Kernelized Sparse Bayesian Matrix Factorization

Caoyuan Li, Hong-Bo Xie, *Member, IEEE,* Xuhui Fan, Richard Yi Da Xu, Sabine Van Huffel, *Fellow, IEEE,* and Kerrie Mengersen

***Abstract*—Extracting low-rank and/or sparse structures using matrix factorization techniques has been extensively studied in the machine learning community. Kernelized matrix factorization (KMF) is a powerful tool to incorporate side information into the low-rank approximation model, which has been applied to solve the problems of data mining, recommender systems, image restoration, and machine vision. However, most existing KMF models rely on specifying the rows and columns of the data matrix through a Gaussian process prior and have to manually tune the rank. There are also computational issues of existing models based on regularization or the Markov chain Monte Carlo. In this study, we develop a hierarchical kernelized sparse Bayesian matrix factorization (KSBMF) model to integrate side information. The KSBMF automatically infers the parameters and latent variables including the reduced rank using the variational Bayesian inference. Also, the model simultaneously achieves low-rankness through sparse Bayesian learning and column-wise sparsity through an enforced constraint on latent factor matrices. We further connect the KSBMF with the nonlocal image processing framework to develop two algorithms for image denoising and inpainting. Experimental results demonstrate that KSBMF outperforms state-of-the-art approaches for these image restoration tasks under various levels of corruption.**

***Index Terms*—Matrix factorization, variational Bayesian inference, low-rankness, sparse Bayesian learning, image restoration**

I. Introduction

Matrix and tensor factorization tools to model data as linear combinations of basis elements have been widely used in machining learning, image restoration, compressed sensing, machine vision, recommender systems, brain signal processing, and speech enhancement. The major idea behind these methods is to extract low-rank and/or sparse structures or to predict missing values of the high-dimensional data by inferring the underlying latent factors. A broad reviews of matrix factorization can be found in [1]–[3] and its specific applications in image and video processing [4], [5], audio processing [6]. Although these methods are successful in many areas, most of them simply ignore side information, or intrinsically, are not capable of exploiting it.

Caoyuan Li is with the School of Computer Science and Technology, Beijing Institute of Technology (BIT), Beijing 100081, China, and also with the Faculty of Engineering and Information Technology, University of Technology Sydney (UTS), Ultimo, NSW 2007, Australia.

Hong-Bo Xie and Kerrie Mengersen are with the ARC Centre of Excellence for Mathematical & Statistical Frontiers, Queensland University of Technology, Brisbane, QLD 4001, Australia (hongbo.xie@qut.edu.au).

Xuhui Fan is with the School of Mathematics and Statistics, University of New South Wales, Sydney, NSW 2033, Australia.

Richard Yi Da Xu is with the Faculty of Engineering and Information Technology, University of Technology Sydney (UTS), Ultimo, NSW 2007, Australia.

Sabine Van Huffel is with the Department of Electrical Engineering, ESAT-Stadius Division, KU Leuven, Leuven 3001, Belgium.

Recently, there has been an intensive interest in integrating side information, i.e., prior knowledge or data attributes for specific data, into the factorization model to improve information extraction or prediction [7]–[16]. More precisely, side information is data that is neither from the input space nor the output space of a model but include useful information for learning it. In this study, we present a generic variational Bayesian (VB) model for matrix low rank and sparse decomposition with side information. We further develop two algorithms based on this VB framework for nonlocal image denoising and inpainting.

Our contribution: Our contributions are at two levels. From the perspective of machine learning, we present a generative model for kernelized sparse Bayesian matrix factorization (KSBMF). To determine the appropriate rank of a model, a common approach is to try different values by performing multiple runs and then choose the one that yields the best performance [17], [18]. Different from existing kernelized matrix factorization methods particularly the two VB realizations [19], [20], we adopt a sparse Bayesian formulation, where we assign the corresponding columns of latent factor matrices to possess the same column-wise sparsity. However, the authors placed the Gaussian-Wishart priors on mean vectors and precision matrices of the latent factor matrices in [19]. Consequently, the update rules of our model are different from those in [19], [20]. With this model specification, our proposed formulation implicitly estimates the rank of the matrix by pruning those columns of latent factor matrices with the diagonal covariance value lower than a pre-defined threshold. Without requiring the prior knowledge on the rank of the matrix, the model frees the user from extensive parameter-tuning and groundless attempts as in [19] and [20]. In addition, KSBMF simultaneously achieves low-rankness through sparse Bayesian learning and column-wise sparsity through the enforced constraint on latent factor matrices. Furthermore, this generic model is applicable to either recovering low rank items from noisy measurements or performing matrix completion. Another significant difference between our model and [20] is that the variance of a number of latent variables in [20] is set as constant, which is feasible for binary matrices with the purpose of multi-label classification. However, this is unacceptable in the case of denoising or inpainting an image with an unknown noise variance. The variance of each latent factor matrix is explicitly assigned as a latent variable with a specified prior in our model. In regard to the specific contribution in image processing, a large number of algorithms have been developed to exploit the nonlocal low-rank and global sparse properties for enhanced image recovery [21]. However, to the best of our knowledge, the side information of similarity between patches has never been taken into account in the image

restoration model. We devise a kernel function based on the similarity between each pair of patches. We further present two algorithms which incorporate the patch similarity-based kernel into the generic KSBMF model for enhanced image denoising and inpainting.

The rest of this paper is organized as follows. Section II provides a systematic review of matrix factorization techniques integrating side information. Section III elaborates on the model specification and inference of kernelized sparse Bayesian matrix factorization. In Section IV we first present the kernel function to integrate the side information of similarity between patches for the specific application of image restoration. Algorithms for image denoising and inpainting based on KSBMF are then described. Experimental results including comparison with state-of-the-art methods and objective assessments are presented in Section V. Finally, Section VI concludes the paper.

II. Related Work

In order to incorporate the document labels into the matrix factorization model to improve word representations for the text classification task, Yang et al. [7] constructed two co-occurrence matrices: a word-context matrix and a word-label matrix. They then defined an objective function which penalised the weighting function related to the latter matrix. Lan et al. [8] proposed a kernel low-rank decomposition formulation which represented the entries using the Nyström sampling method. The convex objective function to integrate the side information in [8] is based on the Frobenius norm, the same as in [7], to measure the closeness between two matrices. Narita et al. [9] introduced two regularization approaches using graph Laplacians induced from the side information of relationships among data, one for moderately sparse cases and the other for extremely sparse cases. They presented two kinds of iterative algorithms for approximate solutions: one based on an EM-like algorithm which is stable but not so scalable, and the other based on gradient based optimization which is applicable to large scale datasets. The matrix factorization model for recommendation in social rating networks in [10] incorporates not only trust but also distrust relationships aiming to improve the quality of recommendations and mitigate the data sparsity and cold-start issues. The social relationships are absorbed into the convex optimization problem with a standard gradient descent method to find the latent feature matrices of users and items in an iterative procedure. Fithian and Mazumder [11] explored a general statistical framework for low-rank modeling of a matrix with missing data and side information, based on convex optimization with a generalized nuclear norm penalty. An augmented Lagrange multiplier (ALM) and the alternating direction method of multipliers (ADMM) were employed to perform a robust principal component analysis with side information in [12], [13]. Nguyen and Lee [14] proposed to incorporate prior anatomical information into PET reconstruction using a nonlocal regularization method. To accelerate convergence, they used the complete-data ordered subsets expectation maximization (COSEM) algorithm, which is free from a seriously inconvenient user-specific relaxation schedule required in conventional relaxed ordered-subsets (OS) methods. In addition, the stochastic gradient decent (SGD) method was utilized in [15] to learn the latent matrix, where the interactions between user/item and field can be captured. Huang et al. [16] explored an alternating gradient descent (AGD) method to perform matrix completion with side information. As for the matrix completion problem, singular value decomposition is another popular method [22], [23]. While the aforementioned methods incorporated explicit side information in the low-rank matrix factorization setting, Shah et al. [24] designed a method to make use of the implicit information, i.e., via random walks on graphs. They casted the problem as factoring a nonlinear transform of the (partially) observed matrix and developed a coordinate descent based algorithm for the same.

Side information can also be presented and utilized in other manners. Choo el al. [25] proposed a weakly supervised nonnegative matrix factorization (NMF) that flexibly accommodates diverse forms of prior information via regularization in clustering applications. Some others assumed to know part entries of the factor matrices and used a parameterization scheme to take them into account of the NMF problem. For example, Delmaire et al. [26] presented an informed NMF model in which some entries of a factor matrix are to be provided or bounded by experts and update rules were proposed for that purpose. Dorffer et al. [27] further assumed that the columns of a matrix factor have a sparse decomposition along with a known dictionary. The idea of a convex NMF in [28] is similar to [26], [27]. However, the update rules are derived by the majorization-minimization algorithm. In another family of sparse representations [28], the kernel matrix is defined based on sample-sample similarity, or sample-basis-vector similarity.

For most of these convex or non-convex methods to utilize the side information, one has to manually choose some regularization parameters to properly control the tradeoff between the data fitting error and the matrix rank when noise is involved. However, due to the lack of the noise variance and the rank, it is often unrealistic to determine the optimal regularization parameters.

Probabilistic frameworks provide another essential principle to perform kernelized matrix factorization. Since the matrix's inner product in probabilistic PCA has an interpretation as a Gaussian process (GP) covariance matrix [24], a number of studies have been devoted to nonlinear probabilistic matrix factorization using GP latent variable models (LVM). The covariance matrix of GP-LVM was replaced by a covariance function of GP containing the side information in [29]. Inspired by this idea, Zhou et al. [30] explicitly proposed the kernelized probabilistic matrix factorization (KPMF) model, which integrated the side information through kernel matrices over rows and columns, respectively. KPMF models a matrix as the product of two latent matrices, which are sampled from two different zero-mean Gaussian processes. The covariance functions of the GPs are derived from the side information, and encode the covariance structure across rows and across columns, respectively. Adams et al. [31] extended this framework for incorporating side information by coupling together multiple dependent matrix factorization problems via

Gaussian process priors. They replaced scalar latent features with functions that vary over the space of side information. However, GP does not scale with big data due to its cubic time complexity. Le et al. [32] addressed these efficiency issues by proposing local GP kernel functions in the context of modeling road network topology.

In order to achieve automatic balance between the matrix rank and the fitting error, Bayesian methods have been recently employed to learn the KMF model parameters. Porteous et al. [33] introduced a nonparametric mixture model for the prior of the rows and columns of the factored matrices that gives a different regularization for each latent class. Besides providing a richer prior, the posterior distribution of mixture assignments inferred by Gibbs sampling reveals the latent classes [33]. This Bayesian approach outperforms other matrix factorization techniques even when using fewer dimensions. Instead of using a nonparametric mixture model for the user and item, Liu et al. [34] proposed two recommendation approaches fusing social relations and item contents with user ratings. One generates user hyperparameters separately for every user vector, while another generates both user hyperparameters and item hyperparameters separately. Xu et al. [35] employed a co-clustering technique to integrate the side information of the user community and item group into the Bayesian matrix factorization. Each community-group pair corresponds to a co-cluster, which is characterized by a rating distribution in exponential family and a topic distribution. Yang and Wang [36] presented a Bayesian hierarchical kernelized probabilistic matrix factorization for matrix-variate normal data with dependent structures induced by rows and columns. The learned the model explicitly captures the underlying correlation among the rows and the columns. The parameters in these models [33]–[36] are all inferred using Gibbs sampling. Zakeri et al. [37] extended the Markov Chain Monte Carlo (MCMC) method to factorize a sparsely filled gene-phenotype matrix with genomic and phenotypic side information, where the objective is to make non-trivial predictions for genes for which no previous disease association is known.

