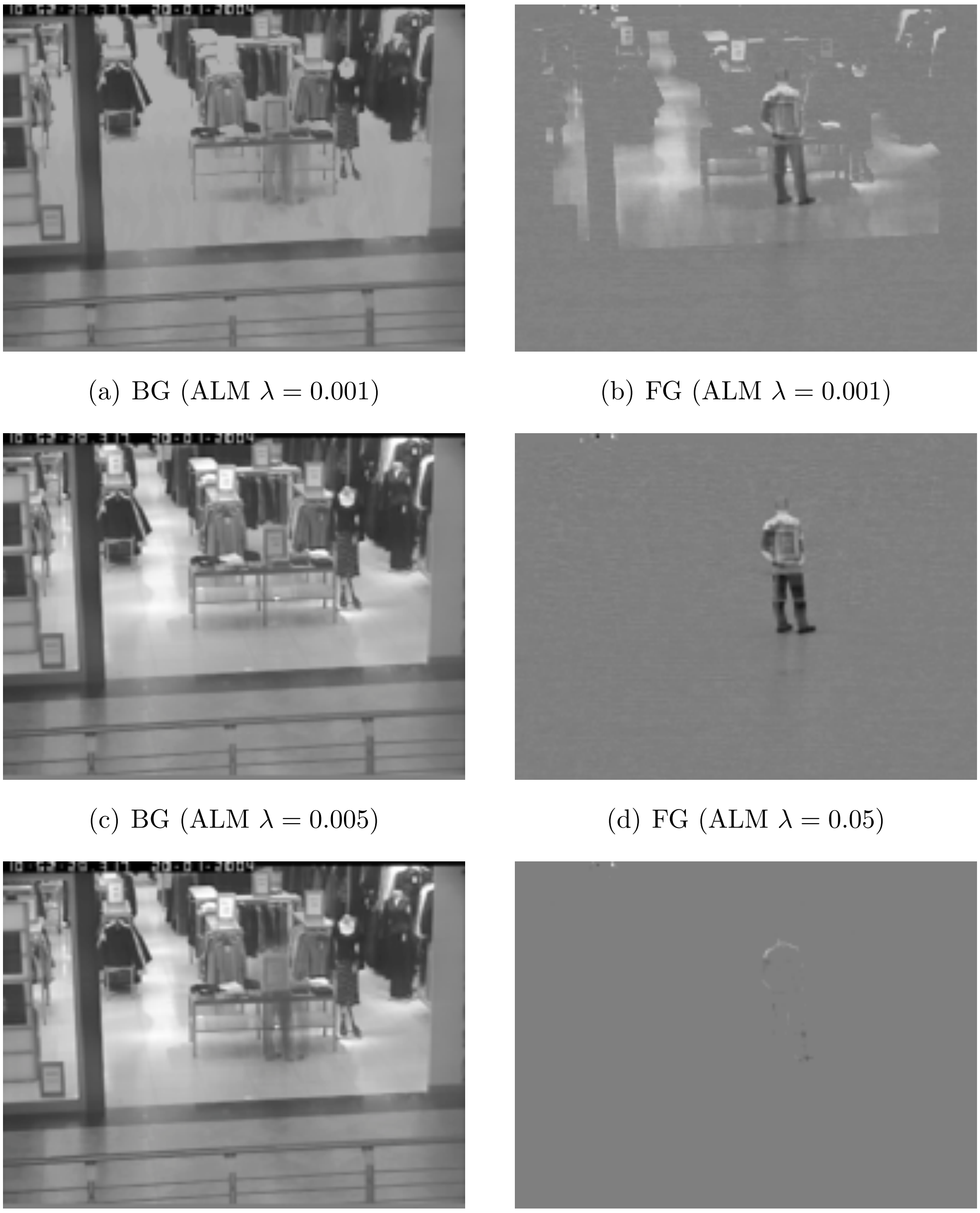
(d) FG (‘LE’-SAMF)

(e) BG (sSAMF)

(f) FG (sSAMF)

Figure 11: ‘LE’-SAMF vs segmentation-based SAMF.

