UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Engineering and Information Technology

Nonparametric Bayesian Models for Signal Processing

by

Caoyuan Li

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Sydney, Australia

2019

Certificate of Authorship/Originality

I certify that this thesis has been written by me. Any help that I have received in my research and in the preparation of the thesis itself has been fully acknowledged. In addition, I certify that all information sources and literature used are quoted in the thesis.

This thesis is the result of the research candidature conducted jointly with Beijing Institute of Technology as part of a collaborative doctoral degree.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed
Signature of Student: prior to publication.

Date: 23/06/2020

ABSTRACT

Nonparametric Bayesian Models for Signal Processing

by

Caoyuan Li

An essential component in signal processing is to remove various kinds of noise from the signal. It is possible to introduce noise during the process of signal storage, transmission and acquisition. Signal quality after denoising affects subsequent signal analysis profoundly. Low-rank representation is a popular method in signal processing. It is aimed to capture underlying low-dimensional structures of high dimensional signal and attracted much attention in the area of the pattern recognition and signal processing. Such successful applications were mainly due to its effectiveness in exploring low dimensional manifolds embedded in data, which can be naturally characterized by low rankness of the data matrix.

This thesis conducts research on processing various signals as well as getting the low-rank representation of the signal via the variational Bayesian inference techniques. This study proposed four different nonparametric Bayesian models for image denoising, inpainting, video foreground/background separation and bio-medical signal processing as follows.

(1) A hybrid denoising model based on variational Bayesian inference and Stein's unbiased risk estimator (SURE) is presented, which consists of two complementary steps. In the first step, the variational Bayesian singular value thresholding (SVT) performs a low-rank approximation of the nonlocal image patch matrix to simultaneously remove the noise and estimate the noise variance. In the second step, the conventional SURE full rank SVT and its divergence formulas for rank-reduced eigen-triplets is modified to remove the residual artefacts.

(2) A hierarchical kernelized sparse Bayesian matrix factorization (KSBMF)

model is developed 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 sparsity through an enforced constraint on latent factor matrices. The KSBMF is further connected with the nonlocal image processing framework to develop two algorithms for image denoising and inpainting.

(3) A robust kernelized Bayesian matrix factorization (RKBMF) model is proposed to decompose a data set into low rank and sparse components. Moreover, the model integrates the side information of similarity between frames to improve information extraction from the video. RKBMF is employed to extract background and foreground information from a traffic video.

(4) A hierarchical Dirichlet process nonnegative matrix factorization (DPNMF) model is presented in which the Gaussian mixture model is used to approximate the complex noise distribution. Moreover, the model is cast in the nonparametric Bayesian framework by using Dirichlet process mixture to infer the necessary number of Gaussian components. A mean-field variational inference algorithm is derived for the proposed nonparametric Bayesian model. The model is tested on synthetic data sets contaminated by Gaussian, sparse and mixed noise. The proposed model is then applied to extract muscle synergies from the electromyographic (EMG) signal and to select discriminative features for motor imagery single-trial electroencephalogram (EEG) classification.

Dissertation directed by Associate Professor Richard Xu
School of Electrical and Data Engineering

Dedication

To my parents and my wife for your love and support.

Acknowledgements

The completion of this dissertation has been possible with the inspiration and encouragement from many people, to whom I am greatly indebted.

Foremost, I would like to express my sincere gratitude to my supervisor Prof. Richard Yi Da Xu, for the continuous support of my PhD study and research in UTS. He is my most admired supervisor. He treats each student as his best friend, trying his best to benefit us and always thinking from the student's perspective. I will never forget the scene where we drink beer together at the bar and the nights we spend together to improve my papers. I gained a lot of knowledge from him, not only the knowledge of machine learning but also experience about how to deal with people and how to survive in the workplace. I believe that I will continue to benefit from this experience for the rest of my life. His guidance helped me in all the time of research and writing of this thesis. I would thank him, for sharing his immense knowledge with me, keep encouraging and motivating me. Without his professional guidance and persistent help, this thesis would not have been possible.

I also would like to appreciate my co-supervisor Dr Xuhui Fan and Dr Hong-bo Xie, for sharing me research ideas and their invaluable experience about research. I wouldn't have finished this thesis without their selfless help.

I thank my fellow labmates in UTS: Xuan Liang, Shuai Jiang, Haodong Chang, Wanming Huang, Wei Huang, etc., for our stimulating discussions and for all the fun we have had in the last several years.

Also, I would like to thank the magic of machine learning. This fantastic world has made me lucky enough to have fun while exploring it during all of my PhD period.

Last but not least, I would like to thank my family: my wife, my parents and

my parents in law, for their unconditional support, both financially and emotionally throughout the whole PhD studying.

Caoyuan Li
Sydney, Australia, 2019.

List of Publications

Journal Papers

J-1. **Caoyuan Li**, Hongbo Xie, Xuhui Fan, Richard Yi Da Xu, et al. Image denoising based on nonlocal Bayesian singular value thresholding and Steins unbiased risk estimator. *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 4899-4911, Oct. 2019

J-2. **Caoyuan Li**, Hongbo Xie, Xuhui Fan, Richard Yi Da Xu, et al. Kernelized sparse Bayesian matrix factorization. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

J-3. **Caoyuan Li**, Hongbo Xie, Kerrie Mengersen, et al. Bayesian nonnegative matrix factorization with Dirichlet process mixtures. *IEEE Transactions on Signal Processing*, 2020.

Conference Papers

C-1. Hongbo Xie, **Caoyuan Li**, Richard Yi Da Xu, et al. Robust kernelized Bayesian matrix factorization for video background/foreground separation. *The Fifth International Conference on Machine Learning, Optimization, and Data Science*, September 10-13, 2019

Contents

List of Figures

List of Tables

Nomenclature and Notation

Example	Description
$\mathbb{R}$	The set of reals
$\mathbf{N}$	the set of natural numbers
$\mathbb{E}[\cdot]$	expection of a random variable
$\langle\cdot\rangle$	expection of a random variable
$\boldsymbol{Y} \in \mathbb{R}^{m \times n}$	Bold and capitalized letters denote random matrices
$\mathbf{u}$	Characters in bold denotes random vectors
	This notation is also used to denote collections of random variables
$\mathbf{a}_{m\cdot}$	m-th row of a matrix
$\mathbf{a}_{\cdot n}$	n-th column of a matrix
y	Characters in italic denote random scalars
$diag(\mathbf{x})$	Diagonal matrix with the values of vector $\mathbf{x}$ on the diagonal.
I_n	The identity matrix of dimension $n \times n$
θ	Parameters of a model are typcally denoted with the Greek lowercase letter θ
$(.)^\top$	denotes the transpose operation
$\mathrm{p}(\mathbf{x})$	Probability density functions (PDFs) and probability mass functions (PMFs)
$\mathrm{p}(\mathbf{x}, \mathbf{y}, \mathbf{z})$	Joint distributions are denoted by $p(\cdot, \cdot)$
$p(\mathbf{x}\|\mathbf{z})$	Condition distributions are denoted by $p(\cdot\|\cdot)$

Chapter 1

Introduction

1.1 Background

A Bayesian nonparametric model is a Bayesian model on an infinite-dimensional parameter space. The parameter space is generally chosen as the set of all possible solutions for a given learning problem. Bayesian nonparametric models have recently been applied to a variety of machine learning problems, such as classification, regression, clustering, source separation, latent variable modeling, image processing and so on. A Bayesian nonparametric model uses a finite subset of the available parameter dimensions to explain a finite sample of observations, with the set of dimensions chosen depending on the sample, such that the effective complexity of the model adapts to the data [1].

Signal processing is an electrical engineering subfield that focuses on analysing, modifying and synthesizing signals such as sound, images and biological measurements [2]. Signal processing techniques can be used to improve transmission, storage efficiency and subjective quality and to also emphasize or detect components of interest in a measured signal [3]. It is an archaic but also active research field, including image signal processing, video signal processing, audio signal processing and so on. In this thesis, four non-parametric probablistic models are proposed to perform tasks including image denoising, inpainting, video foreground/background separation and bio-medical signal processing. Varaitional Bayesian inference is utilized to infer these models efficiently.

The the application of the first nonparametric Bayesian model proposed in this thesis is about image denoising. The research about image denoising has been conducted for decades and it is expected that such a fundamental problem has been solved. For various applications, including astronomy, remote sensing, photography, robotics or medicine, image denoising has been the foundamental process of other subsequent image processing methods. The image quality after denoising affects the subsequent image analysis profoundly. Researchers in various domains have conducted a lot of research in this field. Most of the existing image denoising [4, 5] algorithms require exact noise parameters as input. However, in the actual application scenario, these parameters are often not available, considering they may depend on sensors operational conditions and the calibration data may not be available [6]. The estimated noise parameter affects the performance of the above-mentioned image denoising algorithms heavily. Some noise may remains in the denoised image when the noise level is underestimated, whereas overestimating the noise level results in oversmoothing the output image [7]. For natural images, the quality of denoising results have come close to theoretical limits. State-of-the-art image denoisers preserve every perceivable detail while removing most of the noise. However, all of them suffer from artifacts, particularly visible when in the smooth area or around the edges of objects of an image. The first model is proposed to solve the above mentioned problems, the variational Bayesian singular value thresholding performs a low-rank approximation of the nonlocal image patch matrix to simultaneously remove the noise and estimate the noise parameters. The conventional SURE full rank SVT and its divergence formulas for rank-reduced eigen-triplets is modified to further remove the residual artefacts.

Although the above mentioned model achieved satisfying performance, the similarity information between the image patches is not fully utilized. The second model is proposed to integrate the similarity information into the matrix factorization based nonparametric Bayesian model. Matrix factorization aims to factorize a given matrix $\mathbf{Y}$ into two low-rank latent factor matrices $\mathbf{U}$ and $\mathbf{V}$, so that their product reconstructs the original matrix. Classical factorization methods include nonnegative matrix factorization (NMF), independent component analysis (ICA), principal component analysis (PCA) and sparse component analysis (SCA), among others [8]. 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 [9, 10, 11] and its specific applications in image and video processing [12, 13], audio processing [14]. Although these methods are successful in many areas, most of them simply ignore side information, or intrinsically, are not capable of exploiting it.

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 [15, 16, 17, 18, 19, 20, 21, 22, 23, 24]. 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 the second contribution of this study, a hierarchical kernelized sparse Bayesian matrix factorization model is developed 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 sparsity through an enforced constraint on latent factor matrices. The KSBMF is further connected with the nonlocal image processing framework to develop two algorithms for image denoising and inpainting. This model is further extended to include the sparse components, and is utilized to extract background and foreground information from a traffic video.

In the previous models, the noise contained in the signal is assumed to be following Gaussian distribution. However, in real world applications, the noise could be quite complex, Gaussian distribution is insufficient to deal with complex noise in real scenarios. The fourth model is proposed to fit the complex noise by utilizing Dirichlet process and integrating nonnegative matrix factorization. In many real-life applications, negative factors may contradict physical or physiological reality and lack intuitive meaning. Fortunately, NMF, overcoming the shortcoming of other methods, provides meaningful components with physical or physiological interpretations under the nonnegative constraint. In other words, NMF yields nonnegative factors, which can be advantageous from the point of view of interpretability of the estimated components. Due to the extraordinary effectiveness of NMF in signal processing and machine learning, substantial research effort has been devoted to NMF, both theoretical and applied, to solve challenging problems, including: signal blind separation, hyperspectral unmixing, audio spectra analysis, text mining, image restoration, spectral clustering, and source localization and neural information extraction in neuroscience [8, 25, 26, 27, 28, 29].

In the last contribution of this thesis, a hierarchical Dirichlet process nonnegative matrix factorization (DPNMF) model is presented in which the Gaussian mixture model is used to approximate the complex noise distribution.

1.2 Thesis Organization

This thesis is organised as follows: Chapter 2 provides a comprehensive literature review about the proposed models, Chapter 3 proposes a unified nonlocal image denoising framework based on variational Bayesian inference and Stein's unbiased risk estimator (BSSVT). In Chapter 4, a generic variational Bayesian model is presented for matrix low rank and sparse decomposition with side information. Two algorithms are further developed based on this VB framework for nonlocal image denoising and inpainting. Chapter 5 presents a generative model for robust kernelized Bayesian matrix factorization (RKBMF) which can integrate side information into inference. The performance of the model is tested on simulated datasets and then applied to perform the video background and foreground separation task. Chapter 6 develops a hierarchical Bayesian non-negative matrix factorization model. Gaussian mixture model (GMM), a universal approximator for any continuous distribution, is employed to approximate the complex noise components. Chapter 7 concludes the thesis and gives recommendation for future works.

Chapter 2

Literature Review

The history and tendency of the development of singular value thresholding based image denoising methods are introduced in Section 2.1. The matrix factorization based image processing algorithms are presented in Section 2.2, and the typical non-negative matrix factorization related approaches for signal processing are shown in Section 2.3.

2.1 Singular value thresholding approaches

Singular value thresholding (SVT) aims to recover an approximately low-rank data matrix $\boldsymbol{X}$ from a noisy observation matrix $\boldsymbol{Y}$ by shrinking its singular values (SV). SVT has been widely applied in signal and image processing, computer vision, and pattern recognition. It is well known that, if $\boldsymbol{Y} = \boldsymbol{UDV}^\top = \sum_{i=1}^{\min(m,n)} \lambda_i u_i v_i^\top$ is a singular value decomposition (SVD) for $\boldsymbol{Y}$, the hard thresholding estimator simply truncates the singular spectrum by setting some of the SV to zero. The level of the SV truncating can be determined by cross-validation; however, this approach can be unstable and computationally expensive [30, 31]. Donoho and Gavish [32] proposed an optimal hard threshold of $4/\sqrt{3}/\sqrt{m}\sigma$ for an $m \times m$ square matrix with known noise variance σ^2. Under the framework of nonlocal image denoising, the representative hard threshold algorithms include [33, 34] and the very recent [35].

In contrast to hard thresholding, soft thresholding aims to shrink each SV using the function

$$\hat{\lambda}_i = \lambda_i(1 - \frac{\tau}{\lambda_i})_+, \tag{2.1}$$

where $x_+ = \max(x, 0)$ for $x \in \mathbb{R}$. Candes et al. [36] provided a closed-form expression of Stein's unbiased risk estimate (SURE) to select the threshold $\tau > 0$. Dong et al. [37] extended the principle of wavelet BayesShrink to determine the soft threshold. Their spatially adaptive iterative singular value thresholding (SAIST) method estimates the threshold corresponding to each SV based on the locally estimated signal variance and overall noise variance. To exploit the low-rank structure of the patch matrix, substantial effort has been expended on rank-penalized methods and convex relaxation or, for computational reasons, penalization of the nuclear norm of the matrix. Gu et al. [38] assumed that the noise energy is evenly distributed over each subspace spanned by the eigen-triplets. Specific thresholds are then determined by the individual SV and noise variance. Although this method is termed the weighted nuclear norm minimization (WNNM), it lies in the category of SV soft thresholding methods. Since the algorithms in [37, 38] consider the relative importance of different SVs, the quality of the recovered image is very competitive in terms of the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). Several variants of SAIST and WNNM have been developed [39, 40, 41, 42].

Recently, Josse and Sardy [43] defined a two-parameter threshold function

$$\hat{\lambda}_i = \lambda_i(1 - \frac{\tau^\eta}{\lambda_i^\eta})_+, \tag{2.2}$$

which encompasses hard thresholding for $\eta \to \infty$ and soft thresholding when $\eta = 1$. Their Monte Carlo simulation revealed that such a trade-off between soft and hard thresholding yields the best performance in terms of MSE on both low-rank and general signal matrices across different signal-to-noise ratio regimes. Following the same principle, Verbanck et al. [44] suggested a regularized version of PCA (rPCA) that essentially selects a certain value for the rank and shrinks the corresponding SVs. Jia et al. [45] defined this problem as rank constrained nuclear norm minimization (RNNM), in which the rank and the extent of thresholding are controlled separately. The thresholding function is accordingly denoted as:

$$\hat{\lambda}_i = \begin{cases} \lambda_i(1 - \frac{\tau^\eta}{\lambda_i^\eta})_+ & i = 1, \ldots, r \\ 0 & i = r+1, \ldots, \min(m, n), \end{cases} \tag{2.3}$$

where r is the selected rank with $r < \min(m, n)$.

These methods not only aim to better approximate the original low-rank structure of the patch matrix, but also differentiate the importance of each rank component. Due to this balance between the reduced rank and threshold, these algorithms can achieve superior results compared with benchmark methods such as nonlocal means and BM3D [5].

However, there are a number of issues shared by these existing methods. Firstly, almost all of the aforementioned methods and their variants require the noise variance σ^2 to be known [32, 33, 34, 35, 36, 37, 45, 39, 40, 41, 42, 43, 44], which is not realistic in practice. An extra step is therefore required to pre-determine the noise variance. While the denoising performance of these methods can be substantially degraded when using poor estimates of the noise variance, the numerically impressive results of a number of approaches including BM3D, SAIST, WNNM, and RNNM are obtained simply because of the assumption that the exact noise variance is known [36, 45, 44, 46, 47, 48, 49, 50]. The impact of the error in the estimation of the noise variance on recovered images has not been examined, which casts doubt on the actual performance of these approaches. There are also other free parameters that need to be empirically determined. For example, two extra constants control the weights in WNNM and the pre-specified order in low-rank approximation methods [37, 44, 45]. Furthermore, in order to thoroughly remove the noise, the iterative regularization scheme is frequently adopted in these methods. The variance of the residual noise for the next iteration is estimated from the difference between the initial variance and that of the filtered noise at the previous iteration. The initial error in the estimation of the noise variance therefore propagates and accumulates at each iteration, ultimately degrading the quality indexes in real-world applications. Another issue is that these approaches are largely based on the conventional singular value decomposition (SVD) in the least squares sense. For high-dimensional parameter spaces, the MSE of a least-squares method is often larger than that of a Bayesian estimator [51]. It is also highly susceptible to outlier values in the data. Finally, low-rank approximation-based algorithms, as well as BM3D and WNNM, tend to produce a weak noise-like pattern in low contrast areas of the image when the noise level is moderate or high [37, 52]. This is because the noise in similar patches is partially correlated, which can lead to the incorrect estimation of low-rank patterns as the output of these algorithms [46].

To address the many issues identified above, a unified nonlocal image denoising framework is proposed based on variational Bayesian inference and Stein's unbiased risk estimator (BSSVT). This generic nonlocal denoising framework consists of two complementary steps. In the first step, the variational Bayesian model performs a low-rank approximation of the noisy patch matrix. This is functionally equivalent to other low-rank approximation or nuclear norm minimization methods. More importantly, the noise variance is a latent parameter which is automatically inferred, so does not need to be provided beforehand. The SURE criterion has been employed in a variety of denoising problems to optimize regularization parameters for minimizing the estimation risk or MSE [53, 54, 55, 56, 57]. With the noise variance obtained via the Bayesian model, the second step carries out the SURE-based singular value thresholding on the rank-reduced eigen-triplets to optimally refine the SVs. This further attenuates the very weak noise-like pattern in low contrast areas of the image and reduces artefacts around edges, overcoming the shortcomings of low-rank approximations [46].

The first part of the proposed method is related to Bayesian approaches for orthogonal matrix low-rank approximation and for orthogonal nonnegative matrix factorization [51, 58, 59, 60, 61, 62, 63, 64]. Hoff [51] presented a full Bayesian singular value decomposition model. However, using Gibbs sampling to estimate the parameters makes it unsuitable for nonlocal image denoising because of its huge computational cost. The singular value may also be negative in this model, leading to further issues. The Bayesian inference on the unknown parameters in [59] was also carried out using Markov chain Monte Carlo (MCMC), while the variational Bayesian PCA algorithm in [58] focused on feature extraction and reduction. Variational inference was employed in [60] to perform SVD; however, this model only considered a prior of a singular vector and omitted singular values. Although [61, 62, 63, 64] emphasize orthogonality in their models, the basic framework of these models is nonnegative matrix decomposition.

As reviewed above, most image denoising approaches are developed based on the assumption of a known noise variance [32, 33, 34, 35, 36, 37, 45, 39, 40, 41, 42, 43, 44]. This largely restricts them in terms of practical use. Consequently, the first step of image denoising is often dedicated to estimating the noise variance using the same available image that needs to be denoised. The most well-known noise variance estimator is the scaled Median Absolute Deviation (MAD) method in wavelet denoising [65]. The noise variance is roughly approximated by the median of the absolute value of the wavelet coefficients at the finest decomposition level, which is employed in [37] and its variants. Other methods to estimate the noise variance are mainly based on residual principal components or singular values [66, 67, 68, 69, 70]. It is very common for the noise variance or precision to be estimated via generative models [51]. However, this has not attracted enough attention to be exploited in the image processing community. This is the first research to present a variational Bayesian model to shrink SVs for nonlocal denoising, and in particular, simultaneous noise removal and noise variance estimation.

Low-rank approximations tend to produce a very weak noise-like pattern in flat areas of the image when the noise level is moderate or high [46]. This arises from the fact that the noise in a group of overlapping similar patches is partially correlated, which can incorrectly lead to the reconstruction of a low-rank approximation. The SURE criterion has been well developed to optimally adjust the parameters of a variety of denoising algorithms for edge-preserving filtering and artefact removal [36, 43, 53, 54, 55]. However, the existing SURE is only applicable to shrinking the full rank eigen-triplets. In Chapter 3, the proposed BSSVT method modifies the existing SURE and its divergence formulas to accommodate the rank-reduced eigen-triplets obtained by Bayesian low-rank approximation.

2.2 Matrix Factorization Approaches

Using machine learning methods to find the low-rank and/or sparse approximation of a given data matrix is a fundamental problem in many computer vision applications, for example, background/foreground separation. By casting the problem into the penalization of the regularization term, a number of efforts have been devoted to applying convex or non-convex optimization methods to obtain the low rank and sparse components [71, 72, 73, 74]. For most of these convex or non-convex methods, one has to manually choose some regularization parameters to properly control the trade-off between the data fitting error and the matrix rank when noise is involved. However, due to the lack of noise variance and rank, it is often unrealistic to determine the optimal regularization parameters.

Bayesian inference under probabilistic frameworks provides another essential principle to perform matrix factorization. Ding et al. [75] proposed a Bayesian robust principal component analysis (BRPCA) framework which infers an approximate representation for the noise statistics while simultaneously inferring the low rank and sparse components. However, this model is relatively complex, and the intractable posteriors are inferred by Gibbs sampling. Aicher [76] later improved the parameter inference in [75] by using the factorized variational Bayesian (VB) principle. Wang et al. [77] proposed a Bayesian robust matrix factorization model for image and video analysis. The Gaussian noise model is replaced by a Laplace mixture in [77] to enhance model robustness. Similarly, a Bayesian formulation of hierarchical L_1 norm low-rank matrix factorization is presented in [78]. In addition, Zhao et al. [79] presented a generative robust PCA model under the Bayesian framework with data noise modeled as a mixture of Gaussians (MoG). A common issue of the above models is that the optimal rank of the low rank component has to be manually pre-determined, which potentially either over-fits or under-fits the data. Babacan et al. [80] proposed to employ the automatic relevance determination principle in sparse Bayesian learning to determine the optimal rank of the low rank component.

Although these methods are successful in many areas including video processing, most of them simply ignore side information, or intrinsically, are not capable of exploiting it. On the other hand, many studies have indicated that kernelized matrix factorization to integrate side information, i.e., prior knowledge or data attributes for specific data, can significantly improve the performance of information extraction or prediction [18, 16, 81, 82]. However, the inference of kernelized matrix factorization models using VB is still quite limited. Pork et al. [83] 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 Cramer-Rao bound. Very recently, Gönen and Kaski [84, 85] 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. Moreover, both models in [84, 85], and [83] lack of robustness, which is required to handle the sparse component or outliers in many real-world applications.

In order to incorporate the document labels into the matrix factorization model to improve word representations for the text classification task, Yang et al. [15] 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. [16] 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 [16] is based on the Frobenius norm, the same as in [15], to measure the closeness between two matrices. Narita et al. [17] 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 [18] 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 [19] 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 [20, 21]. Nguyen and Lee [22] 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 [23] to learn the latent matrix, where the interactions between user/item and field can be captured. Huang et al. [24] 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 [86, 87]. While the aforementioned methods incorporated explicit side information in the low-rank matrix factorization setting, Shah et al. [81] 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. [88] 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. [89] 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. [90] further assumed that the columns of a matrix factor have a sparse decomposition along with a known dictionary. Besides, the idea of a convex NMF in [91] is similar to [89, 90]. However, the update rules are derived by the majorization-minimization algorithm. In another family of sparse representation [91], 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 [81], 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 [92]. Inspired by this idea, Zhou et al. [82] 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. [93] 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. [94] 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. [95] 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 [95]. 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. [96] 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. [97] 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 [98] 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 [95, 96, 97, 98] are all inferred using Gibbs sampling. Zakeri et al. [99] 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. [83]

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 [85] 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 [100, 101, 102]. It has been proven that the adoption of suitably combined constraints of low rankness and sparsity is expected to yield substantially enhanced estimation results [100, 101, 102]. 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. However, the proposed KSBMF model in Chapter 4 addresses all these issues together.