In comparison with MCMC sampling methods, variational Bayesian (VB) inference exhibits much lower computational complexity and has been broadly applied to infer the posterior in numerous probabilistic models. However, inference of kernelized matrix factorization models using VB is still quite limited. Pork et al. [19] placed Gaussian-Wishart priors on mean vectors and precision matrices of Gaussian user and item factor matrices, such that the mean of each prior distribution is regressed on corresponding side information. They developed a VB algorithm to approximate the posterior distributions over user and item factor matrices with a Bayesian Cramér-Rao bound. Very recently, Gonen and Kaski [20] extended the kernelized matrix factorization with a full VB treatment and with an ability to work with multiple side information sources expressed as different kernels. However, this model focused specifically on binary output matrices for multi-label classification.

Besides the issue of rank determination, there are at least two limitations of the aforementioned KMF approaches. The first issue is low-rankness and sparsity. In practice, many different data sets, for example, natural images, hyperspectral images and dynamic PET, have both nonlocal low-rank and global sparse structure properties [38]–[40]. It has been proven that the adoption of suitably combined constraints of low rankness and sparsity is expected to yield substantially enhanced estimation results [38]–[40]. However, these KMF approaches focus on either low-rankness or sparsity but fail to emphasize them together. The second issue is noisy and incomplete data. Most of these KMF approaches focus on either noisy data or incomplete data but fail to address them collectively. In this study, our proposed KSBMF model addresses all these issues together.

III. Methods

We first introduce some notations used in this paper. We denote matrices with bold capital letters, vectors with bold letters, scalars with italic letters, for example, $\mathbf{A} \in \mathbb{R}^{M\times N}$ denotes a matrix $\mathbf{A}$ of dimensions $M \times N$, $\mathbf{a}_{m\cdot}$ is its mth row, its nth column is denoted by $\mathbf{a}_{\cdot n}$ and a_{mn} denotes its (m,n)th entry. We use $\mathcal{N}(\mu, \tau^{-1})$ to indicate the univariate Gaussian distribution with mean μ and precision τ, and $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$ to represent the multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and precision matrix (inverse covariance matrix) $\boldsymbol{\Lambda}$.

A. Model specification of KSBMF

Considering the observation data as an $M \times N$ matrix $\mathbf{Y}$ either with or without missing entries, the problem is to recover the actual low-rank matrix $\mathbf{X}$ from $\mathbf{Y} = \mathbf{X} + \mathbf{E}$. We enforce a common column-wise sparsity profile on the underlying factors and thus cast it to the problem of sparse representation of factor matrices $\mathbf{G}$ and $\mathbf{H}$, that is:

$$\mathbf{Y} = \mathbf{X} + \mathbf{E} = \mathbf{G}\mathbf{H}^{\top} + \mathbf{E}, \tag{1}$$

where $\mathbf{Y} \in \mathbb{R}^{M\times N}$, $\mathbf{G} \in \mathbb{R}^{M\times r}$, $\mathbf{H} \in \mathbb{R}^{N\times r}$, $\mathbf{E} \in \mathbb{R}^{M\times N}$, and $r \ll \min(M, N)$ for column-wise sparsity.

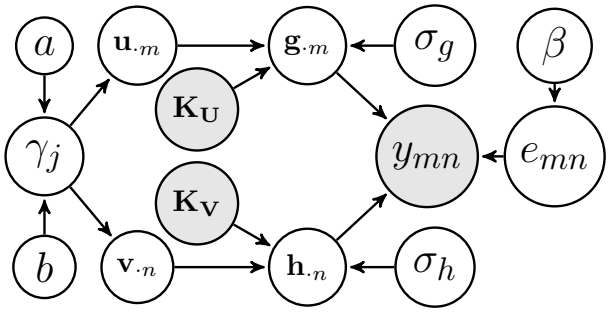


Fig. 1. Directed graphical representation of KSBMF model.

Fig. 1 shows the graphical model of the proposed hierarchical kernelized sparse Bayesian matrix factorization with latent variables and their corresponding priors. Here the actual data x_{mn} is recovered from latent vectors $\mathbf{g}_{\cdot m}$ and $\mathbf{h}_{\cdot n}$ by removing the noise e_{mn} from the observation y_{mn}. Latent vectors $\mathbf{g}_{\cdot m}$ and $\mathbf{h}_{\cdot n}$, are inferred from $\mathbf{u}_{\cdot m}$ and $\mathbf{v}_{\cdot n}$ with integrating side information $\mathbf{K_U}$ and $\mathbf{K_V}$, respectively. σ_g, σ_h, $\boldsymbol{\gamma}$ and β are precision parameters while a and b are the hyperparameters of

$\boldsymbol{\gamma}$. We specify the priors of all latent variables and parameters in this Section. In order to impose column-wise sparsity into the low rank approximation model, we assign multivariate Gaussian priors to the columns of $\mathbf{U}$ and $\mathbf{V}$ with mean vector $\mathbf{0}$ and precision matrices $\gamma_j\mathbf{I}_M$ and $\gamma_j\mathbf{I}_N$, respectively,

$$p(\mathbf{U}|\boldsymbol{\gamma}) = \prod_{j=1}^{r} \mathcal{N}(\mathbf{u}_{\cdot j}|\mathbf{0}, \gamma_j^{-1}\mathbf{I}_M), \tag{2}$$

$$p(\mathbf{V}|\boldsymbol{\gamma}) = \prod_{j=1}^{r} \mathcal{N}(\mathbf{v}_{\cdot j}|\mathbf{0}, \gamma_j^{-1}\mathbf{I}_N), \tag{3}$$

where $\mathbf{I}_J \in \mathbb{R}^{J\times J}$ denotes an identity matrix due to the fact that entries of each column $\mathbf{u}_{\cdot j}$ and $\mathbf{v}_{\cdot j}$ are independent stochastic variables. Therefore, the columns of $\mathbf{U}$ and $\mathbf{V}$ possess the same column-wise sparsity since they are enforced by the same precision γ_j. With such a constraint, most of the precision γ_j will be iteratively updated to very large values. The corresponding columns of $\mathbf{U}$ and $\mathbf{V}$ are removed since they make little contribution to the approximation $\mathbf{X}$, and hence the column-wise sparsity of latent factors $\mathbf{U}$ and $\mathbf{V}$ and low-rank of $\mathbf{X}$ are jointly satisfied. This sparse Bayesian learning formulation has been applied in compressive sensing and robust PCA [41]–[43].

To achieve the joint column-wise sparsity of $\mathbf{U}$ and $\mathbf{V}$, we further assign the conjugate Gamma hyper-prior to the precision γ_j:

$$p(\gamma_j) = \text{Gamma}(a, \frac{1}{b}) \propto \gamma_j^{a-1} exp(-b\gamma_j), \tag{4}$$

where very small values are assigned to the parameters a and b to achieve a diffuse hyper-prior. We also let $\mathbf{U}$ couple with the kernel matrix $\mathbf{K_U}$ result in a latent matrix $\mathbf{G}$, and assume that each column of $\mathbf{G}$ follows Gaussian prior with precision $\sigma_g\mathbf{I}_M$, that is,

$$p(\mathbf{G}|\mathbf{U}, \mathbf{K_U}, \sigma_g) = \prod_{j=1}^{r} \mathcal{N}(\mathbf{g}_{\cdot j}|\mathbf{K_U}^\top \cdot \mathbf{u}_{\cdot j}, \sigma_g^{-1}\mathbf{I}_M). \tag{5}$$

Similarly, the prior of $\mathbf{H}$ is defined over the latent variable $\mathbf{V}$, kernel function $\mathbf{K_V}$, and precision $\sigma_h\mathbf{I}_N$:

$$p(\mathbf{H}|\mathbf{V}, \mathbf{K_V}, \sigma_h) = \prod_{j=1}^{r} \mathcal{N}(\mathbf{h}_{\cdot j}|\mathbf{K_V}^\top \cdot \mathbf{v}_{\cdot j}, \sigma_h^{-1}\mathbf{I}_N). \tag{6}$$

Here, the precisions σ_g and σ_h of the Gaussian distribution obey the Jeffreys prior:

$$p(\sigma_g) \propto \sigma_g^{-1}, \tag{7}$$

$$p(\sigma_h) \propto \sigma_h^{-1}. \tag{8}$$

In Eq. (1), we assume that the noise $\mathbf{E}$ obeys a Gaussian distribution with zero mean and unknown precision β. Hence, $\mathbf{E}$ is modeled as:

$$p(\mathbf{E}|\beta) = \prod_{i=1}^{M}\prod_{j=1}^{N} \mathcal{N}(e_{mn}|0, \beta^{-1}), \tag{9}$$

$$p(\beta) \propto \beta^{-1}, \tag{10}$$

where β also adopts the noninformative Jeffreys prior. Given the priors defined above, the conditional distribution for the observation model is as follows:

$$p(\mathbf{Y}|\mathbf{G}, \mathbf{H}, \beta) = \prod_{i=1}^{M}\prod_{j=1}^{N} \mathcal{N}(y_{mn}|\mathbf{g}_{m\cdot}\mathbf{h}_{n\cdot}^\top, \beta^{-1}). \tag{11}$$

With the conditional probability and all priors in hand, the joint distribution is given by:

$$\begin{aligned} &p(\mathbf{Y}, \mathbf{U}, \mathbf{V}, \mathbf{G}, \mathbf{H}, \sigma_g, \sigma_h, \boldsymbol{\gamma}, \beta) \\ &= p(\mathbf{Y}|\mathbf{G}, \mathbf{H}, \beta)p(\mathbf{G}|\mathbf{U}, \mathbf{K_U}, \sigma_g)p(\mathbf{H}|\mathbf{V}, \mathbf{K_V}, \sigma_h) \\ &\cdot p(\mathbf{U}|\boldsymbol{\gamma})p(\mathbf{V}|\boldsymbol{\gamma})p(\sigma_g)p(\sigma_h)p(\boldsymbol{\gamma})p(\beta). \end{aligned} \tag{12}$$

B. Model inference of KSBMF

Full Bayesian inference using the above joint distribution is computationally intractable since the marginal distribution $p(\mathbf{Y})$ is not available analytically. We resort to variational Bayesian inference [44] to deal with this problem. We use $\mathbf{Z}$ to represent the set of all latent variables such that $\mathbf{Z} = (\mathbf{U}, \mathbf{V}, \mathbf{G}, \mathbf{H}, \sigma_g, \sigma_h, \boldsymbol{\gamma}, \beta)$. The approximate posterior distribution is therefore denoted by $q(\mathbf{Z})$. The principle is to define a parameterized family of distributions over the hidden variables and then update the parameters to minimize the Kullback-Leibler (KL) divergence between $q(\mathbf{Z})$ and the true distribution $p(\mathbf{Z}|\mathbf{Y})$, denoted by

$$\min_{q(\mathbf{Z})} KL(q(\mathbf{Z}) \parallel p(\mathbf{Z}|\mathbf{Y})) = \int q(\mathbf{Z}) ln \frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{Y})} d\mathbf{Z}. \tag{13}$$