$minc = 20$ as recommended by the authors. We also tested other parameter setting, and observed that FG/BG separation by sSAMF performed almost equally for $1 \leq k \leq 100$, despite the visual variation of segmented images (see Figure 13). Overall, we empirically observed that the performance of sSAMF is not very sensitive to the selection of segmented images, unless it is highly under-segmented.



(a) BG (ALM $\lambda = 0.001$)

(b) FG (ALM $\lambda = 0.001$)

(c) BG (ALM $\lambda = 0.005$)

(d) FG (ALM $\lambda = 0.05$)

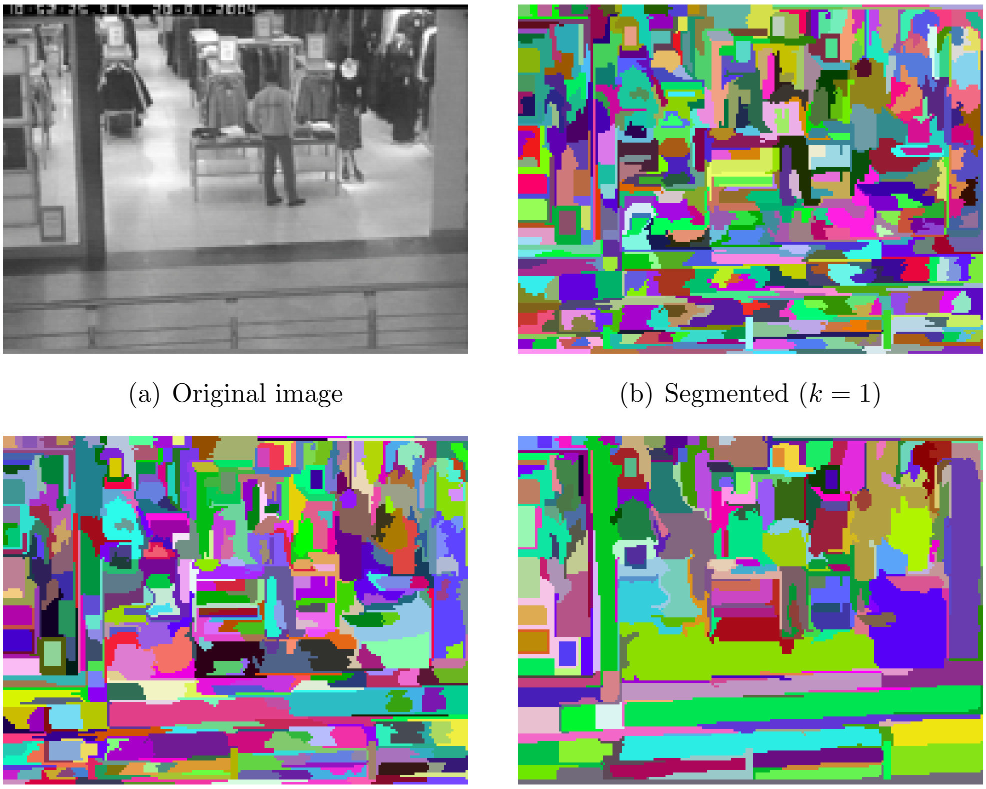
(e) BG (ALM $\lambda = 0.025$)

(f) FG (ALM $\lambda = 0.025$)

Figure 12: Results with the inexact ALM algorithm (Lin et al., 2009) for $\lambda = 0.001$ (top row), $\lambda = 0.005$ (middle row), and $\lambda = 0.025$ (bottom row) .

Computation Time

The computation time for segmentation by EGS was less than 10 sec (for 100 frames). Forming the one-to-one map $\mathcal{X}$ took more than 80 sec (which is expected to be improved). In total, sSAMF took 600 sec on a Linux machine with Xeon X5570(2.93GHz), while 'LE'-SAMF took 700 sec. This slight reduction in computation time comes from the reduction



(a) Original image

(b) Segmented ($k = 1$)

(c) Segmented ($k = 10$)

(d) Segmented ($k = 100$)

Figure 13: Segmented images by the efficient graph-based segmentation (EGS) algorithm with different k values. They are visually different, but with all these segmentations, FB/BG separation results by sSAMF were almost identical. The original image (a) is the same frame ($m = 55$ in the *difficult* video) as the one in Figure 11.

in the number K of partitions for the FG term, and hence the number of calculations of partial analytic solutions.