2.3 Non-Negative Matrix Factorization Methods

Following Lee and Seung's seminal paper on NMF published in Nature [103] in 1999, these authors later presented two further algorithms based on multiplicative updates to minimize the cost functions based on the Frobenius norm and generalized KL-divergence, respectively [104]. In practice, the observed data inevitably contains noise and outliers. Several extensions of [104] either modifying the cost function or update rule, or imposing extra constraints, have been proposed to avoid degrading the performance. Cichocki et al. [105] extended the generalized KL-divergence to generalized Alpha-Beta divergences, which were parameterized by the two tuning parameters α and β. By adjusting these tuning parameters, this generalized family of $\alpha\beta$-multiplicative NMF algorithms can improve robustness concerning noise and outliers. Considering the sparseness of the considerable additive noise, imposing an ℓ_1-norm term into the objective function is a popular method to achieve robust NMF [106]. Kong et al. [107] proposed a robust formulation of NMF based on the mixed $\ell_{2,1}$-norm, which is trained using multiplicative updates. Huang et al. [108] incorporated a manifold regularization term into the model in [107] to encode the geometrical information existing in the data. Du et al. [109] proposed a robust NMF method (CIMNMF) based on the correntropy induced metric, which is much more insensitive to outliers. A half-quadratic minimization algorithm was developed to optimize the non-convex loss function iteratively. Recently, Shen et al. [110] explored a robust NMF which used the hyperbolic tangent ($tanh$) function as a robust loss to evaluate the reconstruction error.

While NMF originated from optimizing a suitable cost function subject to non-negativity constraints, it is well-known that most popular NMF cost functions can be interpreted as the maximum likelihood (ML) estimation of statistical models. For instance, the ℓ_2-norm distance measure is related to Gaussian error statistics, while KL- or IS-divergence can be approximated by alternative error statistics given by Poisson or Gamma distributed noise kernels. Hence, constrained optimization of proper cost functions can be achieved within a statistical framework in terms of maximum likelihood estimation [111]. This results in the development of more conceptually principled approaches based on Bayesian probabilistic interpretations of NMF.

To take advantage of conjugate distributions for more straightforward Bayesian inferences under a nonnegative constraint, a Poisson likelihood or noise function accompanied by a Gamma prior for $\boldsymbol{U}$ and $\boldsymbol{V}$ is one of the most popular Bayesian NMF models [112]. This model and its extensions have resulted in many emerging real-life applications, e.g., image restoration [112], recommendation systems [113], audio source separation [114], and speech enhancement [115]. Since an exponential distribution can be viewed as a special case of the Gamma distribution, Vincent and Hugo [116] used the former to replace the latter to couple with a Poisson likelihood. They also applied automatic relevance determination (ARD) to determine the model order to avoid overfitting. However, the Poisson distribution is formally defined only for integers, which impairs the statistical interpretation of KL-NMF on uncountable data such as real-valued signals or images.

Moreover, as pointed out by Chien and Yang [117], some dependency of the variational lower bound on model parameters was ignored in the original Bayesian Poisson-Gamma NMF model in [112], so the inferred parameters did not reach the true optimum of the variational objective. To circumvent these issues, Schachtner et al. [118] developed a variational Bayesian NMF model with Gaussian likelihood, and a truncated normal distribution as the prior of factors by truncating all negative entries and renormalizing the integral to unity. This algorithm is a straightforward Bayesian generalization of the canonical Lee and Seung methods in the case of a Euclidean distance measure (ℓ_2-norm) for the reconstruction error corresponding to a Gaussian noise kernel. They later modified the prior of the one-factor matrix by deliberately adding a delta peak at the origin of the truncated normal distribution to accommodate both Gaussian and sparse noise [111]. However, this model slightly over-estimates the number of sources under low noise levels, while in the case of high noise levels, the method may fail completely [111]. Since both exponential and Gaussian distributions belong to the exponential family, the truncated normal can be replaced by exponential distribution to formulate a similar Bayesian NMF update formula [119, 120]. In order to integrate prior knowledge about the factor matrices, suitable prior distributions, like Gaussian processes [121] and Gamma chain priors [122], have also been incorporated into the models.

Although these NMF models, based on either regularization or Bayesian inference, have gained a certain degree of success in various signal processing and machine learning applications, a common shortcoming of most of these methods is that they consider only a single noise kernel or noise distribution. In practice, it is well known that most observations contain complex signal and noise components. For example, physiological signals including electroencephalogram (EEG) and electromyogram (EMG) are weak bioelectric recordings contaminated by white Gaussian noise, motion artefact, cross-talk, power line interference, as well as spurious background spikes [123, 124]. These existing models are insufficient to identify the nonnegative factors or extract signal components accurately. To address this issue, this study aims to develop a hierarchical Bayesian non-negative matrix factorization model. A Gaussian mixture model (GMM), a universal approximator for any continuous distribution [125], is employed to approximate the complex noise components. Accompanying the GMM, another issue is how to choose the number of mixture components, which impacts the generalizability of the model heavily. Insufficient components result in under-fitting, while an excessive number of components leads to over-fitting.

The proposed DPNMF model in Chapter 6 is somewhat related to both GMM and nonparametric Bayesian models. GMM has a long history in fitting observed data due to its modelling and approximation properties [126, 127]. Penny et al. [128] proposed to model noise as a mixture of Gaussians (MoG) rather than drawn from a specific distribution in a general linear model (GLM). However, their model is for time series regression, and the model order, i.e., the required number of Gaussian components, has to be pre-specified. Although there are some criteria such as Akaike information criterion (AIC), Bayesian information criterion (BIC), and minimum message length (MML) available to determine the model order, these kind of model selection approaches can be highly computationally demanding, as they need to traverse all candidate numbers of components. The Dirichlet process mixture model (DPMM) assumes that the data is generated from an infinite number of components, model selection and parameter learning are simultaneously performed within one training round [129]. Recently, Shao et al. [130] presented a Dirichlet process mixture of Gaussians to predict chemical processes. Ren et al. [131] introduced a Dirichlet process mixture of principal components to properly choose the latent dimension number of the GLM problem in [128]. Lack of conjugacy due to the nonnegativity constraint makes the inference of the proposed model very different from the conjugate distributions in [130, 131]. The Dirichlet process mixture of Gaussians is employed to model the noise term in the proposed hierarchical model while GMM and/or DPMM are employed to model the observation itself in [128, 130, 131]. Under the more general framework of nonparametric Bayesian NMF, Porteous et al. [95] assumed that there are latent classes for the entities of $\mathbf{Y}$ and regularization should be performed per class. To this end, they used a Dirichlet process mixture to automatically prune the clusters of latent vectors which dominate the posterior in a collaborative filtering task [95].

Similarly, Xuan et al. [132] proposed a doubly sparse nonparametric NMF framework where dependent Indian buffet processes (dIBP) [133] were used to generate two stick weights associated with each column pair of factor matrices while still maintaining their respective marginal distribution specified by IBP. As a consequence, the generation of two-factor matrices is both nonparametric and sparse. However, the nonparametric Bayesian approach works on latent vectors in [95, 132] rather than the noise term in the proposed model.

In Chapter 6, the power of a nonparametric Bayesian technique, i.e., Dirichlet process mixtures, is utilized to determine the required number of Gaussian components. The model is thus termed Dirichlet process nonnegative matrix factorization (DPNMF).

Chapter 3

Image denoising based on nonlocal Bayesian singular value thresholding and Stein's unbiased risk estimator

3.1 Introduction

Singular value thresholding (SVT) or nuclear norm minimization (NNM)-based nonlocal image denoising methods often rely on the precise estimation of the noise variance. However, most existing methods either assume the noise variance is known or require an extra step to estimate it. Under the iterative regularization framework, the error in the noise variance estimate propagates and accumulates with each iteration, ultimately degrading the overall denoising performance. In addition, the essence of these methods is still least squares estimation, which can cause a very high Mean Squared Error (MSE) and is inadequate for handling missing data or outliers. In order to address these deficiencies, this chapter presents a hybrid denoising model based on variational Bayesian inference and Stein's unbiased risk estimator (SURE), which consists of two complementary steps. In the first step, the variational Bayesian singular value thresholding performs a low-rank approximation of the nonlocal image patch matrix to simultaneously remove the noise and estimate the noise variance. In the second step, the conventional SURE full rank SVT and its divergence formulas for rank-reduced eigen-triplets is modified to remove the residual artefacts. The proposed hybrid BSSVT method achieves better performance in recovering the true image compared with state-of-the-art methods.

The main contributions are summarized as follows: (a) A hybrid nonlocal image blind denoising framework is formed which exploits both Bayesian low-rank approximation and Stein's unbiased risk estimation. (b) A variational Bayesian model is adopted to approximate the low-rank structure of the patch matrix, which simultaneously performs the noise removal and noise variance estimation. This Bayesian model was first developed in [58], with a focus on general principal component analysis. In this chapter, its construction is applied and extended for image processing applications. Since the original model in [58] needs to try out all possible values of the rank to determine the reduced rank, the huge computational burden makes the model non-viable for patch-based image restoration tasks. The automatic relevance determination principle [134] is employed to automatically prune the rank, which significantly relieves the computational cost. (c) The full-rank Stein's unbiased risk estimator and its divergence formulas is modified for use in reduced-rank singular value thresholding. This modified SSVT algorithm directly maximizes the PSNR by refining the optimal threshold that minimizes the MSE estimation of rank-reduced eigen-triplets. (d) The modified SURE model is apolied on the rank-reduced eigen-triplets to enhance the initial low-rank approximation and to produce a more precise estimate of the original image.

The experimental results demonstrate that the proposed BSSVT approach has superior performance in comparison with the state-of-the-art methods in terms of both PSNR and SSIM.

The rest of this chapter is organized as follows. In Section 3.2, the details of the Bayesian model is elaborated for low-rank patch recovery in the presence of noise. The nonlocal Stein's unbiased risk estimator is also described in this section. Experimental results, comparison with the state-of-the-art methods and objective assessments are presented in Section 3.3. Finally, the Section 3.4 discusses and concludes this chapter.

3.2 BSSVT model and inference

The proposed BSSVT method consists of two successive and complementary steps: Bayesian singular value thresholding (BSVT) for low-rank approximation representation of nonlocal similarities; and the singular value thresholding based on SURE (SSVT) with respect to the rank-reduced representation. Fig 3.1 shows a schematic diagram of BSSVT, in which the leftmost component is the graphic model of the Bayesian SVD and the rightmost component represents the SURE-based shrinker. The details of these steps are presented below.

3.2.1 Variational Bayesian singular value thresholding

Under the nonlocal framework, an image is divided into small square blocks, i.e. patches. A patch group matrix is constructed by the vectorization of each patch and its nonlocal neighbors. The final output image is formed by reassembling the individually processed patches. The purpose of variational Bayesian singular value thresholding is to learn this low-rank subspace, while simultaneously providing the noise variance and eigen-triplets for refinement at the second stage.

Model Specification

Without loss of generality, assume that the noisy patch matrix is $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{E}$, where $\boldsymbol{Y} \in \mathbb{R}^{n\times m}$ is composed of n vectorized similar patches with size $\sqrt{m} \times \sqrt{m}$ from a noisy image and $\boldsymbol{E}$ denotes the noise matrix with i.i.d. entries $\boldsymbol{E}_{i,j} \sim \mathcal{N}(0, \omega^{-1})$, where $\mathcal{N}(0, \omega^{-1})$ denotes a Gaussian distribution with mean 0 and precision ω. A natural way to represent the low-rank subspace is to truncate the singular values of the observed matrix $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top = \sum_{i=1}^{\min(m,n)} \lambda_i u_i v_i^\top$ to $\hat{\boldsymbol{X}} = \boldsymbol{U}_r\boldsymbol{D}_r\boldsymbol{V}_r^\top$ (for $r < \min(m,n)$), which satisfies $\boldsymbol{U}_r \in \mathbb{R}^{n\times r}$, $\boldsymbol{V}_r \in \mathbb{R}^{m\times r}$ and $\boldsymbol{U}_r^\top\boldsymbol{U}_r = \boldsymbol{I}_r$, $\boldsymbol{V}_r^\top\boldsymbol{V}_r = \boldsymbol{I}_r$. Here, $\boldsymbol{D}_r = diag(\boldsymbol{\lambda}_r) \in \mathbb{R}^{r\times r}$ is a diagonal matrix with non-zero singular values in descending order and r is the rank of the low-rank approximation.

From the point of view of Bayesian inference, the task is to infer the posterior eigen-triplets of $\boldsymbol{U}_r, \boldsymbol{D}_r$ and $\boldsymbol{V}_r$ from their prior distributions and the observed patch matrix. The likelihood of the noisy patch matrix is denoted as:

$$p(\boldsymbol{Y}|\boldsymbol{U}_r, \boldsymbol{D}_r, \boldsymbol{V}_r, \omega, r) = \mathcal{N}(\boldsymbol{U}_r\boldsymbol{D}_r\boldsymbol{V}_r^\top, \omega^{-1}\boldsymbol{I}_n \otimes \boldsymbol{I}_m), \tag{3.1}$$

where $\boldsymbol{I}_n$ denotes an $n \times n$ identity matrix and $\boldsymbol{I}_n \otimes \boldsymbol{I}_m \in \mathbb{R}^{nm \times nm}$ denotes the Kronecker product of matrices $\boldsymbol{I}_n$ and $\boldsymbol{I}_m$.

Since $\boldsymbol{U}_r$ has orthonormal columns, it is constrained to the Stiefel manifold $\mathcal{S}_{n,r}$ [51]. Therefore, both the prior and posterior distributions of $\boldsymbol{U}_r$ have a support confined to $\mathcal{S}_{n,r}$. The finite area $C(n, r)$ of $\mathcal{S}_{n,r}$ is given by [58, 59, 60]

$$C(n, r) = \frac{2^r \pi^{(1/2)nr}}{\pi^{(1/4)r(r-1)} \prod_{j=1}^{r} \Gamma((1/2)(n - j + 1))}, \tag{3.2}$$

where $\Gamma(\cdot)$ is the gamma function. Similarly, $\boldsymbol{V}_r$ is constrained to the manifold $\mathcal{S}_{m,r}$. The priors on $\boldsymbol{U}_r$ and $\boldsymbol{V}_r$ are adopted to be the least informative, i.e. uniform on $\mathcal{S}_{n,r}$ and $\mathcal{S}_{m,r}$, respectively.

$$p(\boldsymbol{U}_r) = C(n, r)^{-1}\chi(\mathcal{S}_{n,r}), \tag{3.3}$$

$$p(\boldsymbol{V}_r) = C(m, r)^{-1}\chi(\mathcal{S}_{m,r}), \tag{3.4}$$

where $\chi()$ denotes the indicator function on the argument set.

In the absence of specific prior knowledge on ω, Jeffreys' prior is utilized for the precision parameter so that

$$p(\omega) \propto \omega^{-1}, \tag{3.5}$$

which corresponds to an improper gamma distribution attained when both shape and scale parameters approach zero [135]. The above uninformative priors can be modified in obvious ways if relevant information is available.

The prior knowledge of $\boldsymbol{D}_r$ can be expressed by an upper bound on the norm of $\boldsymbol{\lambda}_r$:

$$\sum_{i=1}^{r} \lambda_i^2 \leq 1, \tag{3.6}$$

together with the descending order constraint, so that $\boldsymbol{\lambda}_r$ is confined to the space

$$\mathcal{L}_r = \{\boldsymbol{\lambda}_r | \lambda_1 > \lambda_2 > \cdots > \lambda_r > 0, \sum_{i=1}^{r} \lambda_i^2 \leq 1\}, \tag{3.7}$$

which is a segment of the unit hyperball $\mathcal{H}_r$. The volume of $\mathcal{L}_r$ is:

$$\mathcal{V}_r = h_r \frac{1}{2^r(r!)} = \frac{\pi^{r/2}}{\Gamma(r/2+1)2^r(r!)}. \tag{3.8}$$

where h_r is the volume of $\mathcal{H}_r$. The prior distribution on $\boldsymbol{\lambda}_r$ is then chosen to be uniform on $\mathcal{L}_r$ [58, 135]:

$$p(\boldsymbol{\lambda}_r) = \mathcal{U}(\mathcal{L}_r) = \mathcal{V}_r^{-1}\chi(\mathcal{L}_r). \tag{3.9}$$

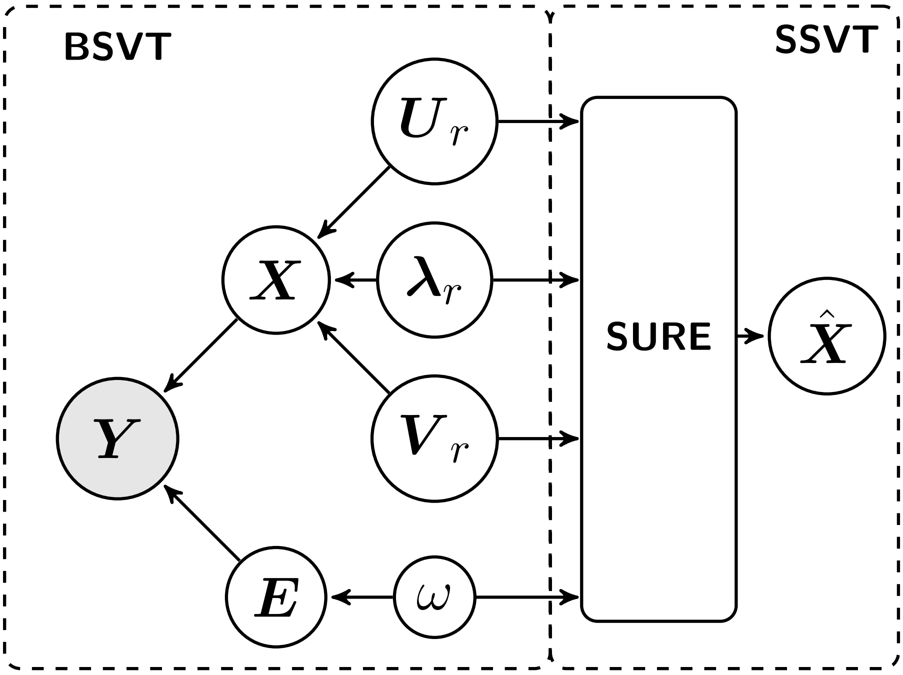


Figure 3.1 : Schematic diagram of BSSVT to denoise a patch matrix using variational Bayesian inference and SURE criterion

The resulting probabilistic graphical model is shown in the leftmost part of Fig 3.1. For notational simplicity, all unknown parameters are collectively denoted by $\mathbf{Z} = \{\boldsymbol{U}_r, \boldsymbol{\lambda}_r, \boldsymbol{V}_r, \omega\}$. Therefore, the joint distribution of the parameters and data is given by

$$p(\boldsymbol{Y}, \mathbf{Z}|r) = p(\boldsymbol{Y}|\boldsymbol{U}_r, \boldsymbol{\lambda}_r, \boldsymbol{V}_r, \omega, r)p(\boldsymbol{U}_r)p(\boldsymbol{\lambda}_r)p(\boldsymbol{V}_r)p(\omega). \tag{3.10}$$

Model Learning via variational Bayesian Inference

Full Bayesian inference using the above joint distribution is computationally intractable since the marginal distribution $p(\mathbf{Y})$ is not available analytically. In comparison with MCMC sampling methods, variational Bayesian (VB) inference [136] exhibits much lower computational complexity, so that variational Bayesian inference is utilized to infer the posterior distribution of Eq. (3.10) [137, 138, 139]. In particular, a distribution $q(\mathbf{Z})$ is constructed to approximate the true posterior distribution $p(\mathbf{Z}|\boldsymbol{Y})$ by minimizing the Kullback-Leibler (KL) divergence:

$$KL(q(\mathbf{Z}) \parallel p(\mathbf{Z}|\boldsymbol{Y})) = -\int q(\mathbf{Z}) \log \frac{p(\mathbf{Z}|\boldsymbol{Y})}{q(\mathbf{Z})} d\mathbf{Z} \geq 0. \tag{3.11}$$

The KL divergence is equal to 0 iff $p(\mathbf{Z}|\boldsymbol{Y})$ is identical to $q(\mathbf{Z})$ [140].

Based on the mean field approximation, the proposed posterior approximation can be factorized as

$$q(\mathbf{Z}) = q(\boldsymbol{U}_r|\boldsymbol{Y}, r)q(\boldsymbol{\lambda}_r|\boldsymbol{Y}, r)q(\boldsymbol{V}_r|\boldsymbol{Y}, r)q(\omega|\boldsymbol{Y}, r). \tag{3.12}$$

Applying the VB theorem to Eq. (3.10), the following approximate posterior distributions can be obtained:

$$q(\boldsymbol{U}_r|\boldsymbol{Y}, r) \sim vMF(\boldsymbol{F}_{\boldsymbol{U}}), \tag{3.13}$$

$$q(\boldsymbol{V}_r|\boldsymbol{Y}, r) \sim vMF(\boldsymbol{F}_{\boldsymbol{V}}), \tag{3.14}$$

$$q(\boldsymbol{\lambda}_r|\boldsymbol{Y}, r) \sim t\mathcal{N}(\boldsymbol{\mu}, \sigma^2\boldsymbol{I}_r; \mathcal{L}_r), \tag{3.15}$$

$$q(\omega|\boldsymbol{Y}, r) \sim \text{Gamma}(\alpha, \beta). \tag{3.16}$$

Here, $vMF(\cdot)$ denotes the von Mises-Fisher distribution [141], $t\mathcal{N}(\boldsymbol{\mu}, \sigma^2\boldsymbol{I}_r; \mathcal{L}_r)$ is the truncated normal distribution with support $\mathcal{L}_r$ and $\text{Gamma}(\alpha, \beta)$ denotes the gamma distribution with shape α and rate β. The analytical forms of the above distributions are provided in the Appendix.

The parameters of Eqs. (3.13)-(3.16) are given by

$$\boldsymbol{F_U} = \hat{\omega}\boldsymbol{Y}\widehat{\boldsymbol{V}}_r\widehat{\boldsymbol{D}}_r, \tag{3.17}$$

$$\boldsymbol{F_V} = \hat{\omega}\boldsymbol{Y}^\top\widehat{\boldsymbol{U}}_r\widehat{\boldsymbol{D}}_r, \tag{3.18}$$

$$\boldsymbol{\mu} = \text{diag}(\widehat{\boldsymbol{V}}_r^\top\boldsymbol{Y}^\top\widehat{\boldsymbol{U}}_r), \tag{3.19}$$

$$\sigma^2 = \hat{\omega}^{-1}, \tag{3.20}$$

$$\alpha = \frac{nm}{2}, \tag{3.21}$$

$$\beta = \frac{1}{2}(\widehat{\boldsymbol{\lambda_r}^\top\boldsymbol{\lambda_r}} + tr(\boldsymbol{Y}\boldsymbol{Y}^\top - 2\boldsymbol{Y}\widehat{\boldsymbol{V}}_r\widehat{\boldsymbol{D}}_r\widehat{\boldsymbol{U}}_r^\top)), \tag{3.22}$$

where $\widehat{\omega}$ denotes the expectation of ω with respect to $q(\omega)$ and similarly for the other variables.

The VB algorithm requires iteration of Eqs. (3.17)-(3.22) until convergence, which in turn requires iterative evaluation of the moments of the distributions (3.13)-(3.16):

$$\widehat{\boldsymbol{U}}_r = \boldsymbol{U}_{F_U}G(n, \boldsymbol{D}_{\boldsymbol{F_U}})\boldsymbol{V}_{\boldsymbol{F_U}}^\top, \tag{3.23}$$

$$\widehat{\boldsymbol{V}}_r = \boldsymbol{U}_{\boldsymbol{F_V}}G(m, \boldsymbol{D}_{\boldsymbol{F_V}})\boldsymbol{V}_{\boldsymbol{F_V}}^\top, \tag{3.24}$$

$$\widehat{\boldsymbol{\lambda}}_r = \boldsymbol{\mu} + \sigma\zeta(\boldsymbol{\mu}, \sigma), \tag{3.25}$$

$$\widehat{\boldsymbol{\lambda}_r^\top\boldsymbol{\lambda}_r} = r\sigma^2 + \boldsymbol{\mu}^\top\widehat{\boldsymbol{l}}_r - \sigma\rho(\boldsymbol{\mu}, \sigma), \tag{3.26}$$

$$\hat{\omega} = \frac{\alpha}{\beta}, \tag{3.27}$$

where $\boldsymbol{U}_{\boldsymbol{F_U}}$, $\boldsymbol{D}_{\boldsymbol{F_U}}$, $\boldsymbol{V}_{\boldsymbol{F_U}}$, $\boldsymbol{U}_{\boldsymbol{F_V}}$, $\boldsymbol{D}_{\boldsymbol{F_V}}$ and $\boldsymbol{V}_{\boldsymbol{F_V}}$ are the SVD parameters of $\boldsymbol{F_U}$ and $\boldsymbol{F_V}$ respectively and the definition of the functions $\zeta(\cdot,\cdot)$, $\rho(\cdot,\cdot)$ and $G(\cdot,\cdot)$ are provided in the Appendix.

If $\widehat{\boldsymbol{U}_r}$ and $\widehat{\boldsymbol{V}_r}$ are formed from scaled singular vectors of the noisy patch matrix $\boldsymbol{Y}$, so that

$$\widehat{\boldsymbol{U}_r} = \boldsymbol{U}_{;r}\boldsymbol{K}_{\boldsymbol{U}}, \tag{3.28}$$

$$\widehat{\boldsymbol{V}_r} = \boldsymbol{V}_{;r}\boldsymbol{K}_{\boldsymbol{V}}, \tag{3.29}$$

where $\boldsymbol{U}_{;r}$ and $\boldsymbol{V}_{;r}$ denote the first r columns of the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ respectively, $\boldsymbol{K}_{\boldsymbol{U}} = diag(\boldsymbol{k}_{\boldsymbol{U}}) \in \mathbb{R}^{r\times r}$ and $\boldsymbol{K}_{\boldsymbol{V}} = diag(\boldsymbol{k}_{\boldsymbol{V}}) \in \mathbb{R}^{r\times r}$ are the proportionality constants, then Eqs. (3.17)-(3.22) and (3.23)-(3.27) can be greatly simplified and each iteration using these equations satisfies (3.28) and (3.29). Detailed derivations of these equations are given in [58]. For the above inference, it is assumed that the rank r was known. One popular method to determine the rank r is to infer the posterior $p(r|\boldsymbol{Y})$ [51, 58]. This method requires trying out all possible values of the rank, i.e. from order 1 to $n-1$ for each patch group matrix, resulting in a huge computational burden in patch-based image processing. Here, the automatic relevance determination principle in Bayesian sparse learning is resorted to determine the rank r [134]. A relatively large value is initialized for r, e.g. $r = n-1$. During iterations, most of the values of $\boldsymbol{k}_{\boldsymbol{U}}$ and $\boldsymbol{k}_{\boldsymbol{V}}$ are driven to very small values, which forces the posterior means of most rows of $\boldsymbol{U}$ and $\boldsymbol{V}$ as well as most SVs to approach zero. The rank is therefore effectively reduced by removing those items from the model. The inferential framework of BSVT is outlined in Algorithm 1.