Equivalently, this corresponds to the following maximum problem

$$\max_{q(\mathbf{Z})} -KL(q(\mathbf{Z}) \parallel p(\mathbf{Z}|\mathbf{Y})) = \int q(\mathbf{Z}) ln \frac{p(\mathbf{Z}, \mathbf{Y})}{q(\mathbf{Z})p(\mathbf{Y})} d\mathbf{Z}. \tag{14}$$

This can be referred to as estimation of the marginal likelihood $p(\mathbf{Y})$ with a maximal lower bound. With a mean field approximation, $q(\mathbf{Z})$ is factorized with respect to its partitions as [44]

$$q(\mathbf{Z}) = \prod_k q(\mathbf{Z}_k). \tag{15}$$

The expression of the optimal posterior approximation $q(\mathbf{Z}_k)$ with other variables fixed can be denoted as

$$ln\ q(\mathbf{Z}_k) = \langle ln\ p(\mathbf{Y}, \mathbf{Z})\rangle_{\mathbf{Z}\setminus\mathbf{Z}_k} + const, \tag{16}$$

where $\langle\cdot\rangle$ denotes the expectation and $const$ denotes a constant which is not dependent on the current variable. $\mathbf{Z}\setminus\mathbf{Z}_k$ means the set of $\mathbf{Z}$ with $\mathbf{Z}_k$ to be removed. Each variable is updated in turn while holding others fixed. We detail the iteration rules for all unknown variables in Eq. (15).

1) Estimation of latent factors $\mathbf{U}$ *and* $\mathbf{V}$*:* Combining the respective priors of $\mathbf{U}$ and $\mathbf{G}$ in Eqs. (2) and (5), the posterior approximation $ln\ q(\mathbf{U})$ is derived from Eq. (16) as:

$$\begin{aligned} ln\ q(\mathbf{U}) &= \langle ln\ P(\mathbf{Y}, \mathbf{Z})\rangle_{\mathbf{Z}\setminus\mathbf{U}} + const \\ &= \sum_j -\frac{1}{2}(\mathbf{u}_{\cdot j}^\top(\langle\sigma_g\rangle\mathbf{K_U}\mathbf{K_U}^\top + \boldsymbol{\Gamma}_{\mathbf{u}_{\cdot j}})\mathbf{u}_{\cdot j} \\ &\quad - 2\langle\sigma_g\rangle\mathbf{u}_{\cdot j}^\top\mathbf{K_U}\langle\mathbf{g}_{\cdot j}\rangle) + const, \end{aligned} \tag{17}$$

where $\mathbf{\Gamma}_{\mathbf{u}_{\cdot j}} = \langle\gamma_j\rangle\mathbf{I}_M$.

From Eq. (17) it is found that the posterior density of the jth column $\mathbf{u}_{\cdot j}$ of $\mathbf{U}$ obeys the multivariate Gaussian distribution:

$$q(\mathbf{u}_{\cdot j}) = \mathcal{N}(\mathbf{u}_{\cdot j}|\langle\mathbf{u}_{\cdot j}\rangle, \Sigma^{\mathbf{u}_{\cdot j}}), \tag{18}$$

with covariance and mean

$$\Sigma^{\mathbf{u}_{\cdot j}} = (\langle\sigma_g\rangle \cdot \mathbf{K_U}\mathbf{K_U}^\top + \mathbf{\Gamma}_{\mathbf{u}_{\cdot j}})^{-1}, \tag{19}$$

$$\langle\mathbf{u}_{\cdot j}\rangle = \langle\sigma_g\rangle \cdot \Sigma^{\mathbf{u}_{\cdot j}}\mathbf{K_U}\langle\mathbf{g}_{\cdot j}\rangle. \tag{20}$$

Apparently, the posterior approximation of $\mathbf{v}_{\cdot j}$ also obeys the multivariate Gaussian distribution with the density denoted by

$$q(\mathbf{v}_{\cdot j}) = \mathcal{N}(\mathbf{v}_{\cdot j}|\langle\mathbf{v}_{\cdot j}\rangle, \Sigma^{\mathbf{v}_{\cdot j}}), \tag{21}$$

and the covariance and mean are given by

$$\Sigma^{\mathbf{v}_{\cdot j}} = (\langle\sigma_h\rangle \cdot \mathbf{K_V}\mathbf{K_V}^\top + \mathbf{\Gamma}_{\mathbf{v}_{\cdot j}})^{-1}, \tag{22}$$

$$\langle\mathbf{v}_{\cdot j}\rangle = \langle\sigma_h\rangle \cdot \Sigma^{\mathbf{v}_{\cdot j}}\mathbf{K_V}\langle\mathbf{h}_{\cdot j}\rangle, \tag{23}$$

where $\mathbf{\Gamma}_{\mathbf{v}_{\cdot j}} = \langle\gamma_j\rangle\mathbf{I}_N$.

2) Estimation of $\boldsymbol{\gamma}$: Applying the priors of $\mathbf{U}, \mathbf{V}$ and $\boldsymbol{\gamma}$ in the same manner to Eq. (16), the posterior approximation of $ln\ q(\boldsymbol{\gamma})$ is given by

$$\begin{aligned} &ln\ q(\boldsymbol{\gamma}) = \langle P(\mathbf{Y}, \mathbf{Z})\rangle_{\mathbf{Z}\setminus\boldsymbol{\gamma}} + const \\ &= ln(\gamma_j^{a-1+\frac{M+N}{2}} exp(-\frac{1}{2}\gamma_j(\langle\mathbf{u}_{\cdot j}^T\mathbf{u}_{\cdot j}\rangle + \langle\mathbf{v}_{\cdot j}^\top\mathbf{v}_{\cdot j}\rangle + 2b))) \\ &+ const. \end{aligned} \tag{24}$$

This is equivalent to

$$q(\gamma_j) \propto \gamma_j^{a-1+\frac{M+N}{2}} exp(-\frac{1}{2}\gamma_j(\langle\mathbf{u}_{\cdot j}^\top\mathbf{u}_{\cdot j}\rangle + \langle\mathbf{v}_{\cdot j}^\top\mathbf{v}_{\cdot j}\rangle + 2b)). \tag{25}$$

So the posterior distribution of γ_j is a Gamma distribution with mean

$$\langle\gamma_j\rangle = \frac{2a + M + N}{2b + \langle\mathbf{u}_{\cdot j}^\top\mathbf{u}_{\cdot j}\rangle + \langle\mathbf{v}_{\cdot j}^\top\mathbf{v}_{\cdot j}\rangle}. \tag{26}$$

The required expectations here are found as

$$\langle\mathbf{u}_{\cdot j}^\top\mathbf{u}_{\cdot j}\rangle = \langle\mathbf{u}_{\cdot j}\rangle^\top\langle\mathbf{u}_{\cdot j}\rangle + \mathrm{tr}(\Sigma^{\mathbf{u}_{\cdot j}}), \tag{27}$$

$$\langle\mathbf{v}_{\cdot j}^\top\mathbf{v}_{\cdot j}\rangle = \langle\mathbf{v}_{\cdot j}\rangle^\top\langle\mathbf{v}_{\cdot j}\rangle + \mathrm{tr}(\Sigma^{\mathbf{v}_{\cdot j}}). \tag{28}$$

3) Estimation of $\mathbf{G}$ and $\mathbf{H}$: Similar to estimation of $\mathbf{U}$ and $\mathbf{V}$, the posterior approximation of $\mathbf{G}$ is given by

$$\begin{aligned} &ln\ q(\mathbf{G}) = \langle ln\ P(\mathbf{Y}, \mathbf{Z})\rangle_{\mathbf{Z}\setminus\mathbf{G}} + const \\ &= \sum_i[-\frac{1}{2}(\mathbf{g}_{i\cdot}(\langle\beta\rangle\langle\mathbf{H}^\top\mathbf{H}\rangle + \langle\sigma_g\rangle\mathbf{I}_r)\mathbf{g}_{i\cdot}^T \\ &- 2\mathbf{g}_{i\cdot}(\langle\mathbf{H}\rangle^\top\mathbf{y}_{i\cdot}^\top + \langle\sigma_g\rangle\langle\mathbf{U}\rangle^\top\mathbf{K}_{\mathbf{U}_{\cdot i}}))] + const, \end{aligned} \tag{29}$$

which indicates that the ith row of $\mathbf{G}$ obeys the multivariate Gaussian distribution

$$q(\mathbf{g}_{i\cdot}) = \mathcal{N}(\mathbf{g}_{i\cdot}|\langle\mathbf{g}_{i\cdot}\rangle, \Sigma^{\mathbf{G}}). \tag{30}$$

The corresponding covariance and mean are denoted as

$$\Sigma^{\mathbf{G}} = (\langle\beta\rangle\langle\mathbf{H}^\top\mathbf{H}\rangle + \langle\sigma_g\rangle\mathbf{I}_r)^{-1}, \tag{31}$$

$$\langle\mathbf{g}_{i\cdot}\rangle^\top = \Sigma^{\mathbf{G}}(\langle\sigma_g\rangle\langle\mathbf{U}\rangle^\top\mathbf{K}_{\mathbf{u}_{\cdot i}} + \langle\beta\rangle\langle\mathbf{H}\rangle^\top\mathbf{y}_{i\cdot}^\top). \tag{32}$$

The jth row of $\mathbf{H}$ obeys another multivariate Gaussian distribution

$$q(\mathbf{h}_{j\cdot}) = \mathcal{N}(\mathbf{h}_{j\cdot}|\langle\mathbf{h}_{j\cdot}\rangle, \Sigma^{\mathbf{H}}), \tag{33}$$

with covariance and mean

$$\Sigma^{\mathbf{H}} = (\langle\beta\rangle\langle\mathbf{G}^\top\mathbf{G}\rangle + \langle\sigma_h\rangle\mathbf{I}_r)^{-1}, \tag{34}$$

$$\langle\mathbf{h}_{j\cdot}\rangle^\top = \Sigma^{\mathbf{H}}(\langle\sigma_h\rangle\langle\mathbf{V}\rangle^\top\mathbf{K}_{\mathbf{V}_{\cdot j}} + \langle\beta\rangle\langle\mathbf{G}\rangle^\top\mathbf{y}_{\cdot j}). \tag{35}$$

The required expectations are expressed as

$$\langle\mathbf{G}^\top\mathbf{G}\rangle = \langle\mathbf{G}\rangle^\top\langle\mathbf{G}\rangle + m\Sigma^{\mathbf{G}}, \tag{36}$$

$$\langle\mathbf{H}^\top\mathbf{H}\rangle = \langle\mathbf{H}\rangle^\top\langle\mathbf{H}\rangle + n\Sigma^{\mathbf{H}}. \tag{37}$$