Online Resource

Online Resource consists of two movies that show the performance of 'LE'-SAMF (a SAMF counterpart of robust PCA) and sSAMF over all frames of the *easy* video (`SAMF_1.mpg`) and the *difficult* video (`SAMF_2.mpg`). The format of both movies is exactly the same as Figure 11, i.e., the top row shows an original frame and its segmentation, the middle row shows the BG and the FG terms obtained by 'LE'-SAMF, and the bottom row shows the BG and the FG terms obtained by sSAMF.

6 Conclusion

In this paper, we proposed a sparse additive matrix factorization (SAMF) model, which allows us to design various forms of factorization that induce various types of sparsity. We then proposed a variational Bayesian (VB) algorithm called the mean update (MU), which gives the global optimal solution for a large subset of parameters in each step. Through experiments, we showed that the MU algorithm compares favorably with the standard VB iteration. We also demonstrated the usefulness of the flexibility of SAMF in a real-world foreground/background video separation experiment, where image segmentation is used for automatically designing an SMF term.

Future work is to overcome the limitations discussed in Section 4.4. Analysis of convergence properties of the MU algorithm, and theoretical elucidation of the reason why the MU algorithm tends to give a better solution than the standard VB algorithm are also our important future work.

Acknowledgements

Shinichi Nakajima and Masashi Sugiyama thank the support from Grant-in-Aid for Scientific Research on Innovative Areas: Prediction and Decision Making, 23120004. S. Derin Babacan was supported by a Beckman Postdoctoral Fellowship.

References

A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

H. Attias. Inferring parameters and structure of latent variable models by variational Bayes. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 21–30, San Francisco, CA, 1999. Morgan Kaufmann.

S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos. Sparse Bayesian methods for low-rank matrix estimation. *IEEE Trans. on Signal Processing*, 60(8):3964–3977, 2012.

C. M. Bishop. Variational principal components. In *Proc. of International Conference on Artificial Neural Networks*, volume 1, pages 514–509, 1999.

C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, New York, NY, USA, 2006.

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

E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), 2011.

X. Ding, L. He, and L. Carin. Bayesian robust principal component analysis. *IEEE Transactions on Image Processing*, 20(12):3419–3430, 2011.

A. D'Souza, S. Vijayakumar, and S. Schaal. The Bayesian backfitting relevance vector machine. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

B. Efron and C. Morris. Stein's estimation rule and its competitors—an empirical Bayes approach. *Journal of the American Statistical Association*, 68:117–130, 1973.

P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

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*, 24:381–385, 1981.

J. Gao. Robust l1 principal component analysis and its Bayesian variational inference. *Neural Computation*, 20:555–578, 2008.

T. Hastie and R. Tibshirani. Generalized additive models. *Statistical Science*, 1(3):297–318, 1986.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

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

A. Ilin and T. Raiko. Practical approaches to principal component analysis in the presence of missing values. *Journal of Machine Learning Research*, 11:1957–2000, 2010.

L. Jacob, G. Obozinski, and J. P. Vert. Group Lasso with overlap and graph Lasso. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *Proceedings of International Conference on Machine Learning*, pages 457–464, 2009.

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

B. Lakshminarayanan, G. Bouchard, and C. Archambeau. Robust Bayesian matrix factorisation. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, volume 15, 2011.

Y. J. Lim and T. W. Teh. Variational Bayesian approach to movie rating prediction. In *Proceedings of KDD Cup and Workshop*, 2007.

Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *UIUC Technical Report UILU-ENG-09-2215*, 2009.