3.2.2 SURE-based singular value thresholding

As noted above, if the image is reconstructed using the low-rank approximation directly, it tends to produce a very weak noise-like pattern in flat areas and around edges, particularly in the case of moderate or high noise levels [46]. SURE is an unbiased statistical estimate of the MSE between an original unknown data source and a processed version of its noisy observation. This estimate depends only on the observed data and does not require any prior assumption on the noise-free source.

Algorithm 1 Variational Bayesian singular value thresholding

```
Require: Noisy patch matrix $\boldsymbol{Y}$ of size $n \times m$
Perform SVD on $\boldsymbol{Y}$ : $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$, $\boldsymbol{D} = diag(\boldsymbol{\lambda}_Y)$
Initialize: $\boldsymbol{k}_{\boldsymbol{U}}^{(0)} = \boldsymbol{k}_{\boldsymbol{V}}^{(0)} = \mathbf{1}_{r,1}$, which is a $r \times 1$ matrix with all entries equal to 1,
    $\widehat{\boldsymbol{\lambda}_r}^{(0)} = \boldsymbol{\lambda}_{\boldsymbol{Y};r}$, $\widehat{w}^{(0)} = mn / \sum_{i=r+1}^{n} \lambda_{\boldsymbol{Y},i}^2$, $t = 1$.
repeat evaluate the following equations:
    $\boldsymbol{k}_{\boldsymbol{U}}^{(t)} = G(n, \widehat{w}^{(t-1)}\boldsymbol{\lambda}_{\boldsymbol{Y};r} \circ \boldsymbol{k}_{\boldsymbol{V}}^{(t-1)} \circ \widehat{\boldsymbol{\lambda}_r}^{(t-1)}),$   (3.30)
    $\boldsymbol{k}_{\boldsymbol{V}}^{(t)} = G(m, \widehat{w}^{(t-1)}\boldsymbol{\lambda}_{\boldsymbol{Y};r} \circ \boldsymbol{k}_{\boldsymbol{U}}^{(t-1)} \circ \widehat{\boldsymbol{\lambda}_r}^{(t-1)}),$   (3.31)
    $\boldsymbol{\mu}^{(t)} = \boldsymbol{k}_{\boldsymbol{V}}^{(t-1)} \circ \boldsymbol{\lambda}_{\boldsymbol{Y};r} \circ \boldsymbol{k}_{\boldsymbol{U}}^{(t-1)},$   (3.32)
    $\sigma^{(t)} = (\widehat{w}^{(t-1)})^{-1/2},$   (3.33)
    $\widehat{\boldsymbol{\lambda}_r}^{(t)} = \boldsymbol{\mu}^{(t-1)} + \sigma^{(t-1)}\zeta(\boldsymbol{\mu}^{(t-1)}, \sigma^{(t-1)}),$   (3.34)
    $\widehat{\boldsymbol{\lambda}_r^\top\boldsymbol{\lambda}_r}^{(t)} = (\boldsymbol{\mu}^{(t-1)})^\top\widehat{\boldsymbol{\lambda}_r}^{(t-1)} + r(\sigma^{(t-1)})^2 - \sigma^{(t-1)}\rho(\boldsymbol{\mu}^{(t-1)}, \sigma^{(t-1)})^\top\mathbf{1}_{r,1},$   (3.35)
    $\widehat{w}^{(t)} = mn[\boldsymbol{\lambda}_{\boldsymbol{Y}}^\top\boldsymbol{\lambda}_{\boldsymbol{Y}} + \widehat{\boldsymbol{l}_r^\top\boldsymbol{l}_r}^{(t-1)} - 2(\boldsymbol{k}_{\boldsymbol{V}}^{(t-1)} \circ \widehat{\boldsymbol{\lambda}_r}^{(t-1)} \circ \boldsymbol{k}_{\boldsymbol{U}}^{(t-1)})^\top\boldsymbol{\lambda}_{\boldsymbol{Y};r}]^{-1}.$   (3.36)
    Set $t = t + 1$
until convergence is reached with reduced $r$
Set $\boldsymbol{U}_r = \boldsymbol{U}_{;r}\boldsymbol{k}_{\boldsymbol{U};r}$, $\boldsymbol{D}_r = \boldsymbol{D}_{;r}$, $\boldsymbol{V}_r = \boldsymbol{V}_{;r}\boldsymbol{k}_{\boldsymbol{V};r}$
Ensure: $\boldsymbol{U}_r$, $\boldsymbol{D}_r$, $\boldsymbol{V}_r$
```

Various studies have demonstrated that SURE is particularly powerful for tuning the regularization parameters for high-quality edge-preserving image filtering [56, 57]. In order to suppress artefacts in smooth areas and around edges, SURE is employed to refine the singular values $\boldsymbol{D}_r$ ($r < \min(m, n)$) with respect to minimizing the estimation risk or the MSE between the actual data $\boldsymbol{X}$ and the approximation $\hat{\boldsymbol{X}}$. This can be performed by selecting a parameter τ to shrink the singular values $\boldsymbol{D}_r$:

$$\mathrm{MSE}(\tau) = E \left\| \boldsymbol{X} - \mathrm{SVT}_\tau(\hat{\boldsymbol{X}}) \right\|_F^2 = E \left\| \boldsymbol{X} - \mathrm{SVT}_\tau(\boldsymbol{U}_r \boldsymbol{D}_r \boldsymbol{V}_r^\top) \right\|_F^2, \tag{3.37}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

Similar to the case for full-rank SVD, the expectation in Eq. (3.37) depends on the true $\boldsymbol{X}$ which is not available. Determination of τ based on minimizing MSE thus cannot be achieved directly. However, it is feasible to construct an unbiased estimate of the MSE, namely, Steins Unbiased Risk Estimator. Assuming $m > n > r$, the unbiased risk estimator can be derived for the rank reduced eigen-triplets:

$$\begin{aligned} &\mathrm{SURE}_s(\mathrm{SVT}_\tau(\boldsymbol{U}_r \boldsymbol{D}_r \boldsymbol{V}_r^\top)) \\ &= -mn\sigma_s^2 + \sum_{i=1}^{\min(m,r)} \min(\tau^2, \lambda_i^2) + 2\sigma_s^2 \mathrm{div}_s(\mathrm{SVT}_\tau(\boldsymbol{U}_r \boldsymbol{D}_r \boldsymbol{V}_r^\top)). \end{aligned} \tag{3.38}$$

In comparison with full-rank SURE [36, 142], please note that σ_s^2 here is the residual noise variance of the rank-reduced $\boldsymbol{X}$ and $\min(m, n)$ degrades to $\min(m, r)$. Considering the soft threshold function of $f(\lambda_i)$ in Eq. (2.1), the divergence for rank-reduced eigen-triplets is modified to:

$$\begin{aligned} &\mathrm{div}_s(\mathrm{SVT}_\tau(\boldsymbol{U}_r \boldsymbol{D}_r \boldsymbol{V}_r)) \\ &= 2 \sum_{i \neq j, i,j=1}^{\min(m,r)} \frac{\lambda_i(\lambda_i - \tau)_+}{\lambda_i^2 - \lambda_j^2} + (|m - r|) \sum_{i=1}^{\min(m,r)} (1 - \frac{\tau}{\lambda_i})_+ + \sum_{i=1}^{\min(m,r)} \mathbb{I}_{\lambda_i > \tau}, \end{aligned} \tag{3.39}$$

where $\mathbb{I}$ denotes the indicator function.

Assume that $\hat{\boldsymbol{X}} = \boldsymbol{X} + \boldsymbol{E}_s$, and σ_s can be estimated from the difference between the noisy observation $\boldsymbol{Y}$ and the estimation $\hat{\boldsymbol{X}}$

$$\Delta = \boldsymbol{Y} - \hat{\boldsymbol{X}} = (\boldsymbol{X} + \boldsymbol{E}) - (\boldsymbol{X} + \boldsymbol{E}_s) = \boldsymbol{E} - \boldsymbol{E}_s. \tag{3.40}$$

The expectation of Eq. (3.40) is denoted as

$$\langle \Delta^2 \rangle = \langle \boldsymbol{E}^2 \rangle + \langle \boldsymbol{E}_s^2 \rangle - 2\langle \boldsymbol{E} \cdot \boldsymbol{E}_s \rangle = \sigma^2 + \sigma_s^2 - 2\langle \boldsymbol{E} \cdot \boldsymbol{E}_s \rangle, \tag{3.41}$$

where $\langle \cdot \rangle$ is the expectation operator.

Since $\boldsymbol{E}_s$ can be viewed as the smoothed version of noise $\boldsymbol{E}$, it is clear that $\langle \boldsymbol{E} \cdot \boldsymbol{E}_s \rangle = \langle (\boldsymbol{E}_s + \Delta) \cdot \boldsymbol{E}_s \rangle = \langle \Delta \cdot \boldsymbol{E}_s \rangle + \langle \boldsymbol{E}_s^2 \rangle$. It is well known that the high-frequency component Δ is much smaller than $\boldsymbol{E}_s$, which results in $\langle \boldsymbol{E} \cdot \boldsymbol{E}_s \rangle \approx \langle \boldsymbol{E}_s^2 \rangle = \sigma_s^2$. Therefore, Eq. (3.41) can be written as

$$\langle \Delta^2 \rangle = \sigma^2 + \sigma_s^2 - 2\sigma_s^2 = \sigma^2 - \sigma_s^2, \tag{3.42}$$

where σ^2 is the noise variance in the observation $\boldsymbol{Y}$ which has been estimated using BSVT in the first step. Eq. (3.42) is thus equivalent to

$$\sigma_s^2 = \sigma^2 - \langle \Delta^2 \rangle = \sigma^2 - \frac{1}{mn} \left\| \boldsymbol{Y} - \hat{\boldsymbol{X}} \right\|_F^2. \tag{3.43}$$

Considering that $\boldsymbol{E}_s$ contains not only the noise residual but also the estimation error of the noiseless image, a scaling factor γ controlling the depth of filtering is required. That is

$$\sigma_s = \gamma \sqrt{(\sigma^2 - \frac{1}{mn} \left\| \boldsymbol{Y} - \hat{\boldsymbol{X}} \right\|_F^2)}. \tag{3.44}$$

It is recommended to set γ around 0.55 to 0.65 to produce satisfactory results for natural image denoising [143, 144].

The outline of the SSVT method is presented in Algorithm 2.

3.2.3 The hybrid BSSVT algorithm

For the complete image BSSVT denoising method, patches with similar spatial structure are clustered to form a patch matrix. BSVT and SSVT are then applied

Algorithm 2 SURE-based singular value thresholding

```
Require: Rank-reduced eigen-triplets $\boldsymbol{U}_r$, $\boldsymbol{D}_r$, $\boldsymbol{V}_r$, and interval $[\tau_{\min}, \tau_{\max}]$
for $\tau$ from $\tau_{\min}$ to $\tau_{\max}$ do
    Compute $\mathrm{div}_s(\mathrm{SVT}_\tau(\boldsymbol{U}_r\boldsymbol{D}_r\boldsymbol{V}_r^\top))$ using Eq. (3.39)
    Compute $\mathrm{SURE}_s(\mathrm{SVT}_\tau)(\boldsymbol{U}_r\boldsymbol{D}_r\boldsymbol{V}_r^\top)$ using Eq. (3.38)
end for
Find the $\tau_0$ with minimal $SURE_s$
Ensure: $\hat{\boldsymbol{X}} = \sum_{i=1}^{r}(\lambda_i - \tau_0)_+\boldsymbol{u}_i\boldsymbol{v}_i^\top$
```

in succession on each patch 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 denoising performance [38]. This scheme is implemented as

$$\boldsymbol{Y}^{(k+1)} = \hat{\boldsymbol{X}}^{(k)} + \delta(\boldsymbol{Y} - \hat{\boldsymbol{X}}^{(k)}), \tag{3.45}$$

where k denotes algorithm iteration and $0 < \delta < 1$ is a relaxation parameter. As reviewed in the Introduction, most existing approaches require an extra step to update the estimation of the noise variance due to the feedback of filtered noise, where the original noise variance propagates in each iteration. BSVT performs the low-rank approximation and infers the noise variance from $\boldsymbol{Y}^{(k+1)}$ itself without needing any prior knowledge of the original observation $\boldsymbol{Y}$ as well as the estimators from $\boldsymbol{Y}^{(1)}$ to $\boldsymbol{Y}^{(k)}$. The complete procedure for the image BSSVT denoising algorithm is summarized in Algorithm 3.

Algorithm 1 proceeds by iteratively estimating one variable while holding the others fixed. By the properties of the variational Bayesian method, the algorithm is guaranteed to converge to a local minimum of the variational bound [145]. Employing Algorithm 2 with a low threshold τ fails to remove noise, while a high threshold removes noise but also induces both spatial blurring and contrast loss. Due to the convex behavior of SURE/MSE, the searching scheme in Algorithm 2 can guarantee to obtain the optimal SURE threshold [146]. Therefore the BSSVT algorithm converges to a local minimum after a number of successive approximation iterations, resulting in an ideal balance offering strong noise reduction while maintaining important image features.

Algorithm 3 Image denoising by BSSVT

```
Require: Noisy image $\boldsymbol{y}$
Initialize: $\hat{\boldsymbol{x}}^{(0)} = \boldsymbol{y}$, $\boldsymbol{y}^{(0)} = \boldsymbol{y}$;
for $k = 1 : K$ do
    Iterative regularization using Eq. (3.45)
    for each patch $\boldsymbol{y}_i$ in $\boldsymbol{y}^{(k)}$ do
        Cluster similar patch to matrix $\boldsymbol{Y}_i$
        Apply Algorithm 1 on $\boldsymbol{Y}_i$ to obtain $\boldsymbol{U}_r$, $\boldsymbol{D}_r$ and $\boldsymbol{V}_r$
        Apply Algorithm 2 on $\boldsymbol{U}_r$, $\boldsymbol{D}_r$ and $\boldsymbol{V}_r$ to estimate $\hat{\boldsymbol{X}}_i$
    end for
    Aggregate $\hat{\boldsymbol{X}}_i$ to form the denoised image $\hat{\boldsymbol{x}}^{(k)}$
end for
Ensure: clean image $\hat{\boldsymbol{x}}^{(K)}$
```

3.3 Experiments

3.3.1 Parameter settings and performance evaluation

The performance of BSSVT is evaluated on twelve benchmark grayscale images, shown in Fig 3.2. 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. The

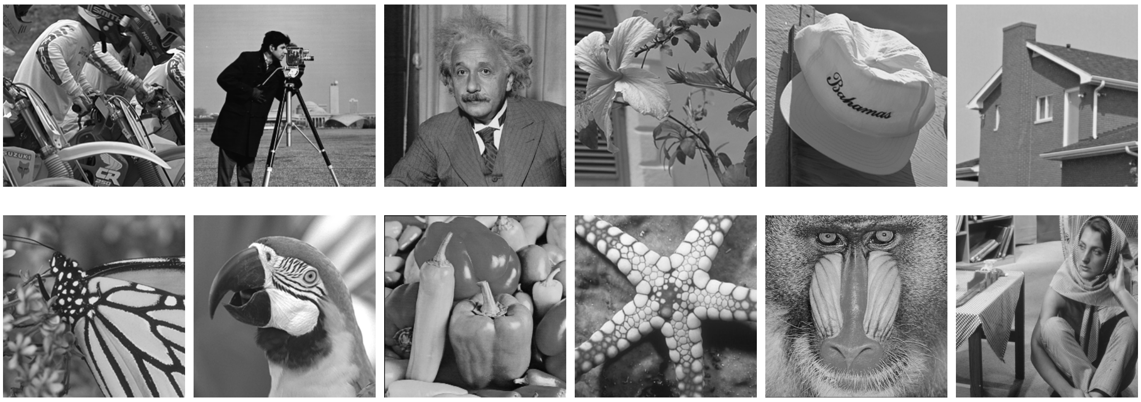


Figure 3.2 : The 12 test images used in image denoising experiments.

setting of patch size and the number of similar patches recommended in previous studies [38, 5] is adopted here: 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 $\sigma = 20$, 50, 70 and 100 respectively. Throughout this chapter, the scaling factor γ is fixed as 0.55.

The performance of BSSVT in terms of PSNR and SSIM [67] are evaluated. Given a ground truth grayscale image $\boldsymbol{X}$, the PSNR of the recovered image $\hat{\boldsymbol{X}}$ is estimated by:

$$PSNR(\boldsymbol{X}, \hat{\boldsymbol{X}}) = 10 \cdot \log_{10}(\frac{255^2}{\left\|\boldsymbol{X} - \hat{\boldsymbol{X}}\right\|_2^2}). \tag{3.46}$$

Assuming an image patch $\boldsymbol{G}$ from $\boldsymbol{X}$ as well as the patch $\boldsymbol{H}$ from the corresponding recovery $\hat{\boldsymbol{X}}$, the SSIM index between $\boldsymbol{G}$ and $\boldsymbol{H}$ is defined by:

$$SSIM(\boldsymbol{G}, \boldsymbol{H}) = \frac{2(\mu_{\boldsymbol{G}}\mu_{\boldsymbol{H}} + C_1)(2\nu_{\boldsymbol{GH}} + C_2)}{(\mu_{\boldsymbol{G}}^2 + \mu_{\boldsymbol{H}}^2 + C_1)(\nu_{\boldsymbol{G}}^2 + \nu_{\boldsymbol{H}}^2 + C_2)}, \tag{3.47}$$

where $\mu_{\boldsymbol{G}}$ and $\nu_{\boldsymbol{H}}$ are the average intensity and standard deviation of $\boldsymbol{G}$ and $\boldsymbol{H}$, respectively. $\nu_{\boldsymbol{GH}}$ denotes the cross correlation between $\boldsymbol{G}$ and $\boldsymbol{H}$, and the small constants C_1 and C_2 are used to avoid numerical instability. The SSIM of the entire image is estimated by averaging the local SSIM indices using a sliding window [147].

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.

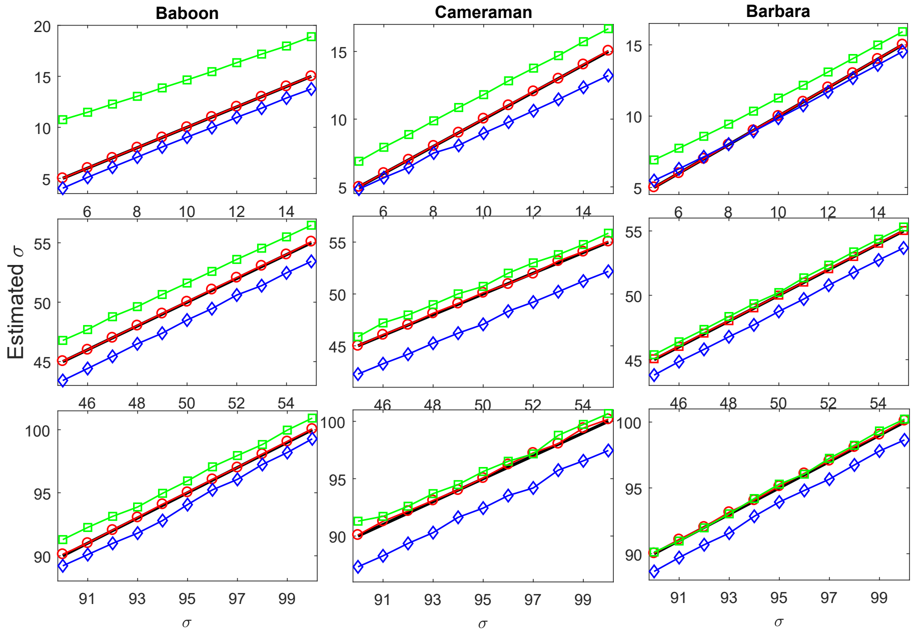


Figure 3.3 : Columns from left to right depict the comparison of the noise estimation results for the Baboon, Cameraman and Barbara images, respectively. Rows from top to bottom describe the comparison of noise estimation results for low ($5 \leq \sigma \leq 15$), moderate ($45 \leq \sigma \leq 55$) and severe ($90 \leq \sigma \leq 100$) levels of noise, respectively. The results of BSVT, MAD and SVK are represented by the circles, squares and diamonds, respectively. The truth is illustrated by the solid black line.

3.3.2 Effect on noise variance estimation

The effectiveness of the proposed variational Bayesian model to estimate the noise variance in the BSVT step is demonstrated below. Three patch group matrices are chosen, i.e. one with structure from Baboon, one with texture from Cameraman, and one with both structure and texture from Barbara. Fig 3.3 shows the average noise variance of 20 noisy samples for each of these three representative patch group matrices. Two other popular methods based on the wavelet MAD and the scale variance of kurtosis (SVK) are also plotted for comparison [65, 70].

It is apparent that the variational Bayesian model accurately tracks the actual noise variance in the cases of low, moderate and severe noise contamination for each image. The difference between the true noise variance (in black) and that estimated by BSVT (in red) is almost unrecognizable in most cases. MAD has been broadly applied to assess different kinds of image denoising algorithms. However, it can be rather problematic when MAD is applied to images containing a considerable component at the HH_1 level in the wavelet domain [148]. Therefore, using the noisy version of these coefficients at this level to estimate the noise variance can result in considerable errors. The error according to MAD in the simulation is the largest across the three images and three noise level intervals. This result is consistent with the findings in [66]. Similar to MAD, the performance of SVK varies significantly across the images and noise levels. Although it is better than MAD, it is much worse than the Bayesian model, particularly for the Cameraman image. Recall that the major purpose of the first step in BSSVT is to remove noise through the Bayesian low-rank approximation. The precise noise variance obtained in this step is a by-product of this procedure, although it is required in the second step of BSSVT as well as in other denoising methods.

3.3.3 Effect of SURE on eigen-triplets thresholding

The second step of BSSVT employs SURE to optimally tune the rank-reduced eigen-triplets in terms of minimizing the estimation risk or MSE. Here the effectiveness of SURE for rank-reduced eigen-triplets thresholding is evaluated. Fig 3.4 shows the SURE and MSE for the rank-reduced SVs as a function of the threshold τ for the representative patch group matrices used in Fig 3.3 with $\sigma = 20$ for Baboon, 50 for Cameraman, and 100 for Barbara, respectively. For each case, with increasing

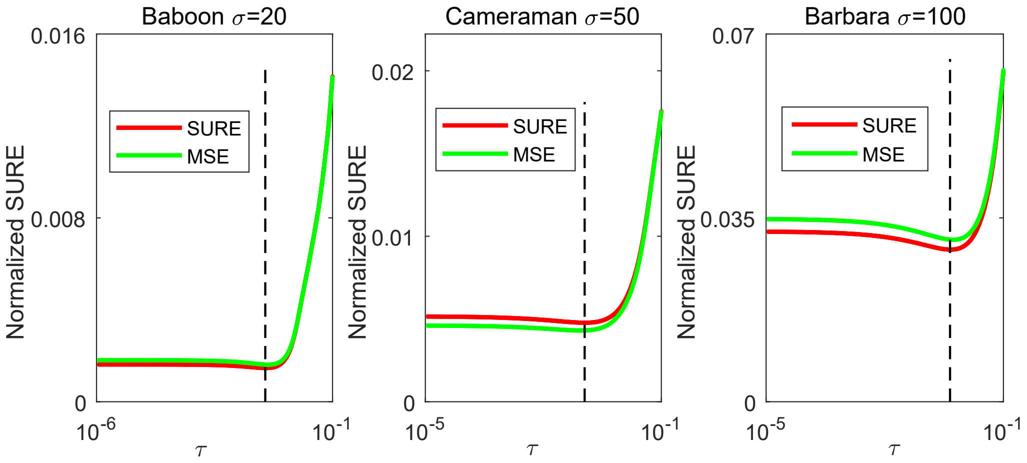


Figure 3.4 : SURE and MSE as a function of threshold value for Baboon ($\sigma = 20$), Cameraman ($\sigma = 50$) and Barbara ($\sigma = 100$). Columns from left to right correspond to noise level $\sigma = 20,\ 50$ and 100.

threshold τ, the estimated risk and MSE first have a relatively high plateau, and then descend to reach the minimum. They increase dramatically from this point with increasing τ. Due to the error of the estimated variance, there is a minor offset between SURE and the actual MSE. However, it was found that the sensitivity of SURE to the estimated values of variance is small, and the locations of the minima of the MSE and SURE are almost the same. These findings are consistent with the results in [146, 149] of SURE for the full rank matrix with estimated variance. These plots thus indicate that SURE can converge to a minimum and approximate the true patch with minimal estimation risk.

3.3.4 Numerical Results

There have been a large number of nonlocal algorithms developed in the past decade. BM3D [5] is the benchmark algorithm in image nonlocal denoising. WNNM [38] is always ranked as one of the most competitive methods in comparative studies while RNNM [45] shares similar principles to BSSVT in aiming to balance between

Table 3.1 : Denoising results (PSNR) by competing methods on the 12 test images. The best results are in bold.