4) Estimation of β, σ_g and σ_h: The posterior probability densities of β, σ_g and σ_h are all found to be Gamma distributed. For the noise precision β, we have

$$q(\beta) \propto \beta^{\frac{MN}{2}-1} exp(-\frac{1}{2}\beta\langle\parallel \mathbf{Y} - \mathbf{G}\mathbf{H}^\top \parallel_F^2\rangle), \tag{38}$$

with its expectation

$$\langle\beta\rangle = \frac{MN}{\langle\parallel \mathbf{Y} - \mathbf{G}\mathbf{H}^\top \parallel_F^2\rangle}. \tag{39}$$

The required expectation to estimate $\langle\beta\rangle$ is denoted as

$$\begin{aligned} \langle\parallel \mathbf{Y} - \mathbf{G}\mathbf{H}^\top \parallel_F^2\rangle =& \parallel \mathbf{Y} - \langle\mathbf{G}\rangle\langle\mathbf{H}\rangle^\top \parallel_F^2 + tr(N\langle\mathbf{G}\rangle^\top\langle\mathbf{G}\rangle\Sigma^{\mathbf{H}}) \\ &+ tr(M\langle\mathbf{H}\rangle^\top\langle\mathbf{H}\rangle\Sigma^{\mathbf{G}}) + tr(MN\Sigma^{\mathbf{G}}\Sigma^{\mathbf{H}}). \end{aligned} \tag{40}$$

The updating rules for σ_g and σ_h are derived in the same manner:

$$\langle\sigma_g\rangle = \frac{Mr}{\langle\parallel \mathbf{G} - \mathbf{K_U}^\top\mathbf{U} \parallel_F^2\rangle}, \tag{41}$$

$$\langle\sigma_h\rangle = \frac{Nr}{\langle\parallel \mathbf{H} - \mathbf{K_V}^\top\mathbf{V} \parallel_F^2\rangle}, \tag{42}$$

with required expectations:

$$\begin{aligned} \langle\parallel \mathbf{G} - \mathbf{K_U}^\top\mathbf{U} \parallel_F^2\rangle =& \parallel \langle\mathbf{G}\rangle - \mathbf{K_U}^\top\langle\mathbf{U}\rangle \parallel_F^2 \\ &+ \mathrm{tr}(M\mathbf{K_U}^\top\mathbf{K_U}\Sigma^{\mathbf{U}}) + \mathrm{tr}(M\Sigma^{\mathbf{G}}), \end{aligned} \tag{43}$$

$$\begin{aligned} \langle\parallel \mathbf{H} - \mathbf{K_V}^\top\mathbf{V} \parallel_F^2\rangle =& \parallel \langle\mathbf{H}\rangle - \mathbf{K_V}^\top\langle\mathbf{V}\rangle \parallel_F^2 \\ &+ \mathrm{tr}(N\mathbf{K_V}^\top\mathbf{K_V}\Sigma^{\mathbf{V}}) + \mathrm{tr}(N\Sigma^{\mathbf{H}}). \end{aligned} \tag{44}$$

We update each parameter in turn while holding others fixed. By the properties of VB, convergence to a local minimum of the algorithm can be guaranteed after iterations [44].

The aim of the above inference is to recover $\mathbf{X}$ from the noisy matrix $\mathbf{Y}$ without missing data. For matrix completion, suppose we have a subset Ω of $\mathbf{Y}$, that is, $\mathbf{Y}_{ij} = \mathbf{X}_{ij} : (i,j) \in \Omega$. The cardinality of Ω is wMN with $0 < w \leq 1$. The observation model of Eq. (11) is thus denoted as:

$$p(W_\Omega(\mathbf{Y})|\mathbf{G}, \mathbf{H}) = \prod_{(i,j)\in\Omega} \mathcal{N}(y_{ij}|\mathbf{g}_{i\cdot}\mathbf{h}_{j\cdot}^\top, \beta^{-1}). \tag{45}$$

The corresponding joint distribution of Eq. (12) is modified as

$$\begin{aligned} &p(W_\Omega(\mathbf{Y}),\mathbf{U},\mathbf{V},\mathbf{G},\mathbf{H},\sigma_g,\sigma_h,\boldsymbol{\gamma},\beta)\\ =&p(W_\Omega(\mathbf{Y})|\mathbf{G},\mathbf{H},\beta)p(\mathbf{G}|\mathbf{U},\mathbf{K_U},\sigma_g)p(\mathbf{H}|\mathbf{V},\mathbf{K_V},\sigma_h)\\ &\cdot p(\mathbf{U}|\boldsymbol{\gamma})p(\mathbf{V}|\boldsymbol{\gamma})p(\sigma_g)p(\sigma_h)p(\boldsymbol{\gamma})p(\beta). \end{aligned} \tag{46}$$

Some of the updating rules need to be modified to accommodate the incomplete matrix $\mathbf{Y}$. The covariance and mean of the posterior density of $\mathbf{G}$ is expressed as

$$\Sigma_i^{\mathbf{G}} = (\langle\beta\rangle\langle\mathbf{H}_\Omega^\top\mathbf{H}_\Omega\rangle + \langle\sigma_g\rangle\mathbf{I}_r)^{-1}, \tag{47}$$

$$\langle\mathbf{g}_{i\cdot}\rangle^\top = \Sigma_i^{\mathbf{G}}(\langle\sigma_g\rangle\langle\mathbf{U}\rangle^\top\mathbf{K}_{\mathbf{U}\cdot i} + \langle\beta\rangle\langle\mathbf{H}_\Omega\rangle^\top\mathbf{y}_{i\cdot}^\top), \tag{48}$$

where the matrix $\mathbf{H}_\Omega$ contains only the jth rows of $\mathbf{H}$ for which $(i,j)\in\Omega$, such that

$$\begin{aligned} \langle\mathbf{H}_\Omega^\top\mathbf{H}_\Omega\rangle &= \sum_{j:(i,j)\in\Omega}\langle\mathbf{h}_{j\cdot}^\top\mathbf{h}_{j\cdot}\rangle\\ &= \sum_{j:(i,j)\in\Omega}\langle\mathbf{h}_{j\cdot}^\top\rangle\langle\mathbf{h}_{j\cdot}\rangle + \Sigma_j^{\mathbf{H}}, \end{aligned} \tag{49}$$

where $\Sigma_j^{\mathbf{H}}$ is the posterior covariance of jth row of $\mathbf{H}$. The row vector $\mathbf{y}_{i\cdot}$ contains those observed entries in the ith row of $\mathbf{Y}$.

Similarly, the mean and covariance of the posterior density of the jth row $\mathbf{h}_{j\cdot}$ is given by

$$\Sigma_j^{\mathbf{H}} = (\langle\beta\rangle\langle\mathbf{G}_\Omega^\top\mathbf{G}_\Omega\rangle + \langle\sigma_h\rangle\mathbf{I}_r)^{-1}, \tag{50}$$

$$\langle\mathbf{h}_{j\cdot}\rangle^\top = \Sigma_j^{\mathbf{H}}(\langle\sigma_h\rangle\langle\mathbf{V}\rangle^\top\mathbf{K}_{\mathbf{v}\cdot j} + \langle\beta\rangle\langle\mathbf{G}_\Omega\rangle^\top\mathbf{y}_{\cdot j}), \tag{51}$$

where $\mathbf{G}_\Omega$ contains the ith rows of $\mathbf{G}$ for which $(i,j)\in\Omega$, such that

$$\begin{aligned} \langle\mathbf{G}_\Omega^\top\mathbf{G}_\Omega\rangle &= \sum_{i:(i,j)\in\Omega}\langle\mathbf{g}_{i\cdot}^\top\mathbf{g}_{i\cdot}\rangle\\ &= \sum_{i:(i,j)\in\Omega}\langle\mathbf{g}_{i\cdot}^\top\rangle\langle\mathbf{g}_{i\cdot}\rangle + \Sigma_i^{\mathbf{G}}, \end{aligned} \tag{52}$$

where $\Sigma_i^{\mathbf{G}}$ is the posterior covariance of ith row of $\mathbf{G}$. The column vector $\mathbf{y}_{\cdot j}$ contains those observed entries in the jth column of $\mathbf{Y}$. Correspondingly, the mean of the posterior approximation of β is given by:

$$\langle\beta\rangle = \frac{wMN}{\langle\parallel W_\Omega(\mathbf{Y}) - W_\Omega(\mathbf{G}\mathbf{H}^\top)\parallel_F^2\rangle}. \tag{53}$$

IV. Algorithms for Image Restoration

A. Construction of the kernel

Construction of an effective kernel plays an essential role in guaranteeing a good performance of kernelized matrix factorization. However, the kernel is problem-dependent, and there is no unified rule to construct kernels. So far, graph kernel, diffusion kernel, commute time kernel, and regularized Laplacian kernel have been developed for utilizing the side information in recommender systems [30]. In the area of image processing, the kernel incorporating the local spatial smoothness of an image has been developed to improve image inpainting [30]. In the past decade, many algorithms based on the nonlocal framework have been proposed for image restoration, most of which significantly outperform methods utilizing image local properties [21]. In this study, we aim to apply the KSBMF model under the nonlocal framework to improve image denoising and inpainting. Here we present a new kernel which incorporates the similarity information between patches into patch group matrix factorization. Denoting the Euclidean distance between a pair of patches (i,j) by $d_E^{i,j} = \parallel \mathbf{y}_i - \mathbf{y}_j \parallel_2$, we define the similarity between them, i.e., entry of $\mathbf{K_U}$ or $\mathbf{K_V}$ as

$$k_{ij} = \sqrt[4]{\frac{1}{1+d_E^{i,j}/M}}, \tag{54}$$

where M is the total number of pixels in the patch.

Under the nonlocal framework, a pixel and its nearest neighbors in the window of $\sqrt{M}\times\sqrt{M}$ are modeled as a column vector. The $M\times N$ patch group matrix $\mathbf{Y}$ is constructed by grouping other $N-1$ patches with similar local spatial structures to the underlying one in the local window. Since each column shares similar underlying image structures, the noise-free patch group matrix $\mathbf{Y}$ has the low-rank property. Previous algorithms mainly focus on this low-rank property while neglecting the similarity between the patches in image restoration. We jointly take account of low-rankness and similarity between patches as side information to recover the image. With the kernel defined in Eq. (54), a nonlocal neighbor patch with larger similarity value has a more substantial contribution in the KSBMF model to recover the target patch.

B. Algorithm for image denoising

For the complete image denoising algorithm, we first cluster patches with a similar spatial structure to form a patch matrix. KSBMF is then applied in succession on each patch group matrix. The denoised patches are aggregated to reconstruct the whole noise-free image. In practice, iterative regularization is often adopted by mapping the filtered noise back to the denoised image, which has been demonstrated to be effective in improving the performance [45], [46]. This scheme is implemented as

$$\mathbf{Y}^{(d+1)} = \hat{\mathbf{X}}^{(d)} + \delta(\mathbf{Y} - \hat{\mathbf{X}}^{(d)}), \tag{55}$$

where d denotes algorithm iteration and $0<\delta<1$ is a relaxation parameter. The complete procedure for the KSBMF based image denoising algorithm is summarized in Algorithm 1.