J. Luttinen, A. Ilin, and J. Karhunen. Bayesian robust pca for in- complete data. In *International Conference on Independent Component Analysis and Signal Separation*, 2009.

S. Nakajima and M. Sugiyama. Theoretical analysis of Bayesian matrix factorization. *Journal of Machine Learning Research*, 12:2579–2644, 2011.

S. Nakajima, M. Sugiyama, and S. D. Babacan. Sparse additive matrix factorization for robust PCA and its generalization. In S. C. H. Hoi and W. Buntine, editors, *Proceedings of Fourth Asian Conference on Machine Learning*, pages 301–316, 2012.

S. Nakajima, M. Sugiyama, S. D. Babacan, and R. Tomioka. Global analytic solution of fully-observed variational Bayesian matrix factorization. *Journal of Machine Learning Research*, 14:1–37, 2013.

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer, 1996.

S. Raman, T. J. Fuchs, P. J. Wild, E. Dahl, and V. Roth. The Bayesian group-lasso for analyzing contingency tables. In *Proceedings of International Conference on Machine Learning*, pages 881–888, 2009.

J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd International Conference on Machine learning*, pages 713–719, 2005.

R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *International Conference on Machine Learning*, 2008.

M. Seeger and G. Bouchard. Fast variational Bayesian inference for non-conjugate matrix factorization models. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, La Palma, Spain, 2012.

N. Srebro, J. Rennie, and T. Jaakkola. Maximum margin matrix factorization. In *Advances in Neural Information Processing Systems 17*, 2005.

R. Tomioka, T. Suzuki, M. Sugiyama, and H. Kashima. An efficient and general augmented Lagrangian algorithm for learning low-rank matrices. In *Proceedings of International Conference on Machine Learning*, 2010.

F. De La Torre and M. J. Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54:117–142, 2003.

S. Watanabe. *Algebraic Geometry and Statistical Learning.* Cambridge University Press, Cambridge, UK, 2009.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society B*, 68(1):49–67, 2006.

A Proof of Theorem 1 and Lemma 1

First, we consider the single-term SAMF ($S = 1$). In this case, the likelihood and the priors are written as follows:

$$p(V|\{A^{(k)}, B^{(k)}\}_{k=1}^{K}) \propto \exp\left(-\frac{1}{2\sigma^2}\left\|V - G(\{B^{(k)}A^{(k)\top}\}_{k=1}^{K}; \mathcal{X})\right\|_{\mathrm{Fro}}^2\right), \tag{40}$$

$$p(A^{(k)}) \propto \exp\left(-\frac{1}{2}\mathrm{tr}\left(A^{(k)}C_A^{(k)-1}A^{(k)\top}\right)\right), \tag{41}$$

$$p(B^{(k)}) \propto \exp\left(-\frac{1}{2}\mathrm{tr}\left(B^{(k)}C_B^{(k)-1}B^{(k)\top}\right)\right). \tag{42}$$

Let $V'^{(k)} \in \mathbb{R}^{L'(k) \times M'(k)}$ be the partitioned-and-rearranged (PR) observed matrix for k-th partition, i.e.,

$$V'^{(k)}_{l',m'} = V_{\mathcal{X}(k,l',m')}. \tag{43}$$

Since the map $\mathcal{X}$ is one-to-one, the following lemma holds:

Lemma 2 *Eq.*(40) *can be factorized as follows:*

$$p(V|\{A^{(k)}, B^{(k)}\}_{k=1}^{K}) \propto \prod_{k=1}^{K} \exp\left(-\frac{1}{2\sigma^2}\left\|V'^{(k)} - B^{(k)}A^{(k)\top}\right\|_{Fro}^2\right). \tag{44}$$

Next, we consider the general case when $S \geq 1$. Substituting Eqs.(10)–(12) and (16) into Eq.(14), we obtain the following lemma:

Lemma 3 *The free energy* (14) *for SAMF under the constraint* (15) *is given by*

$$F = \frac{1}{2}\left(LM\log(2\pi\sigma^2) + \sum_{s=1}^{S}\sum_{k=1}^{K^{(s)}}\left(M'^{(k,s)}\log\frac{|C_A^{(k,s)}|}{|\Sigma_A^{(k,s)}|} + L'^{(k,s)}\log\frac{|C_B^{(k,s)}|}{|\Sigma_B^{(k,s)}|}\right) + \frac{\|V\|^2}{\sigma^2}\right)$$
$$+ \frac{1}{2}\sum_{s=1}^{S}\sum_{k=1}^{K^{(s)}} tr\Big\{C_A^{(k,s)-1}(\widehat{A}^{(k,s)\top}\widehat{A}^{(k,s)} + M'^{(k,s)}\Sigma_A^{(k,s)})$$
$$+C_B^{(k,s)-1}(\widehat{B}^{(k,s)\top}\widehat{B}^{(k,s)} + L'^{(k,s)}\Sigma_B^{(k,s)})\Big\}$$

$$
\begin{aligned}
&+\frac{1}{2\sigma^2}tr\Bigg\{-2V^\top\left(\sum_{s=1}^{S}G(\{\widehat{B}^{(k,s)}\widehat{A}^{(k,s)\top}\}_{k=1}^{K^{(s)}};\mathcal{X}^{(s)})\right)\\
&\quad+2\sum_{s=1}^{S}\sum_{s'=s+1}^{S}G^\top(\{\widehat{B}^{(k,s)}\widehat{A}^{(k,s)\top}\}_{k=1}^{K^{(s)}};\mathcal{X}^{(s)})G(\{\widehat{B}^{(k,s')}\widehat{A}^{(k,s')\top}\}_{k=1}^{K^{(s')}};\mathcal{X}^{(s')})\Bigg\}\\
&+\frac{1}{2\sigma^2}\sum_{s=1}^{S}\sum_{k=1}^{K^{(s)}}tr\left((\widehat{A}^{(k,s)\top}\widehat{A}^{(k,s)}+M'^{(k,s)}\Sigma_A^{(k,s)})(\widehat{B}^{(k,s)\top}\widehat{B}^{(k,s)}+L'^{(k,s)}\Sigma_B^{(k,s)})\right)\\
&-\frac{1}{2}\sum_{s=1}^{S}\sum_{k=1}^{K^{(s)}}(L'^{(k,s)}+M'^{(k,s)})H'^{(k,s)} \qquad (45)
\end{aligned}
$$

Combining Lemma 2 and Lemma 3, we have the following lemma:

Lemma 4 *Given* $\{\widehat{U}^{(s)}\}_{s'\neq s}=\{\{\widehat{B}^{(k,s')}\widehat{A}^{(k,s')\top}\}_{k=1}^{K^{(s')}}\}_{s'\neq s}$*, the free energy* (14) *for SAMF under the constraint* (15) *can be expressed as a function of* $\{\widehat{A}^{(k,s)},\widehat{B}^{(k,s)},\Sigma_A^{(k,s)},\Sigma_B^{(k,s)}\}_{k=1}^{K^{(s)}}$ *as follows:*

$$
F^{(s)}(\{\widehat{A}^{(k,s)},\widehat{B}^{(k,s)},\Sigma_A^{(k,s)},\Sigma_B^{(k,s)}\}_{k=1}^{K^{(s)}})=\sum_{k=1}^{K^{(s)}}F^{(k,s)}+const.,
$$

where

$$
\begin{aligned}
F^{(k,s)}&=\frac{M'^{(k,s)}}{2}\log\frac{|C_A^{(k,s)}|}{|\Sigma_A^{(k,s)}|}+\frac{L'^{(k,s)}}{2}\log\frac{|C_B^{(k,s)}|}{|\Sigma_B^{(k,s)}|}\\
&+\frac{1}{2}tr\Big\{C_A^{(k,s)-1}(\widehat{A}^{(k,s)\top}\widehat{A}^{(k,s)}+M'^{(k,s)}\Sigma_A^{(k,s)})+C_B^{(k,s)-1}(\widehat{B}^{(k,s)\top}\widehat{B}^{(k,s)}+L'^{(k,s)}\Sigma_B^{(k,s)})\\
&+\sigma^{-2}\Big(-2\widehat{A}^{(k,s)\top}Z'^{(k,s)\top}\widehat{B}^{(k,s)}\\
&\qquad+(\widehat{A}^{(k,s)\top}\widehat{A}^{(k,s)}+M'^{(k,s)}\Sigma_A^{(k,s)})(\widehat{B}^{(k,s)\top}\widehat{B}^{(k,s)}+L'^{(k,s)}\Sigma_B^{(k,s)})\Big)\Big\}. \qquad (46)
\end{aligned}
$$