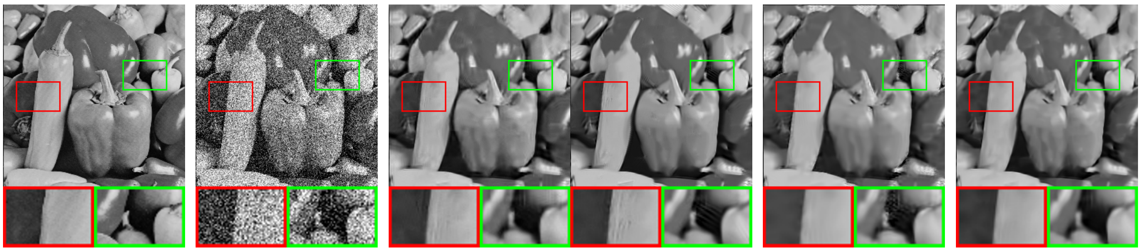
σ	20				50			
schemes	BM3D	WNNM	RNNM	BSSVT	BM3D	WNNM	RNNM	BSSVT
Bike	28.24	28.70	27.99	**28.74**	22.42	22.50	22.47	**22.83**
Cameraman	30.36	**30.68**	30.08	30.43	24.99	25.16	24.93	**25.49**
Einstein	31.29	31.47	30.97	**31.48**	27.11	27.19	26.65	**27.35**
Flower	29.99	30.42	29.73	**30.37**	25.12	25.33	24.87	**25.65**
Hat	31.55	**32.05**	31.35	31.86	27.14	27.23	26.55	**27.59**
House	33.88	**34.14**	33.64	34.12	29.39	**29.87**	28.65	29.41
Monarch	30.52	31.34	29.25	**31.42**	25.46	25.56	25.37	**25.97**
Parrot	29.88	**30.03**	29.68	29.66	24.76	24.69	24.72	**25.08**
Peppers	31.28	31.59	31.07	**31.62**	26.16	26.23	25.87	**26.40**
Starfish	29.45	30.20	29.59	**30.23**	24.29	24.41	24.32	**24.67**
Baboon	25.58	**25.67**	25.49	25.59	21.83	22.15	22.13	**22.51**
Barbara	31.23	**31.68**	31.35	31.58	26.24	26.72	26.61	**26.73**
Average	30.27	**30.66**	30.02	30.59	25.41	25.59	25.26	**25.81**
σ	70				100			
schemes	BM3D	WNNM	RNNM	BSSVT	BM3D	WNNM	RNNM	BSSVT
Bike	20.46	20.08	20.36	**20.87**	18.38	17.83	18.25	**18.63**
Cameraman	22.56	22.72	22.59	**23.30**	19.86	20.25	20.08	**20.90**
Einstein	25.23	24.97	24.39	**25.56**	22.63	21.79	21.99	**22.68**
Flower	23.20	23.47	22.99	**23.77**	20.59	21.60	21.15	**22.32**
Hat	25.46	25.23	24.73	**25.80**	22.90	22.59	22.34	**23.24**
House	26.98	27.15	26.63	**27.58**	23.71	23.27	22.88	**24.15**
Monarch	22.99	23.40	23.14	**23.90**	19.85	20.82	20.35	**21.31**
Parrot	22.15	22.39	22.29	**22.87**	19.17	19.70	19.61	**20.45**
Peppers	23.97	23.63	23.55	**24.21**	21.52	20.82	21.12	**21.63**
Starfish	22.35	21.83	22.19	**22.53**	20.00	19.05	20.01	**20.41**
Baboon	20.58	20.87	20.32	**21.09**	19.17	19.39	19.46	**20.22**
Barbara	24.56	24.89	24.74	**25.25**	23.34	23.18	23.06	**23.98**
Average	23.37	23.39	23.16	**23.89**	20.93	20.87	20.86	**21.66**

Table 3.2 : Denoising results (SSIM) by competing methods on the 12 test images. The Best results are in bold.

σ	20				50			
schemes	BM3D	WNNM	RNNM	BSSVT	BM3D	WNNM	RNNM	BSSVT
Bike	0.887	0.893	0.895	**0.896**	0.688	0.687	0.705	**0.711**
Cameraman	0.872	**0.877**	0.854	0.875	0.747	0.755	0.685	**0.760**
Einstein	0.801	**0.807**	0.806	0.802	0.696	0.699	0.648	**0.701**
Flower	0.874	**0.885**	0.885	0.880	0.716	0.724	0.687	**0.732**
Hat	0.876	0.883	0.856	**0.884**	0.767	0.776	0.666	**0.780**
House	**0.869**	0.871	0.863	0.864	0.812	0.826	0.734	**0.828**
Monarch	0.923	**0.930**	0.912	0.922	0.824	0.829	0.792	**0.831**
Parrot	0.867	0.868	0.857	**0.871**	0.757	0.750	0.697	**0.758**
Peppers	0.890	**0.894**	0.878	0.891	0.786	0.788	0.735	**0.790**
Starfish	0.870	0.885	0.872	**0.887**	0.725	0.720	0.713	**0.737**
Baboon	0.722	**0.730**	0.728	0.726	0.469	0.508	0.485	**0.513**
Barbara	0.909	**0.915**	0.910	0.912	0.762	0.785	0.784	**0.785**
Average	0.863	**0.870**	0.860	0.868	0.729	0.737	0.694	**0.744**
σ	70				100			
schemes	BM3D	WNNM	RNNM	BSSVT	BM3D	WNNM	RNNM	BSSVT
Bike	0.588	0.553	0.598	**0.613**	0.468	0.399	0.475	**0.495**
Cameraman	0.677	0.679	0.583	**0.695**	0.592	0.617	0.488	**0.620**
Einstein	0.646	0.637	0.563	**0.653**	0.592	0.569	0.468	**0.591**
Flower	0.623	0.640	0.584	**0.647**	0.505	0.552	0.465	**0.558**
Hat	0.732	0.738	0.589	**0.743**	0.689	0.683	0.489	**0.689**
House	0.778	**0.795**	0.648	0.791	**0.729**	0.726	0.649	0.726
Monarch	0.758	0.766	0.699	**0.770**	0.649	0.684	0.589	**0.695**
Parrot	0.685	0.693	0.617	**0.702**	0.599	0.624	0.524	**0.632**
Peppers	**0.739**	0.730	0.652	0.735	0.673	0.657	0.562	**0.681**
Starfish	0.652	0.623	0.619	**0.669**	0.556	0.484	0.513	**0.565**
Baboon	0.440	0.467	0.460	**0.470**	0.406	0.448	0.428	**0.452**
Barbara	0.685	0.701	0.694	**0.708**	0.658	0.683	0.669	**0.685**
Average	0.667	0.669	0.609	**0.683**	0.593	0.584	0.527	**0.616**

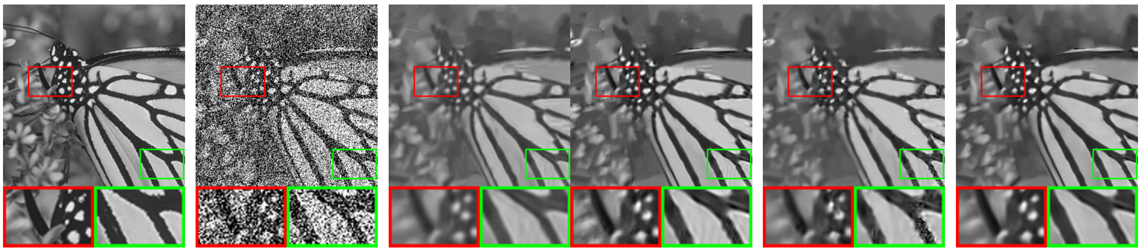
soft and hard thresholding. The performance of BSSVT with BM3D, WNNM and RNNM is compared. BSSVT and RNNM are implemented in MATLAB, while BM3D and WNNM are tested using the executables and source codes provided by the authors. Because the exact noise variance is not available in real applications, the algorithms of BM3D, WNNM and RNNM are fed with the noise variance estimated using SVK [70]. This is fair and reasonable and represents their implementation in practice. The PSNR and SSIM are estimated over 20 realizations for each scheme with $\sigma = 20,\ 50,\ 70$ and 100 dB. The PSNR and SSIM values are displayed in Tables 3.1 and 3.2 respectively, where the best results are bolded. It is apparent that for low noise levels, the performance of BSSVT is, in general, equivalent to WNNM. This is reasonable because less iterations are required to estimate the noise variance. With the increase of the noise level, BSSVT algorithm performs increasingly better than the other algorithms. In particular, compared with WNNM, the improvement in the PSNR values is greater than 0.6 dB for all images at $\sigma = 100$. As for RNNM, BSSVT outperforms it in almost every case. This may be due to its sensitivity to the error of the noise variance and the fact that the low-rank parameter r was set empirically. In addition, BSSVT outperforms BM3D in all cases in terms of PSNR. The SSIM result of BSSVT is also highly competitive against the other methods.

Figs. 3.5 and 3.6 show the comparison between the visual quality of the denoising results on the four methods. Fig 3.5 illustrates the comparison the Peppers picture under a noise level of $\sigma = 50$. BSSVT restores the edges with fewer artefacts. However, BM3D and RNNM suffer from artefacts in smooth areas and around edges. In Fig 3.6, the performance of all these algorithms on the Monarch image are compared under a noise level of $\sigma = 100$, where BSSVT achieves a visually satisfactory result with the least artefacts. In such extreme noise contamination, it is evident that the other three methods are less able to preserve the edge structures and smooth features of the image. Overall, both quantitative assessment and visual



(a) Original (b) Noisy (c) BM3D (d) WNNM (e) RNNM (f) BSSVT

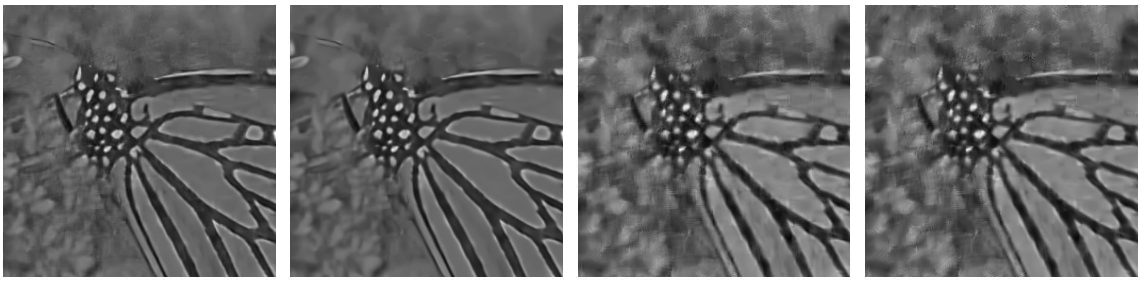
Figure 3.5 : Comparison of denoising results on the Peppers image contaminated by Gaussian white noise with $\sigma = 50$. (a) Original image, (b) noisy image (PSNR=14.12 dB), (c) BM3D (PSNR=26.16 dB), (d) WNNM (PSNR= 26.23 dB), (e) RMMM (PSNR= 25.87 dB), and (f) BSSVT (PSNR= **26.40** dB)



(a) Original (b) Noisy (c) BM3D (d) WNNM (e) RNNM (f) BSSVT

Figure 3.6 : Comparison of denoising results on the Monarch image contaminated by the Gaussian white noise with $\sigma = 100$. (a) Original image, (b) noisy image (PSNR= 8.10 dB), (c) BM3D (PSNR=19.85 dB), (d) WNNM (PSNR= 20.82 dB), (e) RMMM (PSNR= 20.35 dB), and (f) BSSVT (PSNR= **21.31** dB)

inspection demonstrate that BSSVT yields better performance in comparison to the state-of-the-art methods.



(a) BSVT+BM3D (b) BSVT+WNNM (c) BSVT+BM3D (d) BSVT+WNNM

Figure 3.7 : The effect of BSVT-BM3D and BSVT-WNNM to denoise image Monarch contaminated by the Gaussian white noise with $\sigma = 50$ (a, b) and $\sigma = 100$ (c, d). (a) BSVT-BM3D (PSNR=20.90 dB), (b) BSVT-WNNM (PSNR=20.97 dB), (c) BSVT-BM3D (PSNR=17.81 dB), (d) BSVT-WNNM (PSNR=17.85 dB).

The combination of BSVT with SSVT leads to superior performance compared with the state-of-the-art methods. A natural question to ask is whether such a combination can extend to BSVT together with other methods to take advantage of the estimated noise variance. The performance of BSVT-BM3D and BSVT-WNNM are further tested. It was found that both BSVT-BM3D and BSVT-WNNM generate over-smoothed images with performance scores lower than BSSVT. Fig 3.7 shows a typical example of denoised Monarch images using BSVT followed by BM3D and WNNM for the image contaminated by noise with $\sigma = 50$ and $\sigma = 100$, respectively. Many previous studies have indicated that BM3D and WNNM, as well as some other low-rank approximation-based methods, tend to over-smooth images [37, 46, 52]. The consecutive use of BSVT followed by BM3D or WNNM can remove the noise artefacts. However, this also smears out details, which results in over-smoothed images with relatively low PSNR. In the second step of the proposed method, SSVT, complementary to BSVT, directly maximizes the PSNR by refining the optimal threshold that minimizes the MSE estimation of rank-reduced eigen-triplets, avoiding over-smoothing the image.

In terms of computational efficiency, a desktop with a recent 2.2 GHz CPU is employed to execute the code in Matlab 2017b (Mathworks, Massachusetts, US). BSSVT requires around 20 minutes to denoise an image for varying noise levels, while the times for BM3D, WNNM, and RNNM vary from one minute to around ten minutes. Although BSSVT is relatively slow in its current form, the computational efficiency can be significantly improved via parallel computing techniques and optimization of the search interval.

3.4 Discussion and conclusion

In this chapter, a hybrid nonlocal variational Bayesian image denoising framework is proposed. The proposed BSSVT approach is closely related to nuclear norm minimization. It can be interpreted as performing a weighted nuclear norm factorization or low-rank approximation using variational Bayesian inference. The noise variance is a crucial factor that impacts on the denoising quality. Most existing nonlocal image denoising methods either resort to an extra step to pre-determine the noise variance or simply assume the true value is known. However, the error in the noise variance accumulates in any iterative regularization scheme which can further worsen the denoising quality. In contrast to these existing methods, BSSVT simultaneously removes noise and infers the latent parameters including the noise variance and the rank. It adaptively adjusts the noise variance without incurring error propagation between iterations. This is the primary reason that the proposed method outperforms these competitive algorithms in this chapter. BSSVT further refines the rank-reduced SVs based on the SURE criterion in the second step to improve edge-preservation and artefact removal. SURE has an explicit mathematical mechanism to approximate the true image by minimizing the risk or MSE, therefore maximizing PSNR. This provides another indispensable element of BSSVT to enhance the denoised image quality. Since BSVT, the first step of BSSVT, can accurately approximate the noise variance, it can also be separately applied to improve other image denoising of segmentation methods.

In this work, only Gaussian noise was considered in the model. Both Bayesian inference and the SURE criterion are able to handle non-Gaussian noise [113, 150, 151, 149].

Chapter 4

Kernelized Sparse Bayesian Matrix Factorization

4.1 Introduction

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 chapter, a hierarchical kernelized sparse Bayesian matrix factorization (KSBMF) model is developed 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 sparsity through an enforced constraint on latent factor matrices. This chapter 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.

The contributions are at two levels. From the perspective of machine learning, a generative model is presented for kernelized sparse Bayesian matrix factorization

(KSBMF). In most real-world applications, the actual rank r needed for modelling the data is initially unknown. If r used is lower than the underlying rank of the data, the model cannot model the data sufficiently well. Conversely, if r is too large, overfitting occurs. Many previous studies have indicated both cases result in inferior quality of the recovered data [152, 116]. To determine the appropriate rank, a common approach is to try different values of r by performing multiple runs and then choose the one that yields the best performance. In comparison with existing kernelized matrix factorization methods particularly the two VB realizations [83, 85], the proposed formulation implicitly estimates the rank of the matrix without requiring the prior knowledge on the rank of the matrix, which frees the user from extensive parameter-tuning and groundless attempts. In addition, KSBMF simultaneously achieves low-rankness through sparse Bayesian learning and sparsity through an 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. Moreover, the proposed model adopts different graphical model and priors as in [83]. Another significant difference between KSBMF and [85] is that the variance of a number of latent variables in [85] 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 the proposed 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 [153]. However, the side information of similarity between patches has never been taken into account in the image restoration model. A kernel function based on the similarity between each pair of patches is devised. Two algorithms are further presented which incorporate the patch similarity-based kernel into the generic KSBMF model for enhanced image denoising and inpainting.

The rest of this chapter is organized as follows. Section 4.2 elaborates on the model specification and inference of kernelized sparse Bayesian matrix factorization. Section 4.4 presents 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 4.4. Finally, Section 4.5 concludes this chapter.

4.2 KSBMF model and inference

4.2.1 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 $\mathbf{Y} = \mathbf{X} + \mathbf{E}$. A common sparsity profile is enforced on the underlying factors and thus cast it to the problem of sparse representation of factor matrices $\mathbf{U}$ and $\mathbf{V}$, that is:

$$\mathbf{Y} = \mathbf{X} + \mathbf{E} = \mathbf{U}\mathbf{V}^\top + \mathbf{E}, \tag{4.1}$$

where $\mathbf{Y} \in \mathbb{R}^{M\times N}$, $\mathbf{U} \in \mathbb{R}^{M\times r}$, $\mathbf{V} \in \mathbb{R}^{N\times r}$, $\mathbf{E} \in \mathbb{R}^{M\times N}$, and $r \ll \min(M, N)$ for sparsity.

Fig 4.1 shows the graphical model of the proposed hierarchical kernelized sparse Bayesian matrix factorization with latent variables and their corresponding priors. In order to impose sparsity into the low rank approximation model, Gaussian priors are assigned to the columns of $\mathbf{U}$ and $\mathbf{V}$ with precisions (inverse variances) γ_j, namely,

$$p(\mathbf{U}|\boldsymbol{\gamma}) = \prod_{j=1}^{r} \mathcal{N}(\mathbf{u}_{\cdot j}|\mathbf{0}, \gamma_j^{-1}\mathbf{I}_M), \tag{4.2}$$

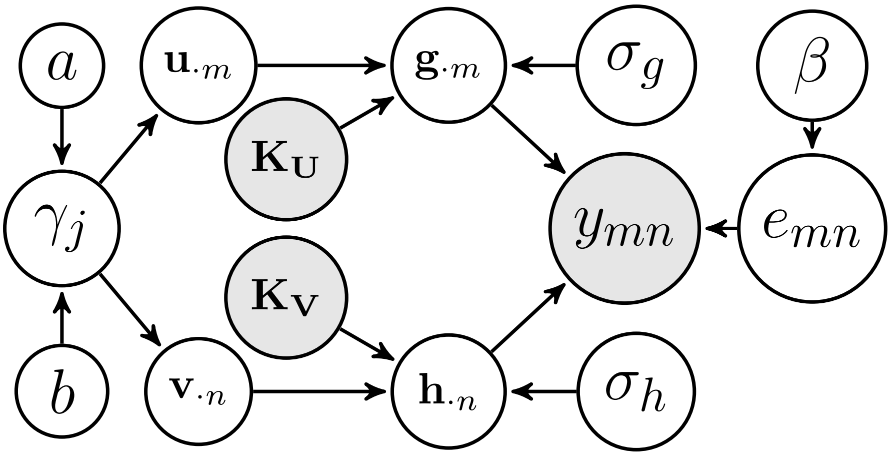


Figure 4.1 : Directed graphical representation of KSBMF model.

$$p(\mathbf{V}|\boldsymbol{\gamma}) = \prod_{j=1}^{r} \mathcal{N}(\mathbf{v}_{\cdot j}|\mathbf{0}, \gamma_j^{-1}\mathbf{I}_N), \tag{4.3}$$

where $\mathbf{I}_J \in \mathbb{R}^{J\times J}$ denotes an identity matrix. Therefore, the columns of $\mathbf{U}$ and $\mathbf{V}$ possess the same 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 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 [80, 134, 154].

To achieve the joint sparsity of $\mathbf{U}$ and $\mathbf{V}$, the conjugate Gamma hyper-prior is assigned to the precision γ_j:

$$p(\gamma_j) = \text{Gamma}(a, \frac{1}{b}) \propto \gamma_j^{a-1} exp(-b\gamma_j), \tag{4.4}$$

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

$$p(\mathbf{G}|\mathbf{U}, \mathbf{K_U}, \sigma_g) = \prod_{j=1}^{r} \mathcal{N}(\mathbf{g}_{\cdot j}|\mathbf{K_U}^\top \cdot \mathbf{u}_{\cdot j}, \sigma_g^{-1}\mathbf{I}_M). \tag{4.5}$$

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

$$p(\mathbf{H}|\mathbf{V},\mathbf{K_V},\sigma_h)=\prod_{j=1}^{r}\mathcal{N}(\mathbf{h}_{\cdot j}|\mathbf{K_V}^{\top}\cdot\mathbf{v}_{\cdot j},\sigma_h^{-1}\mathbf{I}_N). \tag{4.6}$$

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

$$p(\sigma_g)=\sigma_g^{-1}, \tag{4.7}$$

$$p(\sigma_h)=\sigma_h^{-1}. \tag{4.8}$$

In Eq. (4.1), the noise $\mathbf{E}$ is assumed obeys a Gaussian distribution with zero mean and unknown precision β. Hence, $\mathbf{E}$ is modeled as:

$$p(\mathbf{E}|\beta)=\prod_{i=1}^{M}\prod_{j=1}^{N}\mathcal{N}(e_{mn}|0,\beta^{-1}), \tag{4.9}$$

$$p(\beta)=\beta^{-1}, \tag{4.10}$$

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

$$p(\mathbf{Y}|\mathbf{G},\mathbf{H},\beta)=\mathcal{N}(\mathbf{Y}|\mathbf{G}\mathbf{H}^{\top},\beta^{-1}\mathbf{I}_{MN}). \tag{4.11}$$

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

$$\begin{aligned}p(\mathbf{Y},\mathbf{U},\mathbf{V},\mathbf{G},\mathbf{H},\sigma_g,\sigma_h,\boldsymbol{\gamma},\beta)&=p(\mathbf{Y}|\mathbf{G},\mathbf{H},\beta)p(\mathbf{G}|\mathbf{U},\mathbf{K_U},\sigma_g)p(\mathbf{H}|\mathbf{V},\mathbf{K_V},\sigma_h)\\&\quad\cdot p(\mathbf{U}|\boldsymbol{\gamma})p(\mathbf{V}|\boldsymbol{\gamma})p(\sigma_g)p(\sigma_h)p(\boldsymbol{\gamma})p(\beta).\end{aligned} \tag{4.12}$$

4.2.2 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. Variational Bayesian inference [136] is utilized to deal with this problem. Suppose $\mathbf{Z}$

represent the vector 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 Eq.3.11.

This can be refered 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

$$q(\mathbf{Z}) = \prod_k q(\mathbf{Z}_k). \tag{4.13}$$

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

$$ln\ q(\mathbf{Z}_k) = \langle ln\ p(\mathbf{Y}, \mathbf{Z}) \rangle_{\mathbf{Z} \setminus \mathbf{Z}_k} + const, \tag{4.14}$$

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. The iteration rules for all unknown variables in Eq. (4.13) is detailed below.