C. Algorithm for image inpainting

In the case of the image with missing entries, particularly for highly incomplete cases, the similarity between two patches may be highly unreliable. Naturally, such a poorly matched patch group matrix directly degrades the inpainting effect. Hence the algorithm for inpainting is slightly different from denoising: we first perform KSBMF on the entire image to give a proper value for each missing entry. Then we execute the patch matching and re-fill each missing value at the patch

Algorithm 1 Image denoising by KSBMF

Input: Noisy image $\mathbf{y}$
Initialize: $\hat{\mathbf{x}}^{(0)} = \mathbf{y}$, $\hat{\mathbf{y}}^{(0)} = \mathbf{y}$;
for $d = 1 : D$ **do**
  Iterative regularization using Eq. (55)
  for each patch $\mathbf{y}_i$ in $\mathbf{y}^{(d)}$ **do**
    Cluster similar patches to group matrix $\mathbf{Y}_i$ corresponding to $\mathbf{y}_i$;
    Update $\mathbf{G}$ using Eq. (32);
    Update $\mathbf{H}$ using Eq. (35);
    Update β using Eq. (39);
    Update σ_g using Eq. (41);
    Update σ_h using Eq. (42);
    Update $\mathbf{U}$ using Eq. (20);
    Update $\mathbf{V}$ using Eq. (23);
    Update $\boldsymbol{\gamma}$ using Eq. (26);
  end for
  Aggregate $\hat{\mathbf{X}}_i$ to form the denoised image $\hat{\mathbf{x}}^{(d)}$
end for
Output: denoised image $\hat{\mathbf{x}}^{(D)}$

Algorithm 2 Image inpainting by KSBMF

Input: Incomplete image $\mathbf{y}$
Update $\mathbf{G}$ using Eq. (48);
Update $\mathbf{H}$ using Eq. (51);
Update β using Eq. (53);
Update σ_g using Eq. (41);
Update σ_h using Eq. (42);
Update $\mathbf{U}$ using Eq. (20);
Update $\mathbf{V}$ using Eq. (23);
Update $\boldsymbol{\gamma}$ using Eq. (26);
Pre-completed image $y^{(1)}$
for $d = 2 : D$ **do**
  for each patch $\mathbf{y}_i$ in $\mathbf{y}^{(d)}$ **do**
    Cluster similar patches to group matrix $\mathbf{Y}_i$ corresponding to $\mathbf{y}_i$;
    Repeat $1 - 8$;
  end for
  Aggregate $\hat{\mathbf{X}}_i$ to form the inpainted image $\hat{\mathbf{x}}^{(d)}$
end for
Output: Inpainted image $\hat{\mathbf{x}}^{(D)}$

group level. The procedure for image inpainting is summarized as Algorithm 2.

It should be noted that a straightforward extension of algorithms for grayscale image to colour images often introduces perturbing colour artefacts [47], [48]. The alternative option is to convert the usual RGB image to YUV (or YCrCb) colour system where the independent processing of each channel does not create noticeable colour artefacts. Due to the nature of the colour transform, the luminance component contains most of the valuable information about primitive image structures and has a higher SNR than the two chroma channels U and V [47], [48]. To take advantage of this fact and take account for the patch grouping operation sensitive to the presence of noise, the grouping of the patches is first performed only from the luminance channel. Then, the same set of group indices are used for the other two channels. Using these sets, the image restoration (denoising or inpainting) and the aggregation are performed separately on each of the three channels. Finally, the inverse transform converts the result to an RGB image.

V. Experiments on image restoration

A. Parameter setting and performance evaluation

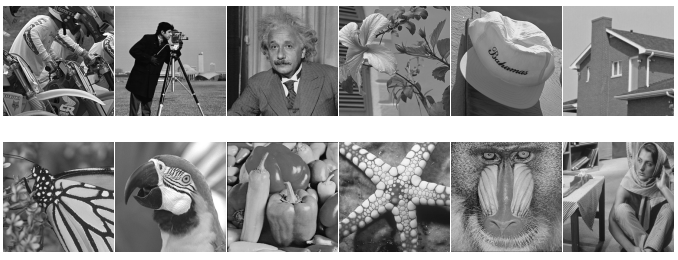


Fig. 2. The 12 test images used in image denoising experiments.

In this section we provide experimental results of image restoration using the KSBMF model. The performance of image denoising is evaluated on twelve benchmark grayscale images, shown in Fig. 2. The sizes of the first 10 images are 256×256 with the size of Baboon and Barbara being 512×512. Noisy images are produced by adding zero mean white Gaussian noise with standard deviation $\sigma = 20,\ 50,\ 70$ and 100. We adopted the setting of patch size and the number of similar patches recommended in previous studies [45], [46]: the former is set to 6×6, 7×7, 8×8 and 9×9, and the latter is set to 70, 90, 120 and 140 for $\sigma \leq 20$, $20 \leq \sigma \leq 40$, $40 < \sigma \leq 60$ and $\sigma > 60$ respectively. Throughout this study, the scaling factor δ is fixed to 0.2 for all noise levels.

In the image inpainting problem, we test the algorithm on part of the grayscale images in Fig. 2 and two colour images. The patch size is fixed as 10×10 and the number of similar patches to 60, which is slightly different from image denoising [49].

The kernel matrix $\mathbf{K_V}$ is set using Eq (54) to utilize the similarity information between the patches. Since there is no such similarity between rows of patch group matrix, $\mathbf{K_U}$ is set as the identity matrix.

We evaluate the performance of KSBMF in terms of PSNR and SSIM. Given a ground truth grayscale image $\mathbf{x}$, the PSNR of the recovered image $\hat{\mathbf{x}}$ is estimated by:

$$PSNR(\mathbf{x}, \hat{\mathbf{x}}) = 10 \cdot \log_{10}\left(\frac{255^2}{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2}\right). \tag{56}$$

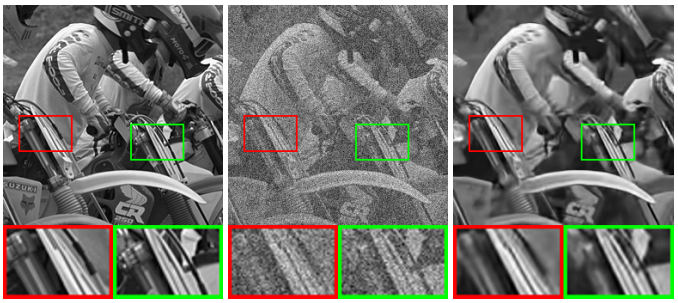
Assuming an image patch $\mathbf{A}$ from $\mathbf{x}$ as well as the patch $\mathbf{B}$ from the corresponding recovery $\hat{\mathbf{x}}$, the SSIM index between $\mathbf{A}$ and $\mathbf{B}$ is defined by:

$$SSIM(\mathbf{A}, \mathbf{B}) = \frac{2(\mu_{\mathbf{A}}\mu_{\mathbf{B}} + C_1)(2\nu_{\mathbf{AB}} + C_2)}{(\mu_{\mathbf{A}}^2 + \mu_{\mathbf{B}}^2 + C_1)(\nu_{\mathbf{A}}^2 + \nu_{\mathbf{B}}^2 + C_2)}, \tag{57}$$

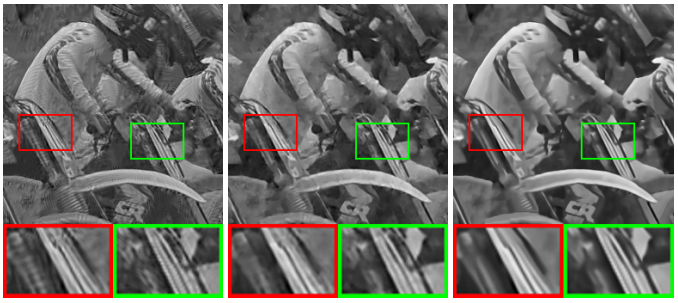
where $\mu_{\mathbf{A}}$ and $\nu_{\mathbf{A}}$ are the average intensity and standard deviation of $\mathbf{A}$, respectively; $\nu_{\mathbf{AB}}$ denotes the cross-correlation between $\mathbf{A}$ and $\mathbf{B}$, and the constants C_1 and C_2 are intended

to avoid instability. The SSIM of the entire image is estimated by averaging the local SSIM indices using a sliding window [50]. Distorted images can have roughly the same mean squared error values with respect to the original image, but very different quality. SSIM gives a much better indication of image quality for measuring the similarity between two images, which integrates luminance, contrast, and structure comparisons into its mathematical representation.

B. Image denoising



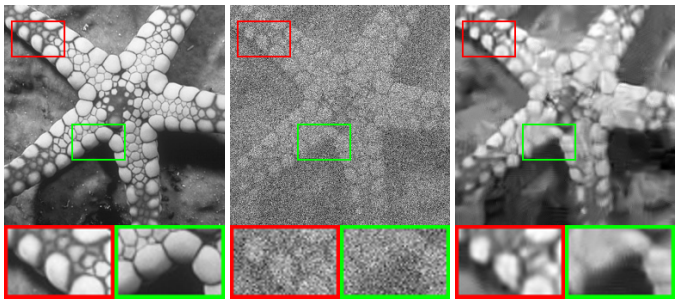
(a) Original (b) Noisy (c) BM3D



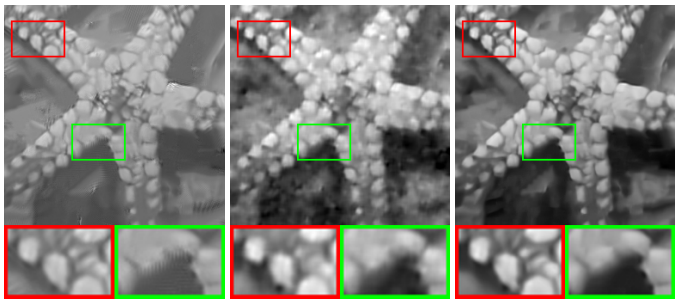
(d) WNNM (e) BPFA (f) KSBMF

Fig. 3. Comparison of denoising results on the Bike image contaminated by Gaussian white noise with $\sigma = 50$. (a) Original image, (b) Noisy image (PSNR= 14.12 dB), (c) BM3D (PSNR= 22.42 dB), (d) WNNM (PSNR= 22.50 dB), (e) BPFA (PSNR= 23.08 dB), and (f) KSBMF (PSNR= $\mathbf{23.11}$ dB).