The following proposition is known:

Proposition 1 *(Bishop, 1999; Lim and Teh, 2007): The VB posterior for the plain MF model*

$$
p(V|A,B)\propto\exp\left(-\frac{1}{2\sigma^2}\|V-BA^\top\|_{Fro}^2\right), \qquad (47)
$$

$$
p(A)\propto\exp\left(-\frac{1}{2}tr\left(AC_A^{-1}A^\top\right)\right), \qquad (48)
$$

$$
p(B)\propto\exp\left(-\frac{1}{2}tr\left(BC_B^{-1}B^\top\right)\right), \qquad (49)
$$

is written as

$$r^{VB}(A,B) = \prod_{m=1}^{M} \mathcal{N}_H(\widetilde{\boldsymbol{a}}_m; \widetilde{\widehat{\boldsymbol{a}}}_m, \Sigma_A) \prod_{l=1}^{L} \mathcal{N}_H(\widetilde{\boldsymbol{b}}_l; \widetilde{\widehat{\boldsymbol{b}}}_l, \Sigma_B). \tag{50}$$

The free energy is written as

$$\begin{aligned} F^{MF} = \frac{M}{2} \log \frac{|C_A|}{|\Sigma_A|} + \frac{L}{2} \log \frac{|C_B|}{|\Sigma_B|} + \mathit{const.} \\ + \frac{1}{2} tr \Big\{ C_A^{-1} \left(\widehat{A}^\top \widehat{A} + M\Sigma_A \right) + C_B^{-1} \left(\widehat{B}^\top \widehat{B} + L\Sigma_B \right) \\ + \sigma^{-2} \left(-2\widehat{A}^\top V^\top \widehat{B} + \left(\widehat{A}^\top \widehat{A} + M\Sigma_A \right) \left(\widehat{B}^\top \widehat{B} + L\Sigma_B \right) \right) \Big\}. \end{aligned} \tag{51}$$

Now, we find that, given $\{\widehat{U}^{(s)}\}_{s' \neq s} = \{\{\widehat{B}^{(k,s')} \widehat{A}^{(k,s')\top}\}_{k=1}^{K^{(s')}}\}_{s' \neq s}$, Eqs.(16) and (46) for each (k,s) reduce to Eqs.(50) and (51), respectively, where V is replaced with $Z'^{(k,s)}$. This completes the proof of Theorem 1.

Finally, we consider the noise variance σ^2 estimation. By assumption, we know all values of $\{\{\widehat{A}^{(k,s)}, \widehat{B}^{(k,s)}, \Sigma_A^{(k,s)}, \Sigma_B^{(k,s)}\}_{k=1}^{K^{(s)}}\}_{s=1}^{S}$ that specify the VB posterior on $\{\Theta_A^{(s)}, \Theta_B^{(s)}\}_{s=1}^{S}$. $\{\{\Sigma_A^{(k,s)}, \Sigma_B^{(k,s)}\}_{k=1}^{K^{(s)}}\}_{s=1}^{S}$ are positive-definite, because they are covariance matrices. Then, Eq.(45) goes to infinity either when $\sigma^2 \to 0$ or when $\sigma^2 \to \infty$. Furthermore, Eq.(45) is differentiable with respect to $\sigma^2 (> 0)$. Consequently, any minimizer of Eq.(45) is necessarily a stationary point. By differentiating Eq.(45), we obtain Eq.(22) as a stationarity condition, which proves Lemma 1. □