***Estimation of latent factors* U *and* V**

Combining the respective priors of $\mathbf{U}$ and $\mathbf{G}$ in Eqs. (4.2) and (4.5), the posterior approximation $ln\ q(\mathbf{U})$ is derived from Eq. (4.14) as:

$$\begin{aligned} ln\ q(\mathbf{U}) &= \langle ln\ P(\mathbf{Y}, \mathbf{Z}) \rangle_{\mathbf{Z} \setminus \mathbf{U}} + const \\ &= \sum_j -\frac{1}{2}(\mathbf{u}_{\cdot j}^\top (\langle \sigma_g \rangle \mathbf{K_U} \mathbf{K_U}^\top + \boldsymbol{\Gamma}_{\mathbf{u}_{\cdot j}}) \mathbf{u}_{\cdot j} - 2\langle \sigma_g \rangle \mathbf{u}_{\cdot j}^\top \mathbf{K_U} \langle \mathbf{g}_{\cdot j} \rangle) + const, \end{aligned} \tag{4.15}$$

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

From Eq. (4.15) it is found that the posterior density of the jth column $\mathbf{u}_{\cdot j}$ of

$\mathbf{U}$ obeys the multivariate Gaussian distribution:

$$q(\mathbf{u}_{\cdot j}) = \mathcal{N}(\mathbf{u}_{\cdot j} | \langle \mathbf{u}_{\cdot j} \rangle, \Sigma^{\mathbf{u}_{\cdot j}}), \tag{4.16}$$

with mean and covariance

$$\Sigma^{\mathbf{u}_{\cdot j}} = (\langle \sigma_g \rangle \cdot \mathbf{K_U K_U}^\top + \mathbf{\Gamma}_{\mathbf{u}_{\cdot j}})^{-1}, \tag{4.17}$$

$$\langle \mathbf{u}_{\cdot j} \rangle = \langle \sigma_g \rangle \cdot \Sigma^{\mathbf{u}_{\cdot j}} \mathbf{K_U} \langle \mathbf{g}_{\cdot j} \rangle. \tag{4.18}$$

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

$$q(\mathbf{v}_{\cdot j}) = \mathcal{N}(\mathbf{v}_{\cdot j} | \langle \mathbf{v}_{\cdot j} \rangle, \Sigma^{\mathbf{v}_{\cdot j}}), \tag{4.19}$$

and the mean and covariance are given by

$$\Sigma^{\mathbf{v}_{\cdot j}} = (\langle \sigma_h \rangle \cdot \mathbf{K_V K_V}^\top + \mathbf{\Gamma}_{\mathbf{v}_{\cdot j}})^{-1}, \tag{4.20}$$

$$\langle \mathbf{v}_{\cdot j} \rangle = \langle \sigma_h \rangle \cdot \Sigma^{\mathbf{v}_{\cdot j}} \mathbf{K_V} \langle \mathbf{h}_{\cdot j} \rangle, \tag{4.21}$$

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

Estimation of $\boldsymbol{\gamma}$

Applying the priors of $\mathbf{U}, \mathbf{V}$ and $\boldsymbol{\gamma}$ in the same manner to Eq. (4.14), the posterior approximation of $ln\ q(\boldsymbol{\gamma})$ is given by

$$\begin{aligned} ln\ q(\boldsymbol{\gamma}) &= \langle P(\mathbf{Y}, \mathbf{Z}) \rangle_{\mathbf{Z} \setminus \boldsymbol{\gamma}} + const \\ &= ln(\gamma_j^{a-1+\frac{m+n}{2}} exp(-\frac{1}{2}\gamma_j(\langle \mathbf{u}_{\cdot j}^T \mathbf{u}_{\cdot j} \rangle + \langle \mathbf{v}_{\cdot j}^\top \mathbf{v}_{\cdot j} \rangle + 2b))) + const. \end{aligned} \tag{4.22}$$

This is equivalent to

$$q(\gamma_j) \propto \gamma_j^{a-1+\frac{M+N}{2}} exp(-\frac{1}{2}\gamma_j(\langle \mathbf{u}_{\cdot j}^\top \mathbf{u}_{\cdot j} \rangle + \langle \mathbf{v}_{\cdot j}^\top \mathbf{v}_{\cdot j} \rangle + 2b)). \tag{4.23}$$

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

$$\langle \gamma_j \rangle = \frac{2a + M + N}{2b + \langle \mathbf{u}_{\cdot j}^\top \mathbf{u}_{\cdot j} \rangle + \langle \mathbf{v}_{\cdot j}^\top \mathbf{v}_{\cdot j} \rangle}. \tag{4.24}$$

The required expectations here are found as

$$\langle \mathbf{u}_{\cdot j}^{\top}\mathbf{u}_{\cdot j}\rangle = \langle \mathbf{u}_{\cdot j}\rangle^{\top}\langle \mathbf{u}_{\cdot j}\rangle + \mathrm{tr}(\Sigma^{\mathbf{u}_{\cdot j}}), \tag{4.25}$$

$$\langle \mathbf{v}_{\cdot j}^{\top}\mathbf{v}_{\cdot j}\rangle = \langle \mathbf{v}_{\cdot j}\rangle^{\top}\langle \mathbf{v}_{\cdot j}\rangle + \mathrm{tr}(\Sigma^{\mathbf{v}_{\cdot j}}). \tag{4.26}$$

***Estimation of* G *and* H**

Similar to estimation of **U** and **V**, the posterior approximation of **G** is given by

$$\begin{aligned}
&ln\ q(\mathbf{G}) = \langle ln\ P(\mathbf{Y},\mathbf{Z})\rangle_{\mathbf{Z}\backslash\mathbf{G}} + const \\
&= \langle ln\ P(Y|G,H,\beta)\cdot P(G|V,K_U,\sigma_g)\rangle_{U,V,H,\sigma_g,} + const \\
&= \sum_i\sum_j[-\frac{1}{2}\langle\beta\rangle(y_{ij}-g_{i\cdot}h_{\cdot j})^2 - \frac{1}{2}\langle\sigma_g\rangle(g_{ij}-K_{U_{i\cdot}}U_{\cdot j})^2] + const \\
&= \sum_i\sum_j[-\frac{1}{2}\langle\beta\rangle(y_{ij}^2-2y_{ij}g_{i\cdot}h_{\cdot j}) + (g_{i\cdot}h_{\cdot j})^2 - \frac{1}{2}\langle\sigma_g\rangle(g_{ij}^2-2g_{ij}K_{U_{i\cdot}}U_{\cdot j}+(K_{U_{i\cdot}}U_{\cdot j})^2)] \\
&+ const \\
&= \sum_i\sum_j[-\frac{1}{2}\langle\beta\rangle(g_{i\cdot}h_{\cdot j})^2 - 2y_{ij}g_{i\cdot}h_{\cdot j}) - \frac{1}{2}\langle\sigma_g\rangle(g_{ij}^2-2g_{ij}K_{U_{i\cdot}}U_{\cdot j})] + const \\
&= \sum_i[-\frac{1}{2}\langle\beta\rangle(g_{i\cdot}\langle H^TH\rangle g_{i\cdot}^T - 2g_{i\cdot}\langle H^T\rangle y_{i\cdot}^T) - \frac{1}{2}\langle\sigma_g\rangle(g_{i\cdot}I_r g_{i\cdot}^T - 2g_{i\cdot}\langle U\rangle^T K_{U_{i:}}^T)] + const \\
&= \sum_i[-\frac{1}{2}(\mathbf{g}_{i\cdot}(\langle\beta\rangle\langle\mathbf{H}^{\top}\mathbf{H}\rangle + \langle\sigma_g\rangle\mathbf{I}_r)\mathbf{g}_{i\cdot}^T - 2\mathbf{g}_{i\cdot}(\langle\mathbf{H}\rangle^{\top}\mathbf{y}_{i\cdot}^{\top} + \langle\sigma_g\rangle\langle\mathbf{U}\rangle^{\top}\mathbf{K}_{\mathbf{U}_{\cdot i}}))] + const,
\end{aligned} \tag{4.27}$$

which indicates that the ith row of **G** obeys the multivariate Gaussian distribution

$$q(\mathbf{g}_{i\cdot}) = \mathcal{N}(\mathbf{g}_{i\cdot}|\langle\mathbf{g}_{i\cdot}\rangle, \Sigma^{\mathbf{G}}). \tag{4.28}$$

The corresponding covariance and mean are denoted as

$$\Sigma^{\mathbf{G}} = (\langle\beta\rangle\langle\mathbf{H}^{\top}\mathbf{H}\rangle + \langle\sigma_g\rangle\mathbf{I}_r)^{-1}, \tag{4.29}$$

$$\langle\mathbf{g}_{i\cdot}\rangle^{\top} = \Sigma^{\mathbf{G}}(\langle\sigma_g\rangle\langle\mathbf{U}\rangle^{\top}\mathbf{K}_{\mathbf{u}_{\cdot i}} + \langle\beta\rangle\langle\mathbf{H}\rangle^{\top}\mathbf{y}_{i\cdot}^{\top}). \tag{4.30}$$

The jth row of **H** obeys another multivariate Gaussian distribution

$$q(\mathbf{h}_{j\cdot}) = \mathcal{N}(\mathbf{h}_{j\cdot}|\langle\mathbf{h}_{j\cdot}\rangle, \Sigma^{\mathbf{H}}), \tag{4.31}$$

with covariance and mean

$$\Sigma^{\mathbf{H}} = (\langle\beta\rangle\langle\mathbf{G}^\top\mathbf{G}\rangle + \langle\sigma_h\rangle\mathbf{I}_r)^{-1}, \tag{4.32}$$

$$\langle\mathbf{h}_{j\cdot}\rangle^\top = \Sigma^{\mathbf{H}}(\langle\sigma_h\rangle\langle\mathbf{V}\rangle^\top\mathbf{K}_{\mathbf{V}\cdot j} + \langle\beta\rangle\langle\mathbf{G}\rangle^\top\mathbf{y}_{\cdot j}). \tag{4.33}$$

The required expectations are expressed as

$$\langle\mathbf{G}^\top\mathbf{G}\rangle = \langle\mathbf{G}\rangle^\top\langle\mathbf{G}\rangle + m\Sigma^{\mathbf{G}}, \tag{4.34}$$

$$\langle\mathbf{H}^\top\mathbf{H}\rangle = \langle\mathbf{H}\rangle^\top\langle\mathbf{H}\rangle + n\Sigma^{\mathbf{H}}. \tag{4.35}$$

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,

$$q(\beta) \propto \beta^{\frac{MN}{2}-1}exp(-\frac{1}{2}\beta\langle\| \mathbf{Y} - \mathbf{GH}^\top \|_F^2\rangle), \tag{4.36}$$

with its expectation

$$\langle\beta\rangle = \frac{MN}{\langle\| \mathbf{Y} - \mathbf{GH}^\top \|_F^2\rangle}. \tag{4.37}$$

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

$$\begin{aligned}\langle\| \mathbf{Y} - \mathbf{GH}^\top \|_F^2\rangle =& \| \mathbf{Y} - \langle\mathbf{G}\rangle\langle\mathbf{H}\rangle^\top \|_F^2 + tr(N\langle\mathbf{G}\rangle^\top\langle\mathbf{G}\rangle\Sigma^{\mathbf{H}}) \\ &+ tr(M\langle\mathbf{H}\rangle^\top\langle\mathbf{H}\rangle\Sigma^{\mathbf{G}}) + tr(MN\Sigma^{\mathbf{G}}\Sigma^{\mathbf{H}}).\end{aligned} \tag{4.38}$$

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

$$\langle\sigma_g\rangle = \frac{Mr}{\langle\| \mathbf{G} - \mathbf{K_U}^\top\mathbf{U} \|_F^2\rangle}, \tag{4.39}$$

$$\langle\sigma_h\rangle = \frac{Nr}{\langle\| \mathbf{H} - \mathbf{K_V}^\top\mathbf{V} \|_F^2\rangle}, \tag{4.40}$$

with required expectations:

$$\langle\| \mathbf{G} - \mathbf{K_U}^\top\mathbf{U} \|_F^2\rangle = \| \langle\mathbf{G}\rangle - \mathbf{K_U}^\top\langle\mathbf{U}\rangle \|_F^2 + \mathrm{tr}(M\mathbf{K_U}^\top\mathbf{K_U}\Sigma^{\mathbf{U}}) + \mathrm{tr}(M\Sigma^{\mathbf{G}}), \tag{4.41}$$

$$\langle \parallel \mathbf{H}-\mathbf{K_V}^\top\mathbf{V} \parallel_F^2\rangle = \parallel \langle\mathbf{H}\rangle-\mathbf{K_V}^\top\langle\mathbf{V}\rangle \parallel_F^2 + \mathrm{tr}(N\mathbf{K_V}^\top\mathbf{K_V}\Sigma^{\mathbf{V}}) + \mathrm{tr}(N\Sigma^{\mathbf{H}}). \quad (4.42)$$

Each parameter is updated in turn while holding others fixed. By the properties of VB, convergence to a local minimum of the algorithm can be guaranteed after iterations [136].

The aim of the above inference is to recover $\mathbf{X}$ from the noisy matrix $\mathbf{Y}$ without missing data. For matrix completion, assume a subset Ω of $\mathbf{Y}$ is observed, 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. (4.11) is thus denoted as:

$$p(W_\Omega(\mathbf{Y})|\mathbf{G},\mathbf{H}) = \prod_{(i,j)\in\Omega} \mathcal{N}(y_{ij}|\mathbf{g}_{i\cdot}\mathbf{h}_{j\cdot}^\top, \beta^{-1}). \quad (4.43)$$

The corresponding joint distribution of Eq. (4.12) is modified as

$$\begin{aligned} & p(W_\Omega(\mathbf{Y}),\mathbf{U},\mathbf{V},\mathbf{G},\mathbf{H},\sigma_g,\sigma_h,\boldsymbol{\gamma},\beta) \\ =& p(W_\Omega(\mathbf{Y})|\mathbf{G},\mathbf{H},\beta)p(\mathbf{G}|\mathbf{U},\mathbf{K_U},\sigma_g)p(\mathbf{H}|\mathbf{V},\mathbf{K_V},\sigma_h) \\ & \cdot p(\mathbf{U}|\boldsymbol{\gamma})p(\mathbf{V}|\boldsymbol{\gamma})p(\sigma_g)p(\sigma_h)p(\boldsymbol{\gamma})p(\beta). \end{aligned} \quad (4.44)$$

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

$$\Sigma_i^{\mathbf{G}} = (\langle\beta\rangle\langle\mathbf{H}_\Omega^\top\mathbf{H}_\Omega\rangle + \langle\sigma_g\rangle\mathbf{I}_r)^{-1}, \quad (4.45)$$

$$\langle\mathbf{g}_{i\cdot}\rangle^\top = \Sigma_i^{\mathbf{G}}(\langle\sigma_g\rangle\langle\mathbf{U}\rangle^\top\mathbf{K}_{\mathbf{U}\cdot i} + \langle\beta\rangle\langle\mathbf{H}_\Omega\rangle^\top\mathbf{y}_{i\cdot}^\top), \quad (4.46)$$

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

$$\langle\mathbf{H}_\Omega^\top\mathbf{H}_\Omega\rangle = \sum_{j:(i,j)\in\Omega}\langle\mathbf{h}_{j\cdot}^\top\mathbf{h}_{j\cdot}\rangle = \sum_{j:(i,j)\in\Omega}\langle\mathbf{h}_{j\cdot}^\top\rangle\langle\mathbf{h}_{j\cdot}\rangle + \Sigma_j^{\mathbf{H}}, \quad (4.47)$$

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

$$\Sigma_j^{\mathbf{H}} = (\langle\beta\rangle\langle\mathbf{G}_\Omega^\top\mathbf{G}_\Omega\rangle + \langle\sigma_h\rangle\mathbf{I}_r)^{-1}, \quad (4.48)$$

$$\langle \mathbf{h}_{j\cdot} \rangle^\top = \Sigma_j^{\mathbf{H}} (\langle \sigma_h \rangle \langle \mathbf{V} \rangle^\top \mathbf{K}_{\mathbf{v}\cdot j} + \langle \beta \rangle \langle \mathbf{G}_\Omega \rangle^\top \mathbf{y}_{\cdot j}), \tag{4.49}$$

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

$$\langle \mathbf{G}_\Omega^\top \mathbf{G}_\Omega \rangle = \sum_{i:(i,j)\in\Omega} \langle \mathbf{g}_{i\cdot}^\top \mathbf{g}_{i\cdot} \rangle = \sum_{i:(i,j)\in\Omega} \langle \mathbf{g}_{i\cdot}^\top \rangle \langle \mathbf{g}_{i\cdot} \rangle + \Sigma_i^{\mathbf{G}}, \tag{4.50}$$

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:

$$\langle \beta \rangle = \frac{wMN}{\langle \| W_\Omega(\mathbf{Y}) - W_\Omega(\mathbf{G}\mathbf{H}^\top) \|_F^2 \rangle}. \tag{4.51}$$

4.3 Algorithms for image restoration

4.3.1 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 [82]. In the area of image processing, the kernel incorporating the local spatial smoothness of an image has been developed to improve image inpaitning [82]. 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 [153]. In this chapter, the aim is to apply the KSBMF model under the nonlocal framework to improve image denoising and inpainting. A new kernel is presented below which incorporates the similarity information between patches into patch group matrix factorization. Denoting the Euclidean distance between a pair of patches (i,j) by $d_E^{i,j} = \| y_{(i)} - y_{(j)} \|_2$, the similarity between them, i.e., entry of $\mathbf{K}_\mathbf{U}$ or $\mathbf{K}_\mathbf{V}$ is defined as

$$k_{ij} = \Big(\frac{1}{1 + d_E^{i,j}/M} \Big)^{\frac{1}{4}}, \tag{4.52}$$

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. The low-rankness and similarity between patches are taken into account jointly as side information to recover the image. With the kernel defined in Eq. (4.52), a nonlocal neighbor patch with larger similarity value has a more substantial contribution in the KSBMF model to recover the target patch.

4.3.2 Algorithm for image denoising

For the complete image denoising algorithm, patches are clustered with a similar spatial structure to form a patch matrix firstly. 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 [5, 38]. This scheme is implemented as

$$\mathbf{Y}^{(d+1)} = \hat{\mathbf{X}}^{(d)} + \delta(\mathbf{Y} - \hat{\mathbf{X}}^{(d)}), \tag{4.53}$$

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 4.

Algorithm 4 Image denoising by KSBMF

```
Require: 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. (4.53)
    for each patch $\mathbf{y}_i$ in $\mathbf{y}^{(d)}$ do
        Cluster similar patch to matrix $\mathbf{Y}_i$;
        Update $\mathbf{G}$ using Eq. (4.30);
        Update $\mathbf{H}$ using Eq. (4.33);
        Update $\beta$ using Eq. (4.37);
        Update $\sigma_g$ using Eq. (4.39);
        Update $\sigma_h$ using Eq. (4.40);
        Update $\mathbf{U}$ using Eq. (4.18);
        Update $\mathbf{V}$ using Eq. (4.21);
        Update $\boldsymbol{\gamma}$ using Eq. (4.24);
    end for
    Aggregate $\hat{\mathbf{X}}_i$ to form the denoised image $\hat{\mathbf{x}}^{(d)}$
end for
Ensure: denoised image $\hat{\mathbf{x}}^{(D)}$
```

4.3.3 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: KSBMF is performed on the entire image to give a proper value for each missing entry. Then the patch matching is executed and each missing value is re-filled at the patch group level. The procedure for image inpainting is summarized as Algorithm 5.

It should be noted that a straightforward extension of KSBMF to colour images often introduces perturbing colour artefacts [155, 156]. 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 [155, 156]. 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 impainting) and the aggregation are performed separately on each of the three channels. Finally, the inverse transform converts the result to an RGB image.

4.4 Experiments on image restoration

4.4.1 Parameter setting and performance evaluation

In this section the experimental results of image restoration using the KSBMF model are provided. The performance of image denoising is evaluated on twelve

Algorithm 5 Image inpainting by KSBMF

```
Require: Incomplete image $\mathbf{y}$
 1: Update $\mathbf{G}$ using Eq. (4.46);
 2: Update $\mathbf{H}$ using Eq. (4.49);
 3: Update $\beta$ using Eq. (4.51);
 4: Update $\sigma_g$ using Eq. (4.39);
 5: Update $\sigma_h$ using Eq. (4.40);
 6: Update $\mathbf{U}$ using Eq. (4.18);
 7: Update $\mathbf{V}$ using Eq. (4.21);
 8: Update $\boldsymbol{\gamma}$ using Eq. (4.24);
 9: Pre-completed image $y^{(1)}$
10: for $d = 2 : D$ do
11:     for each patch $\mathbf{y}_i$ in $\mathbf{y}^{(d)}$ do
12:         Cluster similar patches to matrix $\mathbf{Y}_i$;
13:         Repeat $1 - 8$;
14:     end for
15:     Aggregate $\hat{\mathbf{X}}_i$ to form the inpainted image $\hat{\mathbf{x}}^{(d)}$
16: end for
Ensure: Inpainted image $\hat{\mathbf{x}}^{(D)}$
```

benchmark grayscale images, shown in Fig 3.2. 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. The setting of patch size and the number of similar patches recommended in previous studies [38, 5] are adopted: 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 chapter, the scaling factor δ is fixed to 0.2 for all noise levels.

In the image inpainting problem, the algorithm is tested on part of the grayscale images in Fig 3.2 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 [35].

The kernel matrix $\mathbf{K_V}$ is set using Eq (4.52) 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.

The performance of KSBMF is evaluated 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 Eq. 3.46.

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 Eq. 3.47.

4.4.2 Image denoising

In recent years, nonlocal methods have boosted the performance of image denoising significantly. BM3D is the benchmark algorithm of image nonlocal denoising [5]. Weighted nuclear norm minimization (WNNM) [38] is always ranked as one of the most competitive methods in comparative studies. Bayesian robust matrix factorization (BPFA) [157] shares a similar principle to KSBMF in that VB is used to

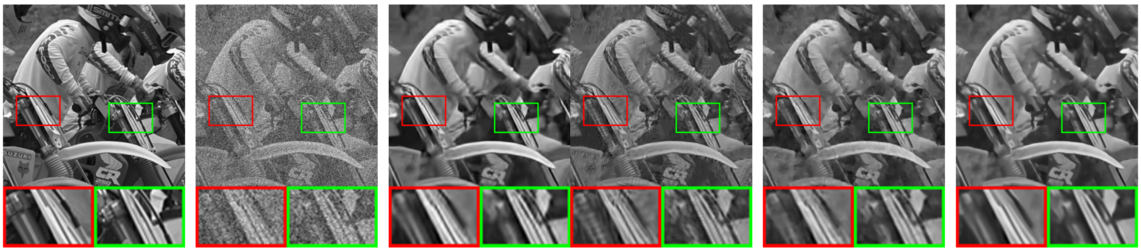
Table 4.1 : Denoising results (PSNR) by competing methods on the 12 test images. Best results are in bold.

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**

Table 4.2 : 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**

infer the factor matrices; however, the former neglects the side information. The performance of KSBMF is compared 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. The PSNR and SSIM are estimated for each scheme with $\sigma = 20, 50, 70$ and 100 dB. The PSNRs and SSIMs values are displayed in Table 4.1 and 4.2 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 of the noise level, KSBMF 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 4.2 and 4.3, the visual quality of the denoising results on four methods are compared. Two close-up views are shown at the bottom of each result for better visualization. In Fig 4.2, the Bike picture under noise level $\sigma = 50$ are compared. 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.



(a) Original (b) Noisy (c) BM3D (d) WNNM (e) BPFA (f) KSBMF

Figure 4.2 : 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= **23.11** dB).

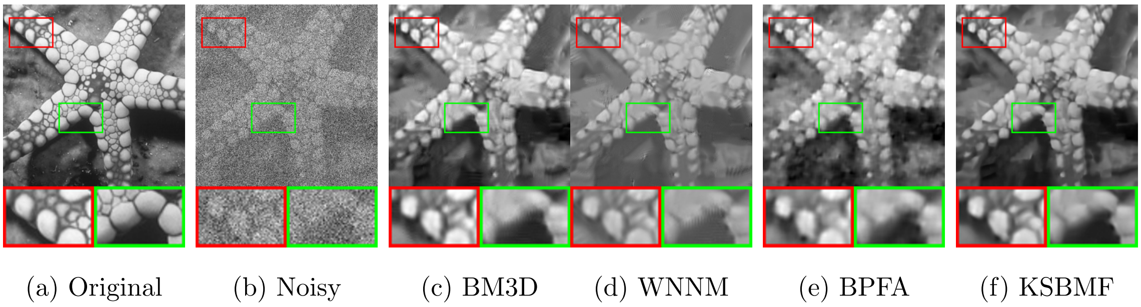

(a) Original (b) Noisy (c) BM3D (d) WNNM (e) BPFA (f) KSBMF

Figure 4.3 : 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= **20.21** dB).

In Fig 4.3, the Starfish picture under noise level $\sigma = 100$ is compared. 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.

4.4.3 Image inpainting

The performance of KSBMF on two inpainting tasks are evaluated, i.e., random missing pixels filling and text removal. The corresponding three comparative algorithms include BPFA based on Bayesian matrix factorization [157], GSR based on group sparse learning [158], and TSLRA based on nuclear norm minimization [35]. The algorithms on recovering random missing pixels are tested with four different observed percentage, i.e., 10%, 20%, 30% and 40%. The first two experiments are

Table 4.3 : 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**

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 4.3. 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.

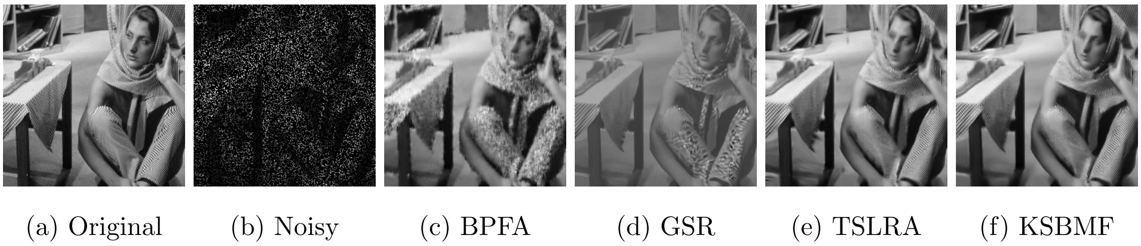
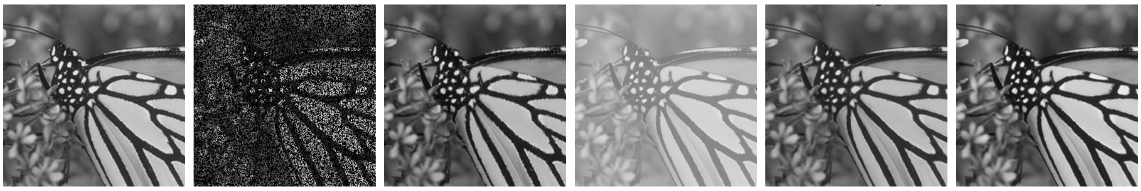

(a) Original (b) Noisy (c) BPFA (d) GSR (e) TSLRA (f) KSBMF

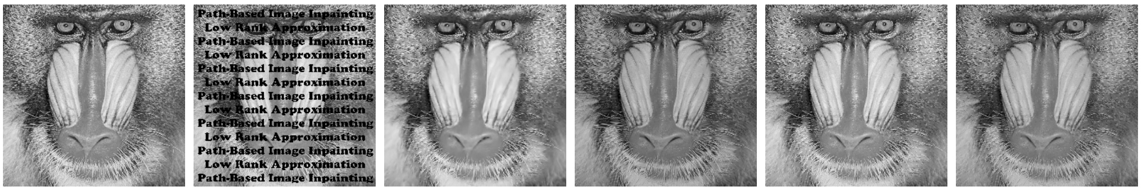
Figure 4.4 : Visual comparison for random missing pixel filling on Barbara. (a) Original image. (b) Image with 20% random samples. (c) BPFA (PSNR=22.30 dB). (d) GSR (PSNR=22.66dB). (e) TSLRA (PSNR=25.34 dB). (f) KSBMF (PSNR=**25.73** dB).