In recent years, nonlocal methods have boosted the performance of image denoising significantly. BM3D is the benchmark algorithm of image nonlocal denoising [45]. Weighted nuclear norm minimization (WNNM) [46] is always ranked as one of the most competitive methods in comparative studies. Bayesian robust matrix factorization (BPFA) [51] shares a similar principle to KSBMF in that VB is used to infer the factor matrices; however, the former neglects the side information. We thus compare the performance of KSBMF with BM3D, WNNM, and BPFA. The proposed algorithm is implemented in MATLAB, while the others are tested using the executables and source codes provided by the authors. We estimate PSNR and SSIM for each scheme with $\sigma = 20, 50, 70$ and 100 dB. The PSNRs and SSIMs values are displayed in Table I and II respectively, where the best results are bolded. One can first find that KSBMF outperforms both BPFA and BM3D for all noise levels. It is reasonable to attribute the superiority of KSBMF over BPFA to the incorporation of side information in the model inference. Besides, with the increase



(a) Original (b) Noisy (c) BM3D



(d) WNNM (e) BPFA (f) KSBMF

Fig. 4. Comparison of denoising results on the Starfish image contaminated by Gaussian white noise with $\sigma = 100$. (a) Original image, (b) Noisy image (PSNR= 8.10 dB), (c) BM3D (PSNR= 20.00 dB), (d) WNNM (PSNR= 19.05 dB), (e) BPFA (PSNR= 19.70 dB), and (f) KSBMF (PSNR= $\mathbf{20.21}$ dB).

of the noise level, our algorithm performs increasingly better than WNNM. However, in the case of low noise level, the performance of KSBMF is in general equivalent to WNNM. In Fig. 3 and 4, we have compared the visual quality of the denoising results on four methods. Two close-up views are shown at the bottom of each result for better visualization. In Fig. 3, we compare the Bike picture under noise level $\sigma = 50$. It is observed from the close-up views that KSBMF reconstructs more image details from the noisy observation. However, methods BM3D and BRMF over-smooth textures while artifacts are visible in the close-up views for WNNM.

In Fig. 4, we compare Starfish under noise level $\sigma = 100$. Due to the much high noise level, the results of all methods suffer from more or less artifacts in smooth areas and around edges. However, KSBMF achieves a much more visually satisfactory result with less fleck and preserves much better the image edge structures, for example, along with the edge between the Starfish image and the background, than other competing methods. Overall, both quantitative assessment and visual inspection demonstrate that KSBMF yields better restoration of edges and fewer artifacts in comparison with the state-of-the-art methods in severe contamination, and is competitive to WNMM at medium noise strength.

C. Image inpainting

We evaluate the performance of KSBMF on four inpainting tasks, i.e., random missing pixels filling, text removal, block completion, and recovery of noisy image with random missing pixels. The corresponding three comparative algorithms include BPFA based on Bayesian matrix factorization [51],

TABLE I
DENOISING RESULTS (PSNR) BY COMPETING METHODS ON THE 12 TEST IMAGES. BEST RESULTS ARE IN BOLD.

σ	20				50			
schemes	BM3D	WNNM	BPFA	KSBMF	BM3D	WNNM	BPFA	KSBMF
Bike	28.24	28.70	27.89	**28.77**	22.42	22.50	23.08	**23.11**
Cameraman	30.36	**30.68**	30.14	30.60	24.99	25.16	24.85	**25.65**
Einstein	31.29	31.47	30.85	**31.51**	27.11	27.19	26.73	**27.31**
Flower	29.99	30.42	29.68	**30.47**	25.12	25.33	24.78	**25.64**
Hat	31.55	**32.05**	31.44	31.92	27.14	27.23	26.58	**27.59**
House	33.88	**34.14**	33.69	34.07	29.39	**29.87**	28.60	29.55
Monarch	30.52	**31.34**	29.45	31.23	25.46	25.56	25.28	**26.06**
Parrot	29.88	**30.03**	29.32	29.81	24.76	24.69	24.75	**25.22**
Peppers	31.28	**31.59**	31.18	31.52	26.16	26.23	25.54	**26.49**
Starfish	29.45	30.20	29.63	**30.27**	24.29	24.41	24.19	**24.72**
Baboon	25.58	**25.67**	25.03	25.60	21.83	22.15	21.90	**22.52**
Barbara	31.23	**31.68**	31.16	31.64	26.24	26.72	26.42	**26.77**
Average	30.27	**30.66**	29.96	30.62	25.41	25.59	25.23	**25.89**
σ	70				100			
schemes	BM3D	WNNM	BPFA	KSBMF	BM3D	WNNM	BPFA	KSBMF
Bike	20.46	20.08	20.29	**20.95**	18.38	17.83	18.18	**18.68**
Cameraman	22.56	22.72	22.38	**23.27**	19.86	20.25	20.13	**20.70**
Einstein	25.23	24.97	24.47	**25.48**	22.63	21.79	21.47	**22.73**
Flower	23.20	23.47	23.30	**23.82**	20.59	21.60	21.04	**21.68**
Hat	25.46	25.23	24.80	**25.79**	22.90	22.59	22.43	**23.21**
House	26.98	27.15	26.47	**27.68**	23.71	23.27	23.00	**24.12**
Monarch	22.99	23.40	23.08	**23.98**	19.85	20.82	20.43	**21.36**
Parrot	22.15	22.39	22.35	**22.98**	19.17	19.70	19.55	**20.35**
Peppers	23.97	23.63	23.48	**24.20**	21.52	20.82	21.12	**21.65**
Starfish	22.35	21.83	22.28	**22.74**	20.00	19.05	19.70	**20.21**
Baboon	20.58	20.87	20.32	**21.15**	19.17	19.39	19.49	**20.16**
Barbara	24.56	24.89	24.31	**25.14**	23.34	23.18	22.92	**24.07**
Average	23.37	23.39	23.13	**23.93**	20.93	20.87	20.79	**21.58**

(a) Original (b) Noisy (c) BPFA

(d) GSR (e) TSLRA (f) KSBMF

Fig. 5. Visual comparison for random missing pixel filling on Barbara. (a) Original image. (b) Image with 20% random samples. (c) BPFA (PSNR=22.30 dB). (d) GSR (PSNR=22.66dB). (e) TSLRA (PSNR=25.34 dB). (f) KSBMF (PSNR=**25.73** dB).

(a) Original (b) Noisy (c) BPFA

(d) GSR (e) TSLRA (f) KSBMF

Fig. 6. Visual comparison for random missing pixels filling on Monarch. (a) Original image. (b) Image with 40% random samples. (c) BPFA (PSNR=29.76 dB). (d) GSR (PSNR=30.76dB). (e) TSLRA (PSNR=30.13 dB). (f) KSBMF (PSNR=**30.87** dB).

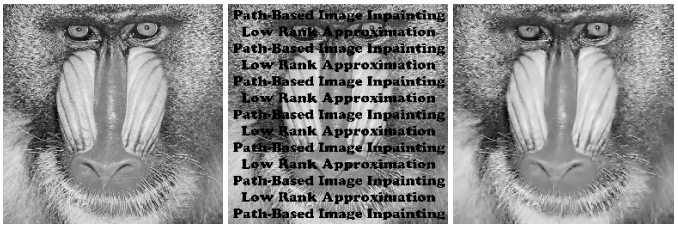
TABLE II
DENOISING RESULTS (SSIM) BY COMPETING METHODS ON THE 12 TEST IMAGES. BEST RESULTS ARE IN BOLD.

σ	20				50			
schemes	BM3D	WNNM	BPFA	KSBMF	BM3D	WNNM	BPFA	KSBMF
Bike	0.887	0.893	0.896	**0.898**	0.688	0.687	0.702	**0.717**
Cameraman	0.872	**0.877**	0.858	0.878	0.747	0.755	0.683	**0.762**
Einstein	0.801	**0.807**	0.803	0.807	0.696	0.699	0.638	**0.700**
Flower	0.874	**0.885**	0.878	0.883	0.716	0.724	0.678	**0.737**
Hat	0.876	0.883	0.858	**0.885**	0.767	0.776	0.667	**0.782**
House	**0.869**	0.871	0.864	0.867	0.812	0.826	0.731	**0.830**
Monarch	0.923	**0.930**	0.914	0.927	0.824	0.829	0.790	**0.832**
Parrot	0.867	0.868	0.852	**0.872**	0.757	0.750	0.699	**0.762**
Peppers	0.890	**0.894**	0.876	0.892	0.786	0.788	0.729	**0.794**
Starfish	0.870	0.885	0.870	**0.890**	0.725	0.720	0.707	**0.739**
Baboon	0.722	**0.730**	0.707	0.728	0.469	0.508	0.486	**0.513**
Barbara	0.909	**0.915**	0.907	0.912	0.762	0.785	0.773	**0.786**
Average	0.863	**0.870**	0.857	0.870	0.729	0.737	0.690	**0.746**
σ	70				100			
schemes	BM3D	WNNM	BPFA	KSBMF	BM3D	WNNM	BPFA	KSBMF
Bike	0.588	0.553	0.586	**0.618**	0.468	0.399	0.471	**0.495**
Cameraman	0.677	0.679	0.585	**0.696**	0.592	0.617	0.463	**0.624**
Einstein	0.646	0.637	0.595	**0.661**	0.592	0.569	0.464	**0.592**
Flower	0.623	0.640	0.585	**0.647**	0.505	0.552	0.466	**0.562**
Hat	0.732	0.738	0.596	**0.745**	0.689	0.683	0.495	**0.691**
House	0.778	**0.795**	0.652	0.794	**0.729**	0.726	0.654	0.728
Monarch	0.758	0.766	0.695	**0.771**	0.649	0.684	0.593	**0.695**
Parrot	0.685	0.693	0.606	**0.702**	0.599	0.624	0.526	**0.633**
Peppers	**0.739**	0.730	0.654	0.736	0.673	0.657	0.559	**0.681**
Starfish	0.652	0.623	0.630	**0.673**	0.556	0.484	0.517	**0.571**
Baboon	0.440	0.467	0.463	**0.472**	0.406	0.448	0.426	**0.452**
Barbara	0.685	0.701	0.692	**0.708**	0.658	0.683	0.656	**0.690**
Average	0.667	0.669	0.612	**0.685**	0.593	0.594	0.524	**0.618**

TABLE III
PSNR AND SSIM VALUES BY INPAINTING METHODS ON PART OF TEST IMAGES FOR DIFFERENT TASKS

Task		BPFA	GSR	TSLRA	KSBMF
Random	10%	21.77/0.8193	22.36/0.8674	22.13/0.8385	**22.52/0.8695**
	20%	22.30/0.8794	22.66/0.9004	25.34/0.9194	**25.73/0.9208**
	30%	26.23/0.9211	28.75/0.9452	27.85/0.9313	**28.89/0.9488**
	40%	29.76/0.9378	30.76/0.9587	30.13/0.9510	**30.87/0.9596**
Text	Mask 1	25.53/0.8793	25.62/0.8867	26.09/0.9146	**26.21/0.9165**
	Mask 2	33.72/0.9101	**36.25/0.9205**	34.94/0.9142	35.73/0.9198
	Mask 3	28.02/0.8429	33.29/0.9197	32.93/0.8956	**33.39/0.9223**
	Mask 4	27.70/0.8187	33.24/0.8832	22.90/0.7532	**33.72/0.8911**

GSR based on group sparse learning [52], and TSLRA based on nuclear norm minimization [49]. We test the algorithms on recovering random missing pixels with four different observed percentage, i.e., 10%, 20%, 30% and 40%. The first two experiments are performed on the Barbara image and the latter two are performed on the Monarch image. The PSNR and SSIM values of the results obtained by four algorithms to recover random missing pixels are displayed in Table III. It is clear that KSBMF achieves the highest PSNR and SSIM values in all the four random missing pixels filling tasks. Overall, BPFA behaves the worst for random pixels missing with the lowest PSNR and SSIM values. For the visual quality comparisons, Fig. 5 shows the results to recover image Barbara by the competing methods from only 20% random samples. The rich textures of Barbara are well recovered by KSBMF and TSLRA with better visual quality than the other two methods. However, BPFA and GSR introduce some incorrect textures with visual artifacts, which is clearly visible on scarf and pants. Fig. 6 shows another example of recovering image Monarch with smooth structures from 40% random samples. KSBMF is competitive in visual quality with TSLRA and GSR, and clearly superior to BPFA. The visual result provided



(a) Original (b) With Text 1 (c) BPFA

(d) GSR (e) TSLRA (f) KSBMF

Fig. 7. Visual comparison for text removal on Baboon. (a) Original image. (b) Image with text mask 1. (c) BPFA (PSNR=25.53 dB). (d) GSR (PSNR=25.62 dB). (e) TSLRA (PSNR=26.09 dB). (f) KSBMF (PSNR=**26.21** dB).

(a) Original (b) With Text 2 (c) BPFA

(d) GSR (e) TSLRA (f) KSBMF

Fig. 8. Visual comparison for text removal on Einstein. (a) Original image. (b) Image with text mask 2. (c) BPFA (PSNR=33.72 dB). (d) GSR (PSNR=**36.25** dB). (e) TSLRA (PSNR=34.94 dB). (f) KSBMF (PSNR=35.73 dB).

by BPFA has some artifacts and blurred edges.

We then apply KSBMF to remove the text on two grayscale images, and further two colour images. The performances of competing algorithms regarding PSNR and SSIM are summarized in Table III. KSBMF outperforms all three competitive algorithms on recovering text-corrupted Baboon, Kid, and Castle images. In regards to the corrupted Einstein image, KSBMF is pretty competitive to GSR with the former inferior to the latter only 0.13 dB. Fig. 7-10 show the visual comparisons of these inpainting algorithms on text-corrupted Baboon, Einstein, Kid and Castle images, respectively. Similar to filling the random missing pixels, it is observed that KSBMF achieves the best overall visual effect with less noise and reconstruction artifacts than competing approaches. However,

(a) Kid Original (b) With Text (c) BPFA

(d) GSR (e) TSLRA (f) KSBMF

Fig. 9. Visual comparison for text removal on Kid. (a) Original image. (b) Image with text mask 3. (c) BPFA (PSNR=28.02 dB). (d) GSR (PSNR=33.29 dB). (e) TSLRA (PSNR=32.93 dB). (f) KSBMF (PSNR=**33.39** dB).

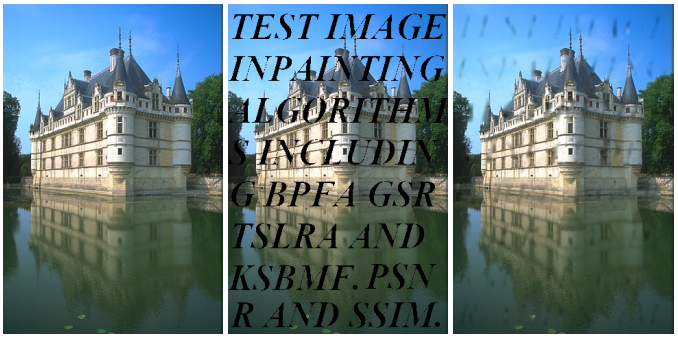
BPFA and TSLRA can hardly remove all texts with some visible stains on the recovered Castle image.

We further test the algorithm to perform a relatively larger size region inpainting. Fig. 11(a) is the original image Kid and Fig. 11(b) shows the image contaminated by three separate blocks. KSBMF and GSR perform well to recover the original blocks with PSNR over 34 and decent visual quality. However, BPFA and TSLRA fail to recover the contaminated image with the black blocks clearly visible and the PSNRs for both algorithms are below 20.

Finally, we consider a more complicated scenario, that is, to recover the noisy image with random missing pixels. Fig. 12(b) shows the image House with 80% random missing pixels and contaminated by Gaussian white noise with $\sigma = 10$. Since GSR and TSLRA are not able to accomplish such image inpainting task, panels of Fig. 12(c) and (d) show the recovered image using BPFA and KSBMF, respectively. KSBMF outperforms BPFA with slightly higher PSNR. In addition, one can observe that KSBMF recovers the edges better than BPFA. In summary, KSBMF performs better than three other competitive methods in four different image completion tasks with the best overall performance.

VI. Conclusions

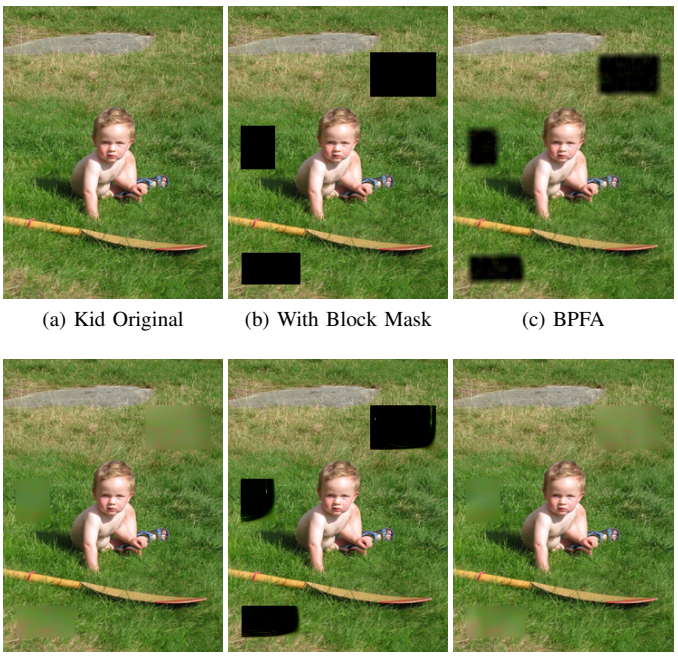
We have presented a new generative model for Bayesian matrix factorization which enables the incorporation of side information through kernel learning. We applied a variational Bayesian learning principle to approximately compute posterior distributions of all parameters and latent variables of the model, in which the low-rank constraint is imposed on the estimation by using sparse representation. Given the nonlocal



(a) Castle Original (b) With Text (c) BPFA

(d) GSR (e) TSLRA (f) KSBMF

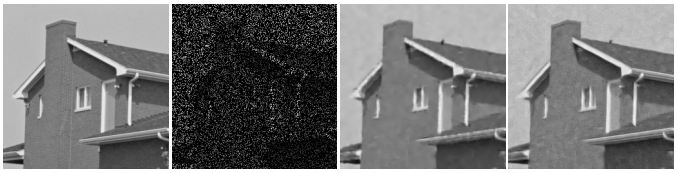
Fig. 10. Visual comparison for text removal on Castle. (a) Original image. (b) Image with text mask 4. (c) BPFA (PSNR=27.70 dB). (d) GSR (PSNR=33.24 dB). (e) TSLRA (PSNR=22.90 dB). (f) KSBMF (PSNR=**33.72** dB).



(a) Kid Original (b) With Block Mask (c) BPFA

(d) GSR (e) TSLRA (f) KSBMF

Fig. 11. Visual comparison for block removal on Kid. (a) Original image. (b) Image with block mask. (c) BPFA (PSNR=19.40 dB). (d) GSR (PSNR=**34.79** dB). (e) TSLRA (PSNR=17.30 dB). (f) KSBMF (PSNR=34.50 dB).



(a) Original (b) Noisy (c) BPFA (d) KSBMF

Fig. 12. Visual comparison for recovery of noisy image House with random missing pixels. (a) Original image. (b) Image with 20% random samples and contaminated by Gaussian white noise with $\sigma = 10$. (c) BPFA (PSNR=29.45 dB). (d) KSBMF (PSNR=**29.71** dB).

similarity and low rankness properties of the patch group matrix, we have further developed two image restoration algorithms which leverage KSBMF under the nonlocal framework. We particularly devise a new kernel to integrate the similarity information between patches into the parameter learning for image denoising and inpainting. The experimental results on three tasks have demonstrated the superiority of KSBMF over not only the conventional Bayesian matrix factorization model but also other state-of-the-art image restoration algorithms. If an image does not possess the low-rank property, the pre-complete step in our inpainting algorithm may result in a relatively large error, degrading the final inpainting quality at the patch level. To avoid this limitation, the first step of inpainting on the entire image can be replaced by an alternative method, for example, total variation based regularization [53], to pre-complete the whole image for accurate patch matching. Then applying KSBMF on the patch group matrix can still guarantee to fill the missing entries accurately. Only Gaussian noise is considered in this study. The model may be extended to a robust version with an extra term to represent outliers, i.e., $\mathbf{Y} = \mathbf{X} + \mathbf{S} + \mathbf{E}$. The sparse component can be modelled by independent Gaussian priors on each of the entries of the matrix $\mathbf{S}$. When an individual precision of s_{ij} goes to infinity, the corresponding entry goes to zero. Hence, the sparsity in $\mathbf{S}$ is achieved when a large number of precision variables are set to high values. In the area of machine vision and image processing, the KSBMF model can be extended to image or video super-resolution, deblurring, and compressed sensing to integrate other appropriate side information, for example, the statistics of offsets of similar patches [54].

Regarding the broad applicability of the proposed model in machine learning, KSBMF is also expected to improve the prediction or completion accuracy over the existing methods that based on the low-rank assumption in recommender systems, documents labels, background subtraction, and so forth. In this study, we have also devised and tested a couple of other kernels including Gaussian function and linear function. The proposed kernel in Eq. (54) yields the best performance for both image denoising and inpainting. However, devising effective kernels to integrate side information for various applications is still an open issue for kernelized matrix factorization in future study.

ACKNOWLEDGMENT

This work is funded by the Australian Research Council (ARC) Laureate Program and the ARC Centre of Excellence

Program. The code associated with this paper will be available online. Xie and Mengersen are the corresponding authors.

REFERENCES

[1] J. Shi, X. Zheng, and W. Yang, "Survey on probabilistic models of low-rank matrix factorizations," *Entropy*, vol. 19, no. 8, p. 424, 2017.

[2] M. A. Davenport and J. Romberg, "An overview of low-rank matrix recovery from incomplete observations," *arXiv preprint arXiv:1601.06422*, 2016.

[3] S. Li and Y. Fu, "Robust subspace learning," in *Robust Representation for Data Analytics*. Springer, 2017, pp. 45–71.

[4] X. Zhou, C. Yang, H. Zhao, and W. Yu, "Low-rank modeling and its applications in image analysis," *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, p. 36, 2015.

[5] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E.-H. Zahzah, "Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset," *Computer Science Review*, vol. 23, pp. 1–71, 2017.

[6] T. Virtanen, J. F. Gemmeke, B. Raj, and P. Smaragdis, "Compositional models for audio processing: Uncovering the structure of sound mixtures," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 125–144, 2015.