(a) Original (b) Noisy (c) BPFA (d) GSR (e) TSLRA (f) KSBMF

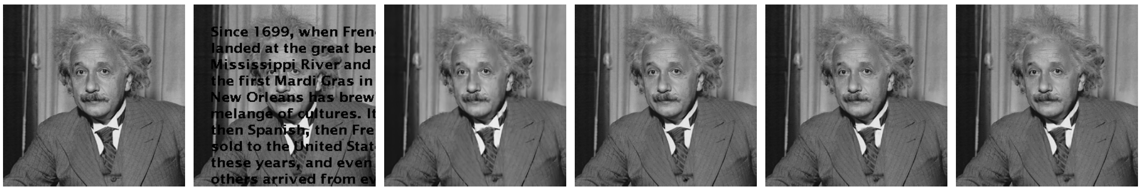
Figure 4.5 : Visual comparison for random missing pixels filling on Monarch. (a) Original image. (b) Image with 40% random samples. (c) BPFA (PSNR=29.76 dB). (d) GSR (PSNR=30.76dB). (e) TSLRA (PSNR=30.13 dB). (f) KSBMF (PSNR=**30.87** dB).

For the visual quality comparisons, Fig 4.4 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 4.5 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 by BPFA has some artifacts and blurred edges. KSBMF is then applied to remove the text on two grayscale images, and further two coluor images. The performances of competing algorithms regarding PSNR and SSIM are summarized in Table 4.3. 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 4.6-4.9 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


Path-Based Image Inpainting
Low Rank Approximation
Path-Based Image Inpainting
Low Rank Approximation
Path-Based Image Inpainting
Low Rank Approximation
Path-Based Image Inpainting
Low Rank Approximation
Path-Based Image Inpainting
Low Rank Approximation
Path-Based Image Inpainting
Low Rank Approximation
Path-Based Image Inpainting

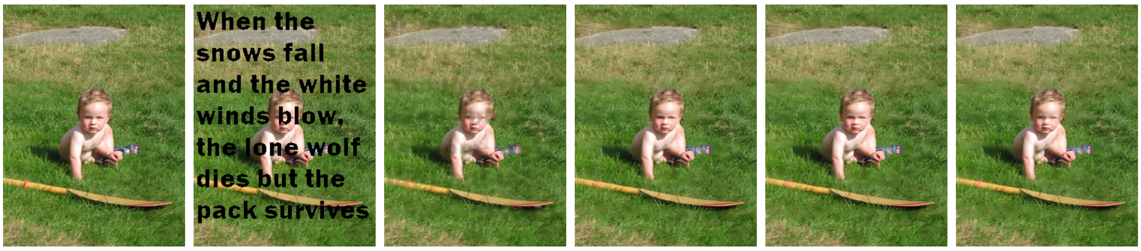
(a) Original (b) With Text 1 (c) BPFA (d) GSR (e) TSLRA (f) KSBMF

Figure 4.6 : 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

Figure 4.7 : 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).



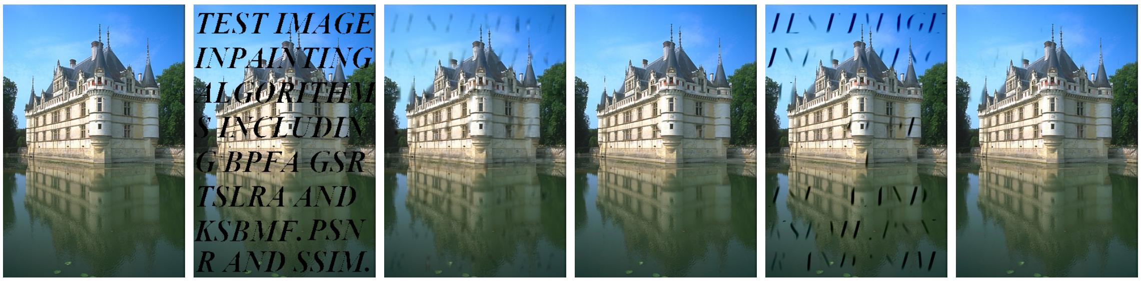
(a) Kid Original (b) With Text (c) BPFA (d) GSR (e) TSLRA (f) KSBMF

Figure 4.8 : Visual comparison for text removal on Kid. (a) Original image. (b) Image with text mask 3. (PSNR=14.33 dB) (c) BPFA (PSNR=28.02 dB). (d) GSR (PSNR=33.29 dB). (e) TSLRA (PSNR=32.93 dB). (f) KSBMF (PSNR=**33.39** dB).

less noise and reconstruction artifacts than competing approaches. However, BPFA and TSLRA can hardly remove all texts with some visible stains on the recovered Castle image.

4.5 Conclusions

This chapter has presented a new generative model for Bayesian matrix factorization which enables the incorporation of side information through kernel learning. A variational Bayesian learning principle is applied 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 similarity and low rankness properties of the patch group matrix, two image restoration algorithms are further developed which leverage KSBMF under the nonlocal framework. A new kernel is devised particularly 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



(a) Castle Original (b) With Text (c) BPFA (d) GSR (e) TSLRA (f) KSBMF

Figure 4.9 : Visual comparison for text removal on Castle. (a) Original image. (b) Image with text mask 4. (PSNR=14.50 dB) (c) BPFA (PSNR=27.70 dB). (d) GSR (PSNR=33.24 dB). (e) TSLRA (PSNR=22.90 dB). (f) KSBMF (PSNR=**33.72** dB).

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 the proposed 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 [159], 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 chapter. 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 [160].

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 only based on the low-rank assumption in recommender systems, documents labels, background subtraction, and so forth. A couple of other kernels are also devised and tested including Gaussian function and linear function. The proposed kernel in Eq. (4.52) 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.

Chapter 5

Robust kernelized Bayesian matrix factorization for video background/foreground separation

5.1 Introduction

Development of effective and efficient techniques for video analysis is an important research area in machine learning and computer vision. Matrix factorization (MF) is a powerful tool to perform such tasks. In this contribution, a hierarchical robust kernelized Bayesian matrix factorization (RKBMF) model is presented to decompose a data set into low rank and sparse components. The RKBMF model automatically infers the parameters and latent variables including the reduced rank using variational Bayesian inference. Moreover, the model integrates the side information of similarity between frames to improve information extraction from the video. RKBMF is employed to extract background and foreground information from a traffic video. Experimental results demonstrate that RKBMF outperforms state-of-the-art approaches for background/foreground separation, particularly where the video is contaminated.

Experimental results on both synthetic data sets and two real-world signals demonstrate that the proposed model achieves a satisfactory performance against several representative baselines including Lee and Seung's NMF algorithm and three other sparse or robust counterparts. The rest of this chapter is organized as follows. Section 5.2 elaborates on the details of model specification and inference for DP-NMF in the presence of different noise types. Results for both synthetic and two experimental signals, which are compared with state-of-the-art methods, and objective assessments, are presented in Section 5.3. Finally, Section 5.4 discusses and concludes the chapter.

Along the line of this research, a generative model for robust kernelized Bayesian matrix factorization (RKBMF) is presented which can integrate side information into inference. The proposed model adopts a different graphical model and priors as in [83]. A significant difference between the proposed model and [84, 85] is that the variance of a number of latent variables in [84, 85] is set as constant, which is unacceptable in the case of video analysis with an unknown noise variance. The variance of each latent factor matrix is explicitly assigned as a latent variable with a specified prior in the proposed model. The similarity information between frames is also integrated into RKBMF to improve the performance of video analysis. The performance of the model on simulated datasets is tested and then this algorithm is applied to perform the video background and foreground separation. The results demonstrated that RKBMF can accurately recover both low rank and sparse components in simulation, and generate background and foreground images with better visual effects than other three state-of-the-art robust matrix factorization approaches.

5.2 Robust kernelized Bayesian matrix factorization

In this section, the model specification of robust kernelized Bayesian matrix factorization is elaborated in detail. The variational Bayesian method is utilized to infer all parameters and latent variables of this model in detail.

5.2.1 Model specification

Considering the observation data as an $M \times N$ matrix $\mathbf{Y}$, the problem is to recover the original low-rank matrix $\mathbf{X}$ and sparse term $\mathbf{S}$, that is:

$$\mathbf{Y} = \mathbf{X} + \mathbf{S} + \mathbf{E} = \mathbf{U}\mathbf{V}^\top + \mathbf{S} + \mathbf{E}, \tag{5.1}$$

where $\mathbf{Y} \in \mathbb{R}^{M\times N}$, $\mathbf{U} \in \mathbb{R}^{M\times r}$, $\mathbf{V} \in \mathbb{R}^{N\times r}$, $\mathbf{S} \in \mathbb{R}^{M\times N}$, $\mathbf{E} \in \mathbb{R}^{M\times N}$, and r the rank or order of the low-rank term.

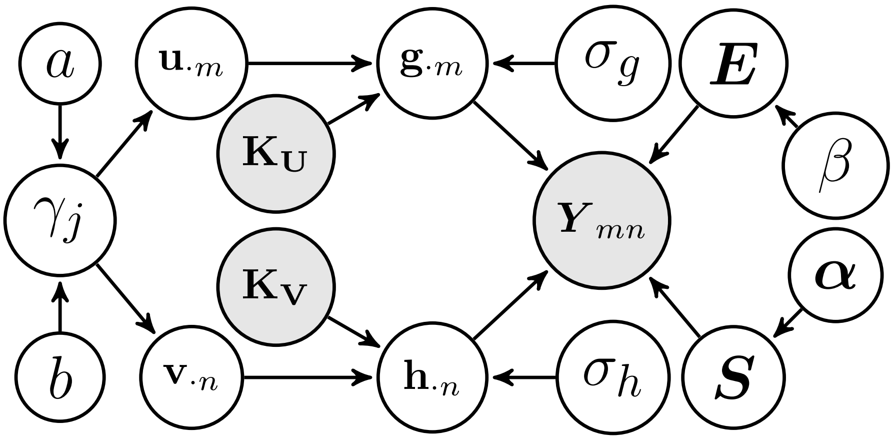


Figure 5.1 : Directed graphical representation of RKBMF model.

Fig 5.1 shows the graphical model of the proposed hierarchical robust kernelized Bayesian matrix factorization with latent variables and their corresponding priors. To automatically infer the rank of the low rank component, sparsity is imposed into the low rank approximation model.

The priors of variables $\mathbf{U}, \mathbf{V}, \boldsymbol{\gamma}, \mathbf{G}, \mathbf{H}, \sigma_g, \sigma_h$ and $\mathbf{E}$ are defined the same with Chapter 4.2, so they are omitted here.

The sparse component $\mathbf{S}$ is modeled with independent priors on each of the entries $\mathbf{S}_{ij}$ of the matrix $\mathbf{S}$, that is

$$p(\mathbf{S}|\boldsymbol{\alpha}) = \prod_{i=1}^{M}\prod_{j=1}^{N}\mathcal{N}(s_{ij}|0, \alpha_{ij}^{-1}). \tag{5.2}$$

Given the priors defined above, the conditional distribution for the observation model is as follows:

$$p(\mathbf{Y}|\mathbf{G}, \mathbf{H}, \mathbf{S}, \beta) = \mathcal{N}(\mathbf{Y}|\mathbf{G}\mathbf{H}^{\top} + \mathbf{S}, \beta^{-1}\mathbf{I}_{MN}). \tag{5.3}$$

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

$$\begin{aligned} & p(\mathbf{Y}, \mathbf{U}, \mathbf{V}, \mathbf{S}, \mathbf{G}, \mathbf{H}, \sigma_g, \sigma_h, \boldsymbol{\gamma}, \boldsymbol{\alpha}, \beta) \\ & = p(\mathbf{Y}|\mathbf{G}, \mathbf{H}, \mathbf{S}, \beta) p(\mathbf{G}|\mathbf{U}, \mathbf{K}_{\mathbf{U}}, \sigma_g) p(\mathbf{H}|\mathbf{V}, \mathbf{K}_{\mathbf{V}}, \sigma_h) \\ & \cdot p(\mathbf{U}|\boldsymbol{\gamma}) p(\mathbf{V}|\boldsymbol{\gamma}) p(\mathbf{S}|\boldsymbol{\alpha}) p(\sigma_g) p(\sigma_h) p(\boldsymbol{\gamma}) p(\boldsymbol{\alpha}) p(\beta). \end{aligned} \tag{5.4}$$

5.2.2 Model inference of RKBMF

Z is utilizeds to represent the vector of all latent variables such that

$$\mathbf{Z} = (\mathbf{U}, \mathbf{V}, \mathbf{G}, \mathbf{H}, \mathbf{S}, \sigma_g, \sigma_h, \boldsymbol{\gamma}, \boldsymbol{\alpha}, \beta). \tag{5.5}$$

The approximate posterior distribution is therefore denoted by $q(\mathbf{Z})$.

$$q(\mathbf{Z}) = \prod_k q(\mathbf{Z}_k). \tag{5.6}$$

Within the VB framework, the expression of the optimal posterior approximation $q(\mathbf{Z}_k)$ can be denoted as

$$ln\ q(\mathbf{Z}_k) = \langle ln\ p(\mathbf{Y}, \mathbf{Z}) \rangle_{\mathbf{Z} \setminus \mathbf{Z}_k} + const, \tag{5.7}$$

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 $\mathbf{Z}_k$ removed. Each variable is updated in turn while holding others fixed. The iteration rules for variables $\mathbf{S}, \mathbf{G}, \mathbf{H}$, $\boldsymbol{\alpha}$ and β in Eq. (5.6) is detailed below. The update equations for other variables are the same with Section 4.2.

Estimation of sparse component S

The posterior distribution of $\boldsymbol{S}$ is decomposed on its each entries s_{ij} which is a Gaussian distribution

$$q(s_{ij}) = \mathcal{N}(s_{ij} | \langle s_{ij} \rangle, \Sigma_{ij}^{\boldsymbol{S}}). \tag{5.8}$$

The covariance and mean are denoted as

$$\Sigma_{ij}^{\boldsymbol{S}} = \frac{1}{\langle\beta\rangle + \langle\alpha_{ij}\rangle}, \tag{5.9}$$

$$\langle s_{ij}\rangle = \langle\beta\rangle\Sigma_{ij}^{\boldsymbol{S}}(y_{ij} - \langle\boldsymbol{g}_{i\cdot}\rangle\langle\boldsymbol{h}_{i\cdot}^{\top}\rangle). \tag{5.10}$$

Estimation of G *and* H

Similar to the estimation of **U** and **V**, the posterior approximation of **G** is given by

$$\begin{aligned} ln\ q(\mathbf{G}) =& \sum_i [-\frac{1}{2}(\mathbf{g}_{i\cdot}(\langle\beta\rangle\langle\mathbf{H}^{\top}\mathbf{H}\rangle + \langle\sigma_g\rangle\mathbf{I}_r)\mathbf{g}_{i\cdot}^{T} \\ & - 2\mathbf{g}_{i\cdot}(\langle\mathbf{H}\rangle^{\top}(\mathbf{y}_{i\cdot} - \mathbf{s}_{i\cdot})^{\top} + \langle\sigma_g\rangle\langle\mathbf{U}\rangle^{\top}\mathbf{K}_{\mathbf{U}\cdot i}))] + const, \end{aligned} \tag{5.11}$$

which indicates that the ith row of **G** obeys the multivariate Gaussian distribution

$$q(\mathbf{g}_{i\cdot}) = \mathcal{N}(\mathbf{g}_{i\cdot}|\langle\mathbf{g}_{i\cdot}\rangle, \Sigma^{\mathbf{G}}). \tag{5.12}$$

The corresponding covariance and mean are denoted as

$$\Sigma^{\mathbf{G}} = (\langle\beta\rangle\langle\mathbf{H}^{\top}\mathbf{H}\rangle + \langle\sigma_g\rangle\mathbf{I}_r)^{-1}, \tag{5.13}$$

$$\langle\mathbf{g}_{i\cdot}\rangle^{\top} = \Sigma^{\mathbf{G}}(\langle\sigma_g\rangle\langle\mathbf{U}\rangle^{\top}\mathbf{K}_{\mathbf{u}\cdot i} + \langle\beta\rangle\langle\mathbf{H}\rangle^{\top}(\mathbf{y}_{i\cdot} - \mathbf{s}_{i\cdot})^{\top}). \tag{5.14}$$

The jth row of **H** obeys another multivariate Gaussian distribution

$$q(\mathbf{h}_{j\cdot}) = \mathcal{N}(\mathbf{h}_{j\cdot}|\langle\mathbf{h}_{j\cdot}\rangle, \Sigma^{\mathbf{H}}), \tag{5.15}$$

with covariance and mean

$$\Sigma^{\mathbf{H}} = (\langle\beta\rangle\langle\mathbf{G}^{\top}\mathbf{G}\rangle + \langle\sigma_h\rangle\mathbf{I}_r)^{-1}, \tag{5.16}$$

$$\langle\mathbf{h}_{j\cdot}\rangle^{\top} = \Sigma^{\mathbf{H}}(\langle\sigma_h\rangle\langle\mathbf{V}\rangle^{\top}\mathbf{K}_{\mathbf{V}\cdot j} + \langle\beta\rangle\langle\mathbf{G}\rangle^{\top}(\mathbf{y}_{\cdot j} - \mathbf{s}_{\cdot j}). \tag{5.17}$$

The required expectations are expressed as

$$\langle\mathbf{G}^{\top}\mathbf{G}\rangle = \langle\mathbf{G}\rangle^{\top}\langle\mathbf{G}\rangle + m\Sigma^{\mathbf{G}}, \tag{5.18}$$

$$\langle\mathbf{H}^{\top}\mathbf{H}\rangle = \langle\mathbf{H}\rangle^{\top}\langle\mathbf{H}\rangle + n\Sigma^{\mathbf{H}}. \tag{5.19}$$

Estimation of β

The posterior probability densities of β, σ_g and σ_h are all found to be Gamma distributed. For the noise precision β,

$$q(\beta) \propto \beta^{\frac{MN}{2}-1} exp(-\frac{1}{2}\beta\langle \| \mathbf{Y} - \mathbf{G}\mathbf{H}^\top - \mathbf{S} \|_F^2 \rangle), \tag{5.20}$$

with its expectation

$$\langle \beta \rangle = \frac{MN}{\langle \| \mathbf{Y} - \mathbf{G}\mathbf{H}^\top - \mathbf{S} \|_F^2 \rangle}. \tag{5.21}$$

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

$$\begin{aligned} \langle \| \mathbf{Y} - \mathbf{G}\mathbf{H}^\top - \mathbf{S} \|_F^2 \rangle =& \| \mathbf{Y} - \langle \mathbf{G} \rangle \langle \mathbf{H} \rangle^\top - \langle \mathbf{S} \rangle \|_F^2 + tr(N \langle \mathbf{G} \rangle^\top \langle \mathbf{G} \rangle \Sigma^{\mathbf{H}}) \\ &+ tr(M \langle \mathbf{H} \rangle^\top \langle \mathbf{H} \rangle \Sigma^{\mathbf{G}}) + tr(MN \Sigma^{\mathbf{G}} \Sigma^{\mathbf{H}}) + tr(\sum_{i=1}^{M} \sum_{j=1}^{N} \Sigma_{ij}^{\boldsymbol{S}}). \end{aligned} \tag{5.22}$$

Estimation of $\boldsymbol{\alpha}$

Similar to β, σ_g and σ_h, the posterior probability density of α_{ij} is also found to be a Gamma distribution with

$$\langle \alpha_{ij} \rangle = \frac{1}{\langle s_{ij} \rangle^2 + \Sigma_{ij}^{\boldsymbol{S}}} \tag{5.23}$$

Each parameter is updated in turn while holding others fixed. By the properties of VB, convergence to a local minimum of the algorithm can be guaranteed after iterations [136].

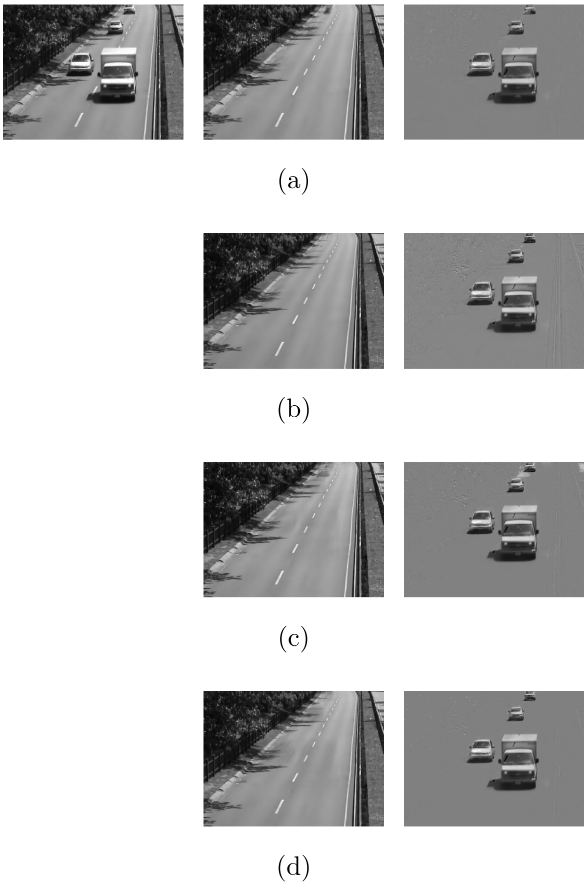
The proposed RKBMF model is applied with integrated side information to improve background subtraction and foreground detection. The kernel proposed in Eq. 4.52 is utilized to incorporate the similarity information between video frames into RKBMF.

In the proposed method, the target frame is first vectorized as a column vector. The $M \times N$ matrix $\mathbf{Y}$ is constructed by grouping other $N - 1$ frames with similar local spatial structures to the underlying one. Since each column shares similar underlying image structures, the noise-free low-rank matrix $\mathbf{X}$ corresponds to the background, while the sparse component corresponds to the foreground. With the kernel defined in Eq. (4.52), a similar frame with larger similarity value has a more substantial contribution in the RKBMF model to separate the background and foreground.

5.3 Results

5.3.1 Numerical simulation

The performance of the proposed algorithm is tested on simulated matrices firstly. Four square matrices with size $M = N = 500$, 1000, 1500 and 2000 are considered. The low-rank component $\mathbf{X}$ is simulated by the product of two matrices whose entries are independently drawn from $\mathcal{N}(0, 1/M)$. The sparse component $\mathbf{S}$ is simulated by the non-zero entries located uniformly at random with amplitudes obeyed uniform distribution within the range of $[-1, 1]$. The observation is generated as $\mathbf{Y} = \mathbf{X} + \mathbf{S} + \mathbf{E}$ with entries of $\mathbf{E}$ independently drawn from $\mathcal{N}(0, 10^{-4})$. The hyperparameters related to $\boldsymbol{\alpha}$, β, σ_g, σ_h are specified with a relatively small value, i.e., 10^{-6}. Three metrics, i.e., rank($\hat{\mathbf{X}}$), $\parallel \hat{\mathbf{S}} \parallel_0$ and $\parallel \hat{\mathbf{S}} - \mathbf{S} \parallel_F / \parallel \mathbf{S} \parallel_F$ are used to evaluate the performance of the algorithm. In this simulation, since no side information is available, it is reasonable to set $\boldsymbol{K_U}$ and $\boldsymbol{K_V}$ as identity matrices. From Table 5.1, it is clear that RKBMF can accurately approximate the rank of the low-rank component and $\parallel \hat{\mathbf{S}} \parallel_0$ with a very small reconstruction error of $\parallel \hat{\mathbf{S}} - \mathbf{S} \parallel_F / \parallel \mathbf{S} \parallel_F$.



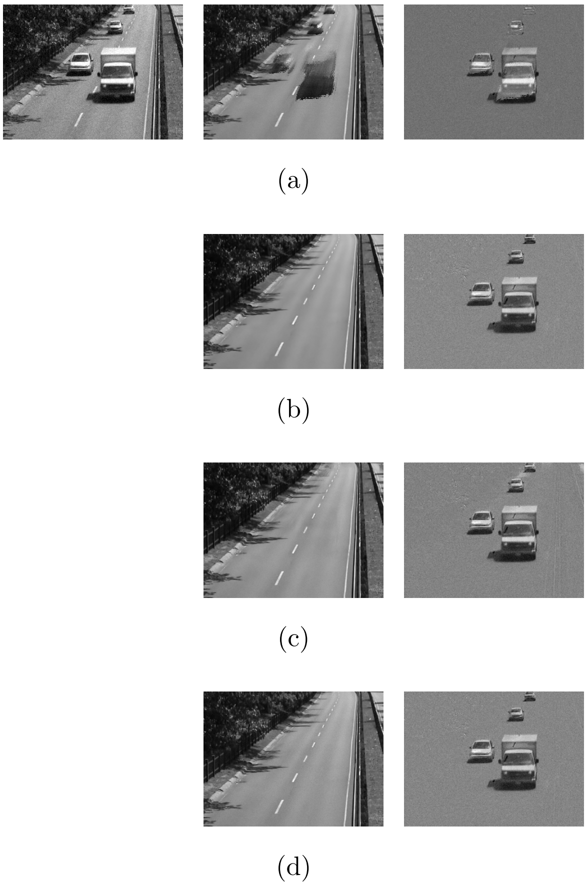
(a)

(b)

(c)

(d)

Figure 5.2 : Reconstruction of the background and the foreground. The video sequence contains 520 frames of size 320 × 240 pixels, and the results for frame 260 are shown. Left column: original image; middle: reconstruction of the low-rank component (background); and right: reconstruction of the sparse component (foreground). (a) Bayesian Robust PCA, (b) Mixture of Gaussians RPCA, (c) Online Stochastic Tensor Decomposition and (d) RKBMF.



(a)

(b)

(c)

(d)

Figure 5.3 : Reconstruction of the background and the foreground under noisy observation. The additive white Gaussian noise has a standard deviation $\sigma = 10$. Left column: original noisy image; middle: background reconstruction; and right: foreground reconstruction. (a) Bayesian Robust PCA, (b) Mixture of Gaussians RPCA, (c) Online Stochastic Tensor Decomposition and (d) RKBMF.

Table 5.1 : Comparison of reconstruction accuracy for noisy observation, with noise standard deviation $\sigma = 10^{-4}$. The true rank of the matrix $\mathbf{X}$ is $5\%N$, and the number of nonzero sparse elements is $5\%MN$.

N	rank($\mathbf{X}$)	$\parallel \mathbf{S} \parallel_0$	rank($\hat{\mathbf{X}}$)	$\frac{\|\hat{\mathbf{S}}-\mathbf{S}\|_F}{\|\mathbf{S}\|_F}$	$\parallel \hat{\mathbf{S}} \parallel_0$
500	25	12500	25	3.1×10^{-5}	12498
1000	50	50000	50	2.5×10^{-5}	50003
1500	75	112500	75	3.3×10^{-5}	112500
2000	100	200000	100	2.9×10^{-5}	199990

5.3.2 Video Example

The performance of RKBMF to reconstruct the static background and moving foreground from a video sequence in traffic surveillance with a fixed camera[*] is evaluated. Experiments are also conducted using Bayesian robust PCA [75], mixture of Gaussians RPCA [79], and online stochastic tensor decomposition [161], for comparison. The data are organized such that column of is constructed by concatenating all pixels of the frame from a grayscale video sequence. The background is then modeled as the low-rank component, and the moving foreground is modeled as the sparse component. The hyperparameters are the same as in 5.3.1. The kernel function for the latent factor matrix $\boldsymbol{K_V}$ is estimated using Eq. (4.52) while the kernel function for $\boldsymbol{K_U}$ is set as an identity matrix. The video sequence contains 520 frames of 320 × 240 pixels. Fig 5.2 shows the reconstruction of the background and the foreground for Frame 260 over four methods. It is observed that the four models produce good reconstructed background/foreground in this situation since the observation is relatively noise-free.

[*] http://jacarini.dinf.usherbrooke.ca/dataset2012/

Gaussian white noise with standard deviation 20 is then added into the video sequence. Such noisy observations are common in practical applications. Fig 5.3 shows the reconstruction results of all four methods. It is clear that RKBMF still successfully separates the foreground from the background. However, Bayesian Robust PCA fails to separate background/foreground with part of the foreground existing in the low-rank component. Artifacts can still be found on the foreground extracted by the mixture of Gaussians RPCA and online stochastic tensor decomposition. In contrast, RKBMF generates the overall best background and foreground with fewer artifacts. To further evaluate the performance of the proposed model, RKBMF and the competitive algorithms are tested on the CAVIAR Test Case Scenarios[†]. The experimental results are similar to the case of the traffic surveillance with RKBMF yielding the best separation performance.

5.4 Conclusions

In this chapter, a novel full Bayesian model for robust matrix factorization is proposed which integrates the side information for the low rank and sparse component extraction. Using both synthetic and real datasets, experimental results show that the proposed method outperforms other three state-of-the-art robust matrix factorization approaches. In particular, the proposed method can recover the background and slow-moving foreground even under high noise level. RKBMF can be further improved to accommodate streaming video and to integrate multiple side information.

[†] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

Chapter 6

Bayesian nonnegative matrix factorization with Dirichlet process mixtures

6.1 Introduction

Nonnegative Matrix Factorization (NMF) is valuable in many applications of blind source separation, signal processing and machine learning. A number of algorithms that can infer nonnegative latent factors have been developed, but most of these assume a specific noise kernel. This is insufficient to deal with complex noise in real scenarios. In this chapter, a hierarchical Dirichlet process nonnegative matrix factorization (DPNMF) model is proposed in which the Gaussian mixture model is used to approximate the complex noise distribution. Moreover, the model is cast in a nonparametric Bayesian framework by using a Dirichlet process mixture to infer the necessary number of Gaussians. A mean-field variational inference algorithm is derived for the proposed nonparametric Bayesian model. The model is tested on synthetic data sets contaminated by Gaussian, sparse and mixed noise firstly. Then it is applied to select discriminative features for motor imagery single trial electroencephalogram (EEG) classification and to extract muscle synergies from the electromyographic (EMG) signal. Experimental results demonstrate that DPNMF performs better in extracting the latent nonnegative factors in comparison with state-of-the-art methods.

The main contributions of this chapter are summarized as follows. (a) To deal with complex noise in real scenarios, a hierarchical Dirichlet process nonnegative matrix factorization model is proposed. (b) The Gausian mixture model is utilized as a universal approximator to fit various types of noise rather than a single noise kernel in existing NMF models. (c)A nonparametric Bayesian technique, i.e., Dirichlet process, is employed to determine the number of Gaussians needed, instead of doing heuristic pruning or trying ungrounded guesses. (d) The model is formulated into the variational Bayesian update rules instead of the usual multiplicative updating rules for NMF. (e) It is demonstrated that DPNMF significantly improves the performance of two real-world problems, i.e., muscle synergies extraction and movement imagery EEG classification, which heavily rely on the NMF technique.

Experimental results on both synthetic data sets and two real-world signals (EEG and EMG) demonstrate that the proposed DPNMF model achieves a satisfactory performance against several representative baselines including Lee and Seung's NMF algorithm and three other sparse or robust counterparts. The rest of this chapter is organized as follows. Section 6.2 elaborates on the details of model specification and inference for DPNMF in the presence of different noise types. Results for both synthetic and two experimental signals, which are compared with state-of-the-art methods, and objective assessments, are presented in Section 6.3. Finally, Section 6.4 discusses and concludes the chapter.

6.2 DPNMF model and inference

6.2.1 Model specification of DPNMF

For the observation matrix $\mathbf{Y} \in \mathbb{R}^{m\times n}$, nonnegative matrix factorization can be formulated as decomposing $\mathbf{Y}$ into two latent matrices $\mathbf{U} \in \mathbb{R}_{+}^{m\times r}$ and $\mathbf{V} \in \mathbb{R}_{+}^{n\times r}$, whose values are constrained to be positive. In other words, the task is to solve

$$\mathbf{Y} = \mathbf{U}\mathbf{V}^{\top} + \mathbf{E}, \tag{6.1}$$

where $\mathbf{E} \in \mathbb{R}^{m\times n}$ represents the noise term. The variational Bayesian approach is employed to this problem. Fig 6.1 shows the graphical model of the proposed

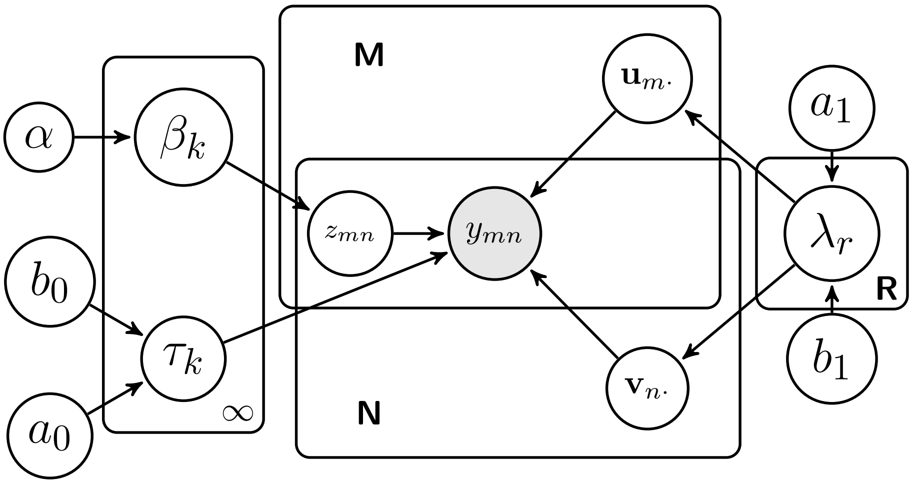


Figure 6.1 : Directed graphical representation of DPNMF model

hierarchical Bayesian Dirichlet process nonnegative matrix factorization with latent variables and their corresponding priors. In particular, the mixture of Gaussian distributions is utilized to model the noise:

$$p(e_{mn}) = \sum_{k=1}^{\infty} \theta_k \mathcal{N}(0, \tau_k), \tag{6.2}$$

where $\theta_k = \beta_k \prod_{l=1}^{k-1}(1 - \beta_l)$ and β_k is drawn independently from Beta distribution $\mathcal{B}(1, \alpha)$ according to the stick-breaking construction [129]. In the model, a_0, b_0, α, a_1 and b_1 are hyperparameters. Let z_{mn} be a latent variable that assigns the index of the parameter associated with the entry e_{mn}. The distribution of z_{mn} can be regarded as a multinomial distribution with parameters $\{\theta_1, \cdots, \theta_\infty\}$. In practice, a relatively large K is set as the initial number of Gaussians to fit the noise term.

A Gamma distribution with shape $a_0 > 0$ and rate $b_0 > 0$ is utilized to model the precision τ_k,

$$p(\tau_k) \sim \mathcal{G}(a_0, b_0). \tag{6.3}$$

An exponential prior is set over $\mathbf{U}$ and $\mathbf{V}$. In order to automatically prune the rank of $\mathbf{U}$ and $\mathbf{V}$, rate parameters λ_r is assigned to the exponential prior of both columns of $\mathbf{u}_{\cdot r}$ and $\mathbf{v}_{\cdot r}$.

$$u_{mr} \sim f(u_{mr}|\lambda_r), \tag{6.4}$$

$$v_{nr} \sim f(v_{nr}|\lambda_r), \tag{6.5}$$

where $f(x|\lambda) = \lambda exp(-\lambda x)s(x)$ is the density of the exponential distribution, and $s(x)$ is the unit step function. With the constraint of the same rate parameters λ_r across $\mathbf{u}_{\cdot r}$ and $\mathbf{v}_{\cdot r}$, most of the rate parameters λ_r 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 $\mathbf{Y}$, and hence the rank of latent factors $\mathbf{U}$ and $\mathbf{V}$ are automatically determined.

According to the previous studies, the likelihood obeys a Gaussian distribution. Combining the likelihood and the priors, the joint distribution can be formulated as:

$$\begin{aligned} & p(\mathbf{Y}, \mathbf{U}, \mathbf{V}, \mathbf{z}, \boldsymbol{\tau}, \boldsymbol{\lambda}, \boldsymbol{\beta}|a_0, b_0, a_1, b_1, \alpha) \\ & = p(\boldsymbol{Y}|\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{z}, \boldsymbol{\tau})p(\boldsymbol{U}|\boldsymbol{\lambda})p(\boldsymbol{V}|\boldsymbol{\lambda})p(\boldsymbol{\lambda}|a_1, b_1) \\ & \cdot p(\boldsymbol{\tau}|a_0, b_0)p(\boldsymbol{z}|\boldsymbol{\beta})p(\boldsymbol{\beta}|\alpha) \end{aligned} \tag{6.6}$$

Based on the mean-field variational approach, the goal changes to infer the posterior of all variables using the following variational distribution:

$$\begin{aligned} & q(\boldsymbol{Y}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{z}, \boldsymbol{\tau}, \boldsymbol{\lambda}, \boldsymbol{\beta}) = \prod_{m=1}^{M}\prod_{r=1}^{R} q_{u_{mr}}(\mu^U_{mr}, \tau^U_{mr}) \\ & \prod_{n=1}^{N}\prod_{r=1}^{R} q_{v_{nr}}(\mu^V_{nr}, \tau^V_{nr}) \prod_{r=1}^{R} q_{\lambda_r}(a^*_r, b^*_r) \prod_{k=1}^{K} q_{\beta_k}(\gamma_{k,1}, \gamma_{k,2}) \\ & \prod_{k=1}^{K} q_{\tau_k}(\rho_{k,1}, \rho_{k,2}) \prod_{m=1}^{M}\prod_{n=1}^{N} q_{z_{mn}}(\phi_{mn}), \end{aligned} \tag{6.7}$$

where each entry of $\mathbf{U}$ follows a truncated normal distribution with mean μ^U_{mr} and covariance τ^U_{mr} and it is similar for $\mathbf{V}$, τ_k and λ_r follow gamma distribution parametrized by $\rho_{k,1}$, $\rho_{k,2}$ and a^*_r, b^*_r, respectively, $q_{\beta_k}(\gamma_{k,1}, \gamma_{k,2})$ is a beta distribution, and $q_{z_{mn}}(\phi_{mn})$ is a multinomial distribution.

6.2.2 Model inference of DPNMF

Parameter β_k can be derived from its mean-field posterior representation.

$$ln\ q(\beta_k) = \mathbb{E}_q[ln\ p(\beta_k|\alpha)] + \prod_{m=1}^{M}\prod_{n=1}^{N}\mathbb{E}_q[ln\ p(z_{mn} = k|\boldsymbol{\theta})] + const. \tag{6.8}$$

Since $p(z_{mn}|\boldsymbol{\theta})$ is a multinomial distribution, expanding $\boldsymbol{\theta}$ and using the assumption that $q(\beta_K = 1) = 1$, it is obvious that $1 - \beta_K = 0$, and $q(z_{mn} > K) = 0$, then

$$\begin{aligned}\mathbb{E}_q[ln\ p(z_{mn}|\boldsymbol{\theta})] &= \mathbb{E}_q[ln(\prod_{k=1}^{\infty}(1-\beta_k)^{\mathbb{I}(z_{mn}>k)}\beta_k^{\mathbb{I}(z_{mn}=k)})] \\ &= \sum_{k=1}^{K} q(z_{mn} > k)\mathbb{E}_q ln\ (1-\beta_k) + q(z_{mn} = k)\mathbb{E}_q ln\ \beta_k.\end{aligned} \tag{6.9}$$

Since $q(z_{mn} > K) = 0$, then $q(z_{mn} > k) = \sum_{t=k+1}^{K} q(z_{mn} = t)$. Substitute Eq. (6.9) into Eq. (6.8) and simplify it to

$$\begin{aligned}ln\ q(\beta_k) &= const + (\sum_{m=1}^{M}\sum_{n=1}^{N} q(z_{mn} = k))ln\ \beta_k \\ &+ (\alpha - 1 + \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{t=k+1}^{K} q(z_{mn} = t))ln\ (1-\beta_k).\end{aligned} \tag{6.10}$$

Obviously, $q(\beta_k)$ follows a Beta distribution,

$$q(\beta_k) = \text{Beta}(\beta_k|\gamma_{k,1}, \gamma_{k,2}), \tag{6.11}$$

with

$$\gamma_{k,1} = 1 + \sum_{m=1}^{M}\sum_{n=1}^{N}\phi_{mnk}, \tag{6.12}$$

$$\gamma_{k,2} = \alpha + \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{t=k+1}^{K}\phi_{mnt}, \tag{6.13}$$

and

$$\mathbb{E}_q(ln\ \beta_k) = \psi(\gamma_{k,1}) - \psi(\gamma_{k,1} + \gamma_{k,2}), \tag{6.14}$$

$$\mathbb{E}_q(ln\ (1-\beta_k)) = \psi(\gamma_{k,2}) - \psi(\gamma_{k,1} + \gamma_{k,2}), \tag{6.15}$$

where ψ denotes the digamma function.

Then the parameters related to z_{mn} can be updated based on the stick-breaking process.

$$\begin{aligned}
&\ln q(z_{mn}=k)\\
&=\mathbb{E}_{\boldsymbol{Z}/z_{mn}=k}[ln\ p(\boldsymbol{Y}|\boldsymbol{U},\boldsymbol{V},\boldsymbol{z},\boldsymbol{\tau})+ln\ p(\boldsymbol{z}|\boldsymbol{\beta})]\\
&=\mathbb{E}_q\ ln\ \sqrt{\frac{\tau_{z_{mn}=k}}{2\pi}}exp\{-\frac{\tau_{z_{mn}=k}}{2}(y_{mn}-\boldsymbol{u}_{m\cdot}\boldsymbol{v}_{n\cdot}^{\top})^2\\
&+\mathbb{I}(z_{mn}>k)\mathbb{E}_q ln\ (1-\beta_k)+\mathbb{I}(z_{mn}=k)\mathbb{E}_q ln\ \beta_k\}\\
&=\frac{1}{2}[\psi(\rho_{k,1})-log\rho_{k,2}]-\frac{\rho_{k,1}}{2\rho_{k,2}}\mathbb{E}_q\{(y_{mn}-\boldsymbol{u}_{m\cdot}\boldsymbol{v}_{n\cdot}^{\top})^2\\
&+\sum_{t=1}^{k-1}\mathbb{E}_q ln\ (1-\beta_t)+\mathbb{I}(z_{mn}=k)\mathbb{E}_q ln\ \beta_k\}
\end{aligned}\tag{6.16}$$

According to Eq. (6.16), it is clear that $q(z_{mn})$ follows a multinomial distribution, and its parameters ϕ_{mnk} for $k=1,\ 2,\cdots K$ can be represented as:

$$\begin{aligned}
&\phi_{mnk}\propto exp\{\frac{1}{2}(\psi(\rho_{k,1})-log\ \rho_{k,2})+\mathbb{E}_q ln\ \beta_k\\
&-\frac{1}{2}\frac{\rho_{k,1}}{\rho_{k,2}}\mathbb{E}_q\{(y_{mn}-\boldsymbol{u}_{m\cdot}\boldsymbol{v}_{n\cdot}^{\top})^2\}+\sum_{t=1}^{k-1}\mathbb{E}_q ln\ (1-\beta_t)\}.
\end{aligned}\tag{6.17}$$

The approximation to the posterior of entry u_{mr} can be expressed as

$$\begin{aligned}
&q(u_{mr})\propto exp\{\mathbb{E}_q[log\ p(\boldsymbol{Y}|\boldsymbol{U},\boldsymbol{V},\boldsymbol{z},\boldsymbol{\tau})+log\ p(u_{mr}|\lambda_r)]\}\\
&\propto s(x)exp\{log\ (\lambda_r exp\{-\lambda_r u_{mr}\})+\mathbb{E}_q\sum_{n\in\Omega_m}\sum_{k=1}^{K}\\
&\left(p(z_{mn}=k)log\ [\sqrt{\frac{\tau_k}{2\pi}}exp\{-\frac{\tau_k}{2}(y_{mn}-\mathbf{u}_m\mathbf{v}_n^{\top})^2\}]\right)\}\\
&\propto exp\{-\frac{u_{mr}^2}{2}[\sum_{n\in\Omega_m}\sum_{k=1}^{K}\phi_{mnk}\langle\tau_k\rangle\langle v_{nr}^2\rangle]+u_{mr}[-\langle\lambda_r\rangle+\\
&\sum_{n\in\Omega_m}\sum_{k=1}^{K}\phi_{mnk}\langle\tau_k\rangle(y_{mn}-\sum_{r'\neq r}\langle u_{mr'}\rangle\langle v_{nr'}\rangle)\langle v_{nr}\rangle]\}s(x)\\
&\propto exp\{-\frac{\tau_{mr}^u}{2}(u_{mr}-\mu_{mr}^u)^2\}\times s(x)\\
&\propto\mathcal{TN}(u_{mr}|\mu_{mr}^U,\tau_{mr}^U),
\end{aligned}\tag{6.18}$$

where $\langle\cdot\rangle$ is the expectation operator, Ω is all the entries of the matrix and $\Omega_m = \{m|(m,n) \in \Omega\}$ and $\Omega_n = \{n|(m,n) \in \Omega\}$. Eq. 6.18 demonstrates that $q(u_{mr})$ obeys a truncated normal (TN) distribution.

$$\mu_{mr}^U = \frac{1}{\tau_{mr}^U}(-\langle\lambda_r\rangle + \sum_{n\in\Omega_m}\sum_{k=1}^{K}\phi_{mnk}\langle\tau_k\rangle(y_{mn} - \sum_{r'\neq r}\langle u_{mr'}\rangle\langle v_{nr'}\rangle)\langle v_{nr}\rangle), \tag{6.19}$$

$$\tau_{mr}^U = \sum_{n\in\Omega_m}\sum_{k=1}^{K}\phi_{mnk}\langle\tau_k\rangle\langle v_{nr}^2\rangle. \tag{6.20}$$

Similarly, the entry v_{nr} of factor matrix $\mathbf{V}$ obeys truncated normal distribution with mean and precision:

$$\mu_{nr}^V = \frac{1}{\tau_{nr}^V}(-\langle\lambda_r\rangle + \sum_{m\in\Omega_n}\sum_{k=1}^{K}\phi_{mnk}\langle\tau_k\rangle(y_{mn} - \sum_{r'\neq r}\langle u_{mr'}\rangle\langle v_{nr'}\rangle)\langle u_{mr}\rangle), \tag{6.21}$$

$$\tau_{nr}^V = \sum_{m\in\Omega_n}\sum_{k=1}^{K}\phi_{mnk}\langle\tau_k\rangle\langle u_{mr}^2\rangle. \tag{6.22}$$

Here,

$$\mathbb{E}_q[(y_{mn} - \mathbf{u}_m\mathbf{v}_n^\top)^2] = (y_{mn} - \sum_{r=1}^{R}\langle u_{mr}\rangle\langle v_{nr}\rangle)^2 + \sum_{r=1}^{R}(\langle u_{mr}^2\rangle\langle v_{nr}^2\rangle - \langle u_{mr}\rangle^2\langle v_{nr}\rangle^2). \tag{6.23}$$

As for precision τ,

$$\begin{aligned}
&ln\ q(\tau_k) = \mathbb{E}_q[ln\ p(\tau_k|a_0, b_0)] + const\\
&+ \prod_{m=1}^{M}\prod_{n=1}^{N} q(z_{mn} = k)\mathbb{E}_q[ln\ p(y_{mn}|\boldsymbol{U}, \boldsymbol{V}, z_{mn} = k, \boldsymbol{\tau})]\\
&= (a_0 + \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N} q(z_{mn} = k)ln\ \tau_k - (b_0 + \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N} q(z_{mn} = k)\mathbb{E}_q[(y_{mn} - \mathbf{u}_{m\cdot}\mathbf{v}_{n\cdot}^\top)^2])\tau_k
\end{aligned}$$

$$+const. \tag{6.24}$$

Obviously, it is still a Gamma distribution.

$$q(\tau_k) = G(\tau_k|\rho_{k,1}, \rho_{k,2}), \tag{6.25}$$

where

$$\begin{aligned}\rho_{k,1} &= a_0 + \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N}\phi_{mnk}, \\ \rho_{k,2} &= b_0 + \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N}\phi_{mnk}\mathbb{E}_q[(y_{mn} - \mathbf{u}_{m\cdot}\mathbf{v}_{n\cdot}^{\top})^2].\end{aligned} \tag{6.26}$$

where $\mathbb{E}_q[(y_{mn} - \mathbf{u}_{m\cdot}\mathbf{v}_{n\cdot}^{\top})^2]$ has already been indicated in Eq. 6.23.

After updating parameters $\boldsymbol{\rho}$, the weight coefficients $\boldsymbol{\theta}$ of the K clusters are also updated. A θ_k smaller than a pre-defined threshold means that the probability of some entries to be represented by this cluster is very rare. So that those clusters can be pruned, and the noise is represented by a limited number of Gaussians.

Finally, the rate parameter of the exponential prior related to factor matrices $\mathbf{U}$ and $\mathbf{V}$ can be updated as:

$$\begin{aligned}q(\lambda_r) &\propto exp\{\mathbb{E}_q(\sum_{m=1}^{M} log\ p(u_{mr}|\lambda_r)) + \mathbb{E}_q(\sum_{n=1}^{N} log\ p(v_{nr}|\lambda_r)) + \mathbb{E}_q(log\ p(\lambda_r|a_1, b_1))\} \\ &\propto exp\{(M + N + a_1 - 1)log\lambda_r - (\sum_{m=1}^{M}\langle u_{mr}\rangle + \sum_{n=1}^{N}\langle v_{nr}\rangle + b_1)\lambda_r\}.\end{aligned} \tag{6.27}$$

Obviously, λ_r obeys a Gamma distribution

$$q(\lambda_r) = \mathcal{G}(\lambda_r|a_r^*, b_r^*) \tag{6.28}$$

where

$$a_r^* = a_1 + M + N, \tag{6.29}$$

and

$$b_r^* = b_1 + \sum_{m=1}^{M}\langle u_{mr}\rangle + \sum_{n=1}^{N}\langle v_{nr}\rangle. \tag{6.30}$$

The parameters can be updated in turn while holding others fixed. By the properties of variational Bayesian analysis, convergence to a local minimum of the DPNMF algorithm can be guaranteed after a suitable number of iterations [129].

6.3 Results

In this Section, the proposed DPNMF model is compared empirically with several state-of-the-art methods including MUNMF in [104], a sparseness-constrained NMF (SCNMF) [29], a Bayesian NMF (PSNMF) [120], and the outlier-robust MahNMF [162]. MUNMF is the standard NMF with multiplicative update rules to minimize KL divergence. SCNMF is a sparseness constrained NMF method with a Gaussian noise distribution. We set the sparseness level to 0.5 across activation coefficients so as to be consistent with the neural sparse coding scheme [163]. MahNMF minimizes the Manhattan distance between $\mathbf{Y}$ and $\mathbf{U}\mathbf{V}^\top$ for modelling the heavy tailed Laplacian noise. PSNMF is a variational Bayesian spare NMF model which assumes the column-wise sparseness of two latent matrices with Gaussian noise distribution. The equivalent graphic representation of these models is shown in Fig 6.2.

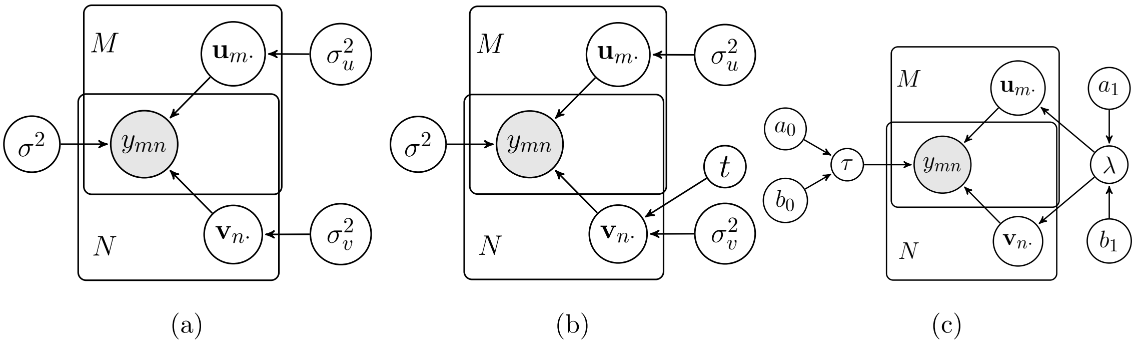


Figure 6.2 : The equivalent graphic representation of (a) MUNMF and MahNMF, (b) SCNMF, and (c) PSNMF.