[7] L. Yang, X. Chen, Z. Liu, and M. Sun, "Improving word representations with document labels," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 863–870, 2017.

[8] L. Lan, K. Zhang, H. Ge, W. Cheng, J. Liu, A. Rauber, X.-L. Li, J. Wang, and H. Zha, "Low-rank decomposition meets kernel learning: A generalized nyström method," *Artificial Intelligence*, vol. 250, pp. 1–15, 2017.

[9] A. Narita, K. Hayashi, R. Tomioka, and H. Kashima, "Tensor factorization using auxiliary information," *Data Mining and Knowledge Discovery*, vol. 25, no. 2, pp. 298–324, 2012.

[10] R. Forsati, M. Mahdavi, M. Shamsfard, and M. Sarwat, "Matrix factorization with explicit trust and distrust side information for improved social recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 32, no. 4, p. 17, 2014.

[11] W. Fithian, R. Mazumder *et al.*, "Flexible low-rank statistical modeling with missing data and side information," *Statistical Science*, vol. 33, no. 2, pp. 238–260, 2018.

[12] K.-Y. Chiang, C.-J. Hsieh, and I. Dhillon, "Robust principal component analysis with side information," in *International Conference on Machine Learning*, 2016, pp. 2291–2299.

[13] N. Xue, Y. Panagakis, and S. Zafeiriou, "Side information in robust principal component analysis: Algorithms and applications," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4317–4325.

[14] V.-G. Nguyen and S.-J. Lee, "Incorporating anatomical side information into PET reconstruction using nonlocal regularization," *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 3961–3973, 2013.

[15] Z. Zhang, Y. Liu, and Z. Zhang, "Field-aware matrix factorization for recommender systems," *IEEE Access*, vol. 6, pp. 45 690–45 698, 2018.

[16] L. Huang, X. Li, P. Guo, Y. Yao, B. Liao, W. Zhang, F. Wang, J. Yang, Y. Zhao, H. Sun *et al.*, "Matrix completion with side information and its applications in predicting the antigenicity of influenza viruses," *Bioinformatics*, vol. 33, no. 20, pp. 3195–3201, 2017.

[17] V. Y. Tan and C. Févotte, "Automatic relevance determination in nonnegative matrix factorization with the β-divergence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1592–1605, 2012.

[18] V. Renkens and H. V. Hamme, "Automatic relevance determination for nonnegative dictionary learning in the Gamma-Poisson model," *Signal Processing*, vol. 132, pp. 121–133, 2017.

[19] S. Park, Y.-D. Kim, and S. Choi, "Hierarchical Bayesian matrix factorization with side information." in *IJCAI*, 2013, pp. 1593–1599.

[20] M. Gönen, S. Khan, and S. Kaski, "Kernelized Bayesian matrix factorization," in *International Conference on Machine Learning*, 2013, pp. 864–872.

[21] M. H. Alkinani and M. R. El-Sakka, "Patch-based models and algorithms for image denoising: a comparative review between patch-based images denoising methods for additive noise reduction," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, p. 58, 2017.

[22] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2012.

[23] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and Bregman iterative methods for matrix rank minimization," *Mathematical Programming*, vol. 128, no. 1-2, pp. 321–353, 2011.

[24] V. Shah, N. Rao, and W. Ding, "Matrix factorization with side and higher order information," *stat*, vol. 1050, p. 4, 2017.

[25] J. Choo, C. Lee, C. K. Reddy, and H. Park, "Weakly supervised nonnegative matrix factorization for user-driven clustering," *Data Mining and Knowledge Discovery*, vol. 29, no. 6, pp. 1598–1621, 2015.

[26] G. Delmaire, M. Omidvar, M. Puigt, F. Ledoux, A. Limem, G. Roussel, and D. Courcot, "Informed weighted non-negative matrix factorization using $\alpha\beta$-divergence applied to source apportionment," *Entropy*, vol. 21, no. 3, p. 253, 2019.

[27] C. Dorffer, M. Puigt, G. Delmaire, and G. Roussel, "Informed nonnegative matrix factorization methods for mobile sensor network calibration," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 4, pp. 667–682, 2018.

[28] Y. Li and A. Ngom, "Sparse representation approaches for the classification of high-dimensional biological data," *BMC Systems Biology*, vol. 7, no. 4, p. S6, 2013.

[29] N. D. Lawrence and R. Urtasun, "Non-linear matrix factorization with Gaussian processes," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 601–608.

[30] T. Zhou, H. Shan, A. Banerjee, and G. Sapiro, "Kernelized probabilistic matrix factorization: Exploiting graphs and side information," in *Proceedings of the 2012 SIAM international Conference on Data mining*. SIAM, 2012, pp. 403–414.

[31] R. P. Adams, G. E. Dahl, and I. Murray, "Incorporating side information in probabilistic matrix factorization with Gaussian processes," *arXiv preprint arXiv:1003.4944*, 2010.

[32] T. V. Le, R. Oentaryo, S. Liu, and H. C. Lau, "Local Gaussian processes for efficient fine-grained traffic speed prediction," *IEEE Transactions on Big Data*, vol. 3, no. 2, pp. 194–207, 2017.

[33] I. Porteous, A. U. Asuncion, and M. Welling, "Bayesian matrix factorization with side information and Dirichlet process mixtures." in *AAAI*, 2010.

[34] J. Liu, C. Wu, and W. Liu, "Bayesian probabilistic matrix factorization with social relations and item contents for recommendation," *Decision Support Systems*, vol. 55, no. 3, pp. 838–850, 2013.

[35] Y. Xu, Q. Yu, W. Lam, and T. Lin, "Exploiting interactions of review text, hidden user communities and item groups, and time for collaborative filtering," *Knowledge and Information Systems*, vol. 52, no. 1, pp. 221–254, 2017.

[36] H. Yang and J. Wang, "Bayesian hierarchical kernelized probabilistic matrix factorization," *Communications in Statistics-Simulation and Computation*, vol. 45, no. 7, pp. 2528–2540, 2016.

[37] P. Zakeri, J. Simm, A. Arany, S. ElShal, and Y. Moreau, "Gene prioritization using Bayesian matrix factorization with genomic and phenotypic side information," *Bioinformatics*, vol. 34, no. 13, pp. i447–i456, 2018.

[38] M. Zhang and C. Desrosiers, "High-quality image restoration using low-rank patch regularization and global structure sparsity," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 868–879, 2019.

[39] Y.-Q. Zhao and J. Yang, "Hyperspectral image denoising via sparse representation and low-rank constraint," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 1, pp. 296–308, 2015.

[40] S. Chen, H. Liu, Z. Hu, H. Zhang, P. Shi, and Y. Chen, "Simultaneous reconstruction and segmentation of dynamic PET via low-rank and sparse matrix decomposition," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 7, pp. 1784–1795, 2015.

[41] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Sparse Bayesian methods for low-rank matrix estimation," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 3964–3977, 2012.

[42] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[43] P. Lu, B. Gao, W. L. Woo, X. Li, and G. Y. Tian, "Automatic relevance determination of adaptive variational bayes sparse decomposition for micro-cracks detection in thermal sensing," *IEEE Sensors Journal*, vol. 17, no. 16, pp. 5220–5230, 2017.

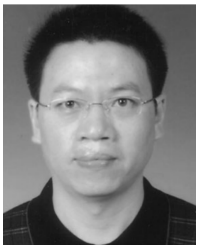
[44] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[45] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[46] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2862–2869.

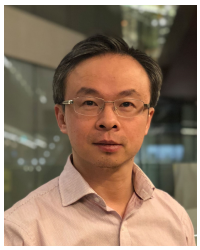
[47] M. Lebrun, A. Buades, and J.-M. Morel, "A nonlocal Bayesian image denoising algorithm," *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1665–1688, 2013.
[48] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug 2007.
[49] Q. Guo, S. Gao, X. Zhang, Y. Yin, and C. Zhang, "Patch-based image inpainting via two-stage low rank approximation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 6, pp. 2023–2036, 2018.
[50] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
[51] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro, and L. Carin, "Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 130–144, 2012.
[52] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3336–3351, 2014.
[53] M. V. Afonso and J. M. R. Sanches, "Blind inpainting using ℓ_0 and total variation regularization," *IEEE Transactions on Image Processing*, vol. 24, no. 7, pp. 2239–2253, July 2015.
[54] K. He and J. Sun, "Image completion approaches using the statistics of similar patches," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2423–2435, 2014.

Caoyuan Li is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Beijing Institute of Technology and also with the Faculty of Engineering and Information Technology, University of Technology Sydney. His research interests include machine learning, pattern recognition and data mining.



Hong-Bo Xie (M'12) received the Ph.D. degree in biomedical engineering from Shanghai Jiao Tong University, Shanghai, China. He is currently a senior research fellow at the ARC Centre of Excellence for Mathematical and Statistical Frontiers, Queensland University of Technology, Brisbane, Australia. His research interests include machine learning, signal and image processing, nonlinear time series analysis, and their applications in biomedical engineering. He has published over 70 peer-reviewed journal and conference papers.

Xuhui Fan received the bachelor's degree in mathematical statistics from the University of Science and Technology of China, Hefei, China, in 2010, and the Ph.D. degree in computer science from the University of Technology Sydney, Chippendale, NSW, Australia, in 2015. His current research interests include stochastic random partition and Bayesian nonparametrics.



Richard Yi Da Xu is currently an Associate Professor with the Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia. He has authored about 50 papers, including IEEE Transactions on Image Processing, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Neural Networks and Learning Systems, Pattern Recognition, ACM Transactions on Knowledge Discovery from Data, Association for the Advancement of Artificial Intelligence, and The International Conference on Image Processing. His current research interests include machine learning, computer vision, and statistical data mining.

Sabine Van Huffel (Fellow, 2009) is a full professor at the Department of Electrical Engineering, KU Leuven, since 2002. She was elected as a Fellow of Royal Flemish Academy of Belgium for Sciences and the Arts in 2017. The research topics of Sabine Van Huffel are fundamental/theoretical as well as application oriented and are performed in the domain of (multi)linear algebra, (non)linear signal analysis, classification and system identification with special focus to the development of numerically reliable and robust algorithms for improving medical diagnostics. Sabine Van Huffel has the following publications record: two monographs, about 304 articles in peer reviewed international journals, about 257 conference papers, 4 edited books, 7 edited journal special issues, 21 book chapters. She was a guest professor at Stanford University, CA, USA and at Uppsala University, Sweden, and a visiting fellow and scientist at the University of Minnesota, MN, USA.

Kerrie Mengersen is a full professor at the School of Mathematical Sciences, Queensland University of Technology, Australia. She was elected as a Fellow of Australian Academy of Science in 2018. She was the National President of the Statistical Society of Australia from 2011 to 2012, and the President of International Society for Bayesian Analysis from 2015 to 2017. In 2016, QUT awarded the title of Distinguished Professor to Professor Kerrie Mengersen in recognition of her outstanding achievements, both nationally and internationally, in mathematics and statistical research. She has over 300 refereed journal publications, over 40 keynote and invited international conference presentations, and attraction of over 30 large research grants.