6.3.1 Results on synthetic data

The performance of all the above NMF methods are compared using synthetic data sets firstly. The two latent matrices are generated from unit mean exponential distributions with three different ranks r = 5, 10 and 15. The ground-truth $\mathbf{Y}_0$ is the product of $\mathbf{U} \in \mathbb{R}_+^{500\times r}$ and $\mathbf{V} \in \mathbb{R}_+^{500\times r}$. Three types of noise, i.e., Gaussian, sparse, and mixed, are considered in the simulation. Details of the noise parameters are shown in Table 6.1. The initial number of Gaussian components K is set to 300 for DPNMF, which is large enough to represent various types of noise.

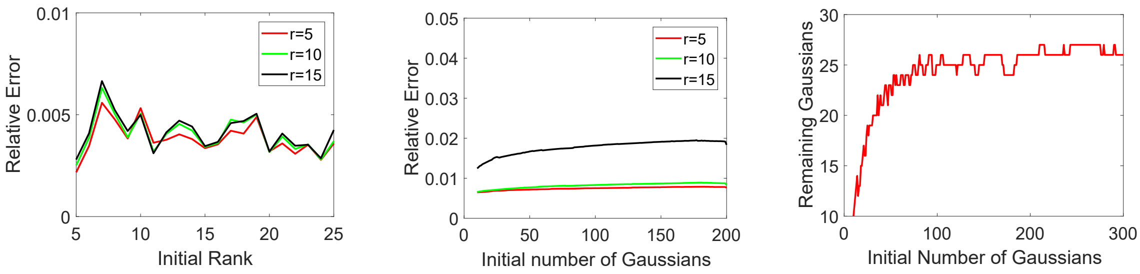
Although some methods for comparison are incapable of inferring the rank automatically, the ground-truth rank r is set as a known model input parameter. The relative error of the Frobenius norm with respect to the ground truth, defined by

$$Error = \frac{|\mathbf{Y}_0 - \overline{\mathbf{U}}\overline{\mathbf{V}}^\top|_F}{|\mathbf{Y}_0|_F}, \tag{6.31}$$

is used as the metric to quantify the performance of each algorithm, where $\overline{\mathbf{U}}$ and $\overline{\mathbf{V}}$ are recovered latent matrices. For each noise setting, the algorithms are run 20 times with different random input of $\mathbf{Y}_0$. The average relative errors are shown in Table 6.2.

One can find from Table 6.2 that DPNMF yields the smallest relative error under most cases of the three low-rank settings and three noise settings. For the Gaussian noise, PSNMF achieves comparable performance as DPNMF. However, the relative errors for the three other methods are much higher than DPNMF and PSNMF, even based on the correct rank input, which is difficult to estimate in practical applications. Besides, it is shown that the relative error of these methods increase significantly when the initial rank deviates from the ground-truth rank. In the case of sparse noise, it is evident that the performance of MUNMF and MahNMF methods degrade. This is reasonable since only Gaussian noise is considered in these two models. The DPNMF algorithm performs better than PSNMF over rank

5, although it is slightly inferior to DPNMF over rank 10 and 15. Finally, for the mixed noise, it is not surprising that it outperforms all the remaining methods with significantly smaller amount of errors. When the noise type is simple, PSNMF achieves comparable and even slightly better results than DPNMF. However, the performance of PSNMF is significantly inferior to DPNMF when the noise becomes more complicated. This is attributed to the superior capability of DPNMF to fit unknown complex noise as well as to automatically tune the rank even without prior knowledge about the exact rank.



(a) (b) (c)

Figure 6.3 : Effect of initial number of Gaussians and rank on the performance of DPNMF.

These experiments have demonstrated the superiority of DPNMF over competitive methods to recover the latent matrices under mixed noise contamination. The stability of DPNMF is empirically evaluated in terms of the varying initial number of Gaussians K and rank r. Fig 6.3(a) shows the average relative error in terms of the different initial rank for r = 5, 10, and 15, respectively. For each case, the average relative error has tiny fluctuation when the initial rank varies. However, with the increasing magnitude of the initial rank, the relative error remains stable at small values. Fig 6.3(b) shows the effect of the initial number of Gaussians on the average relative error for r = 5, 10, 15, respectively. Similar to the effect of fixing K in Fig 6.3(a) illustrates the average relative error is almost flat for the initial number of Gaussians varying from 10 to 300. Fig 6.3(c) shows the remaining number of Gaussians versus the initial number of Gaussians. For the case of mixed noise in Table 6.1, the required number of Gaussians stays stable around 25 for rank $r = 5$. From the plots in Fig 6.3, it is clear that DPNMF is quite robust to the initialization and input parameters.

Table 6.1 : Noise parameter setting for the synthetic data sets. U denotes the uniformly distributed noise followed by its range.

	$\mathcal{N}(0, 0.5^2)$	$\mathcal{N}(0, 0.1^2)$	U[-5, 5]
Gaussian Noise	100%	0	0
Sparse Noise	0	0	30%
Mixture Noise	60%	20%	20%

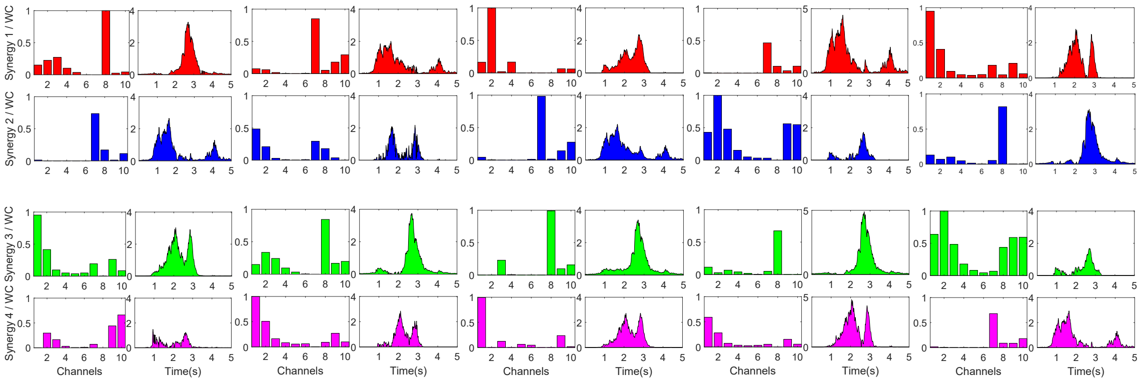
Table 6.2 : The average relative error of five algorithms under three types of noise with three different initial ranks. Best results are shown in bold .

Noise	r	DPNMF	MUNMF	SCNMF	MahNMF	PSNMF
Gaussian	5	**0.0110**	0.0350	0.6230	0.0508	0.0111
	10	**0.0087**	0.0774	0.4583	0.0661	0.0088
	15	**0.0075**	0.0945	0.2570	0.0747	**0.0075**
Sparse	5	**0.0212**	0.0535	0.6232	0.0572	0.0350
	10	0.0468	0.0815	0.4585	0.1348	**0.0275**
	15	0.0330	0.0975	0.2542	0.1766	**0.0233**
Mixture	5	**0.0079**	0.0551	0.6229	0.0929	0.0449
	10	**0.0083**	0.0856	0.4578	0.1623	0.0355
	15	**0.0154**	0.0965	0.2585	0.2004	0.0300

6.3.2 Extraction of muscle synergies

In neuroscience, it is supposed that the central neural system controls muscle synergies, or groups of co-activated muscles, rather than individual muscles, to organize any simple or complex actions and movements. Muscle synergies, i.e., the nonnegative factor vectors of $\mathbf{U}$, extracted from multichannel EMG signals using NMF have been widely applied in human-machine interfaces, prosthetic controls, neural system disease diagnoses, and stroke rehabilitation. As reviewed in the Introduction, EMG is a typical signal contaminated by multiple types of noise. Here the performance of DPNMF with competitive methods to extract synergies is compared. Since the ground truth synergies are not available, Following the study in [164], this study investigate the classification accuracy of synergies extracted by DPNMF to recognize six wrist motions. The Ninapro first dataset [165] which consists of EMG recordings for wrist, hand and finger movements is utilized. Each movement/task has 10 repetitions from 27 healthy subjects. To this end, the Ninapro real dataset is divided into training and testing sets with 60% (6 repetitions of each task) of the data assigned to training for each subject. For each factorization technique, synergies are estimated from training repetitions for each task. Those synergies are used to train support vector machines (SVM) to classify six movements, i.e., wrist flexion and extension, wrist radial and ulnar deviation, and wrist supination and pronation. DPNMF automatically selects the number of synergies, i.e. the rank r, while four is assigned as the best number of synergies for other methods based on previous recommendations [164]. The other four repetitions of each task are used to test those classifiers. The training and test samples are selected 10 times randomly to obtain the average accuracy.

Fig 6.4 shows the first two synergies and the corresponding weight coefficients of a representative wrist extension movement extracted via five NMF models. Due to space restriction, other synergies are listed in Supplementary Material. From Fig 6.4, the proposed method indicates that the primary driver of these movements comes from channels 7 and 8, i.e., front arm. This is more consistent than other methods with the physiological origin of muscles involved in wrist extension [165]. Fig 6.5 is



(a) DPNMF (b) MUNMF (c) SCNMF (d) MahNMF (e) PSNMF

Figure 6.4 : Four muscle synergies (odd columns) and the weight coefficients (WC) (even columns) extracted via the five NMF models for a representative EMG recording of wrist extension movement.

the violin graph of the classification accuracy for five methods with DPNMF yielding 82.4% accuracy. The average accuracy of all remaining methods is inferior to at least 10% to that of DPNMF.

In this example, the performance of DPNMF is investigated over a motor imagery EEG classification problem in the brain-machine interface, which heavily relies on the nonnegative matrix factorization technique. A BCI competition data set provided by the Department of Medical Informatics, Institute for Biomedical Engineering, Graz University of Technology, Austria is used [166]. This data set consists of 140 labeled trials for training and 140 unlabeled trials for the test with the subjects performing left/right imagery hand movement. Each trial has a duration of 9 seconds with the first 3 seconds as preparation period. Therefore only the remaining 6 seconds of the EEG is analyzed to perform the imagination task. The criterion is to dynamically identify the actual class as soon as possible with higher probability rather than to provide a class label using the entire data segment. Following up the previous recommendation [166], only signals in channels C_3 and C_4 are used since

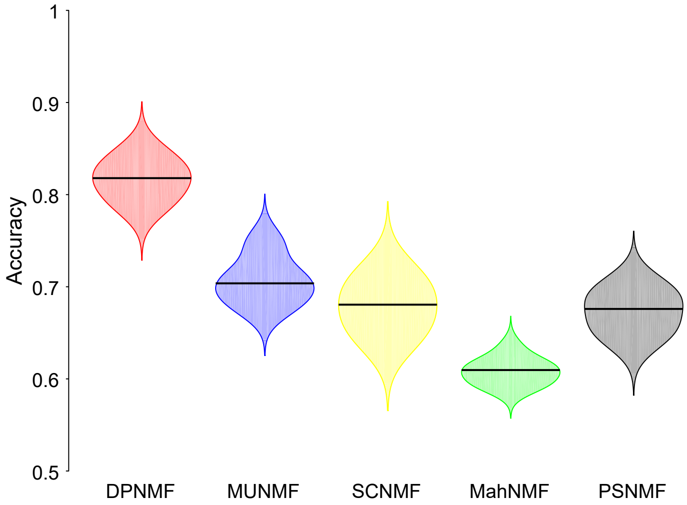


Figure 6.5 : Violin graph for the accuracy of the full synergy to classify six wrist movements for five NMF models. The black line represents the average accuracy.

channel C_z contains little information for discriminant analysis.

6.3.3 Classification of motor imagery EEG

EEG signal is firstly decomposed into wavelet coefficients using complex Morlet wavelet over the frequency range $[4, \cdots, 30]$ Hz. Fig 6.6 shows the contours of wavelet coefficients for representative left and right imagery EEGs. One can find the different μ rhythm (8-12 Hz) and β rhythm (18-25 Hz) during two different movements. For simplicity, most previous studies assumed that the noise distribution of wavelet coefficients subjects to the same Gaussian distribution as in time domain [167]. The training sampling matrix $\mathbf{Y} \in \mathbb{R}^{54 \times TP}$ is factorized by DPNMF and four other competitive methods to generate the factor matrices, which are specifically termed as basis vectors $\mathbf{U}$ and encoding variable matrix $\mathbf{V}$ in BCI, where T represents trials and P data points of EEG.

For the test sample $\tilde{\mathbf{Y}}$, its encoding variable matrix $\tilde{\mathbf{V}}$ is recovered by the product of $pi(\mathbf{U})$ and $\tilde{\mathbf{Y}}$, where $pi(\mathbf{U})$ is the pseudo-inverse of $\mathbf{U}$. With the encoding variable matrices $\mathbf{V}$ and $\tilde{\mathbf{V}}$, the decision can be made to have the class label at a single point

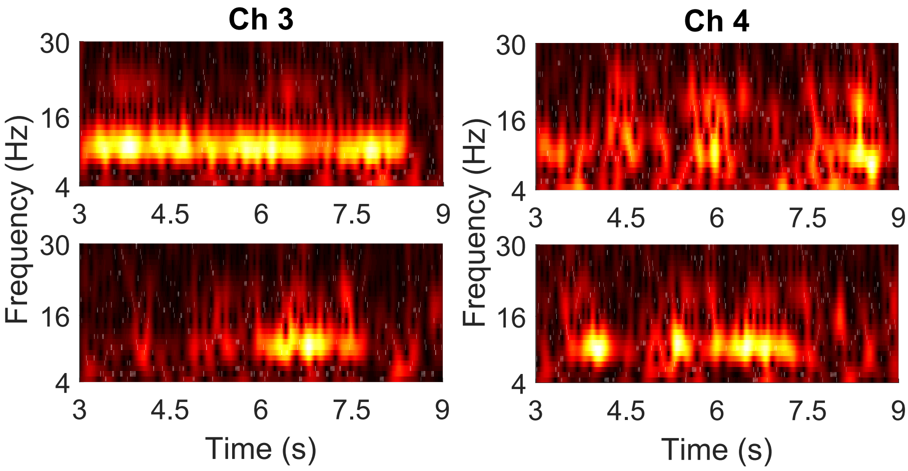
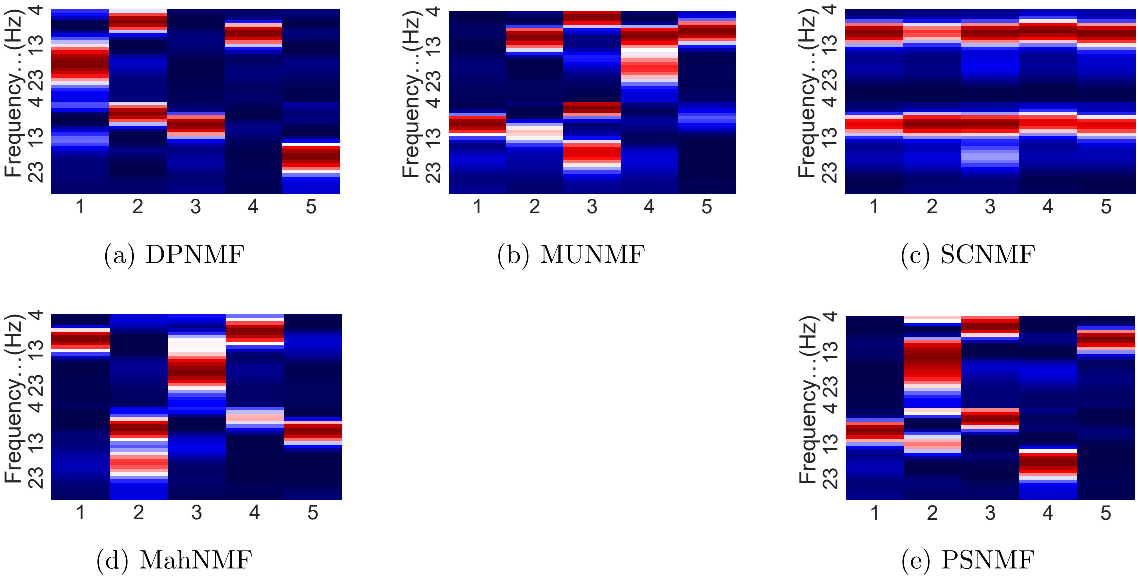


Figure 6.6 : The contours of wavelet coefficients for a representative left imagery (upper panels) and a right imagery (bottom panels) EEGs.



(a) DPNMF

(b) MUNMF

(c) SCNMF

(d) MahNMF

(e) PSNMF

Figure 6.7 : Basis vectors extracted from imagery movement two-channel EEG training samples using five NMF models.

in time from the maximal posterior probability. Readers are referred to [168] for the details of this online Bayesian classifier. Fig 6.7 shows the basis vectors of training samples obtained by five NMF models. All basis vectors reveal some useful characteristics, for example, μ rhythm, β rhythm, and sensori-motor rhythm (12-16 Hz). Fig 6.8 indicates the time course of five methods to continuously classify the single-trial imagery movements on test data. Five methods have a similar time course profile of classification accuracy. The accuracy is much lower at the beginning of the imagery and suddenly rises to a local maximum at around 3.2 s. However, the accuracy of DPNMF is superior to others with a 2-3% increment. Then, following the subject continuously performing the mental task, the accuracy of all methods slightly decreases and suddenly rises again. DPNMF reaches the highest accuracy of 83% at 3.8 s while others also rise with the maximum lower than 81% at almost the same time. It can be concluded from this figure that DPNMF performs much better than others to extract discriminative features for this imagery movement classification problem.

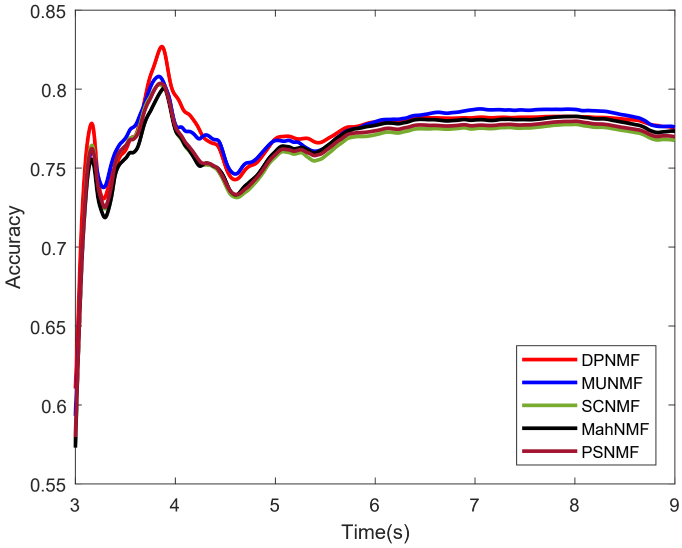


Figure 6.8 : Time course of the classification accuracy using the encoding variable matrix extracted from five NMF models.

6.4 Discussion and Conclusion

A new NMF method is proposed by using a Dirichlet process to model noise as a mixed Gaussian distribution under the variational Bayesian framework. To make variational inference feasible, a stick-breaking representation of the Dirichlet process and a factorization assumption for the posterior distribution are used. In addition, the order of latent matrices is automatically pruned by placing an ARD prior. Compared with the existing NMF methods, which assume a certain noise distribution (e.g., Gaussian or sparse noise) on data, DPNMF can extract the latent factor matrices under more complex noise distributions. The effectiveness of DPNMF has been demonstrated by synthetic data with artificial noise and by muscle synergies extraction and imagery movement classification problems with real noise. The proposed DPNMF model yields much better performance result over existing methods with respect to its capability to accurately extract the latent structure and elaborately model the multimodal noise configuration from observed signals. Although this chapter focuses on two applications in biomedical engineering and neuroscience, DPNMF is expected to have a broader range of applications in signal processing and machine learning.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This thesis had presented works on nonparametric Bayesian models and their applications on signal processing, including image denoising, inpainting and biomedical signal processing. In the following, the key results and findings of this thesis are summarised as follows.

(1) A hybrid nonlocal image blind denoising framework is proposed which exploits both Bayesian low-rank approximation and Stein's unbiased risk estimation. A variational Bayesian model is utilized to approximate the low-rank structure of the patch matrix, which simultaneously performs the noise removal and noise variance estimation. The full-rank Stein's unbiased risk estimator and its divergence formulas are modified for use in reduced-rank singular value thresholding. This modified SSVT algorithm directly maximizes the PSNR by refining the optimal threshold that minimizes the MSE estimation of rank-reduced eigen-triplets. The modified SURE model is applied on the rank-reduced eigen-triplets to enhance the initial low-rank approximation and to produce a more precise estimate of the original image.

(2) A generative model is presented for kernelized sparse Bayesian matrix factorization (KSBMF). The proposed formulation implicitly estimates the rank of the matrix without requiring the prior knowledge on the rank of the matrix, which frees the user from extensive parameter-tuning and groundless attempts. In addition, KSBMF simultaneously achieves low-rankness through

sparse Bayesian learning and sparsity through an 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. Based on the model, two algorithms are presented which incorporate the patch similarity-based kernel into the generic KSBMF model for enhanced image denoising and inpainting.

(3) To deal with complex noise in real scenarios, a hierarchical Dirichlet process nonnegative matrix factorization model is proposed. The model takes advantage of the GMM as a universal approximator to fit various types of noise rather than a single noise kernel in existing NMF models. Dirichlet process is employed to determine the number of Gaussians needed, instead of doing heuristic pruning or trying ungrounded guesses. The model is formulated into the variational Bayesian update rules instead of the usual multiplicative updating rules for NMF. The proposed model is demonstrated to significantly improve the performance of two real-world problems, i.e., muscle synergies extraction and movement imagery EEG classification, which heavily rely on the NMF technique.

7.2 Future Work

To better handle the noise in the signal, the future research can be conducted in but not limited to the following aspects.

(1) The proposed image denoising models can naturally extend to the models to remove Poisson, Gamma, Rician as well as hybrid noise in the image for real-world applications. The combination of the current model with deep neural networks to form a hybrid deep Bayesian learning scheme for improved image denoising, deblurring, and completion is also worth to investigate.

(2) A number of NMF algorithms have been extended to nonnegative tensor decomposition. To leverage the capacity of DP to model complex noise and tensor to represent multiway data together, it is worth to try to develop DP-based Bayesian nonnegative tensor decomposition model, which is expected to improve the performance of tensor signal processing tasks, for example, the tensor EEG involving both temporal and trial coordinates or the tensor EMG in a temporal-spatial-spectral domain.

Appendix

Von Mises-Fisher distribution

For a matrix random variable $\boldsymbol{D} \in \mathbb{R}^{p\times q}$ with restriction $p \geq q$ and $\boldsymbol{D'D} = \boldsymbol{I}_q$, the von Mises-Fisher distribution of $\boldsymbol{D}$ is given by

$$f(\boldsymbol{D}|\boldsymbol{F}) = \text{vMF}(\boldsymbol{F}) = \frac{1}{\kappa(p, \boldsymbol{F'F})} exp(tr(\boldsymbol{F'D})), \tag{7.1}$$

$$\kappa(p, \boldsymbol{FF'}) = {}_0F_1(\frac{1}{2}p, \frac{1}{4}\boldsymbol{F'F})C(p,q), \tag{7.2}$$

where $\boldsymbol{F} \in \mathbb{R}^{p\times q}$ is a matrix parameter of the same dimensions as $\boldsymbol{D}$ and $\kappa(p, \boldsymbol{F'F})$ is the normalizing constant. ${}_0F_1(\cdot)$ denotes a hypergeometric function of matrix argument $\boldsymbol{F'F}$. $C(p,q)$ denotes the area of the relevant Stiefel manifold $\mathfrak{F}$.

Truncated normal distribution

The probability density function of the truncated normal distribution $f(x)$ for $x \in (a,b)$ is given by

$$f(x|\mu,\sigma,a,b) = \frac{\sqrt{2}exp(-(1/2)((x-\mu)/\sigma)^2)}{\sigma\sqrt{\pi}(\text{erf}(\beta) - \text{erf}(\alpha))}\chi((a,b]), \tag{7.3}$$

where $\alpha = (a-\mu)/\sigma\sqrt{2}$, $\beta = (b-\mu)/\sigma\sqrt{2}$. The first two moments of Eq. (7.3) are $\widehat{x} = \mu - s\zeta(\mu,s)$ and $\widehat{x^2} = s^2 + \mu\widehat{x} - s\rho(\mu,s)$, which depend on the auxiliary functions

$$\zeta(\mu,\sigma) = \frac{\sqrt{2}[\exp(-\beta^2) - \exp(-\alpha^2)]}{\sqrt{\pi}(\text{erf}(\beta) - \text{erf}(\alpha))}, \tag{7.4}$$

$$\rho(\mu,\sigma) = \frac{\sqrt{2}[b\text{exp}(-\beta^2) - a\text{exp}(-\alpha^2)]}{\sqrt{\pi}(erf(\beta) - erf(\alpha))}. \tag{7.5}$$

Here erf(x) denotes the error function.

Gamma distribution

The probability density function of the gamma distribution with shape parameter α and rate parameter β is denoted as

$$f(x|a,b) = \frac{\beta^{\alpha} x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)} \tag{7.6}$$

where $x > 0$ and $\alpha, \beta > 0$. $\Gamma(\cdot)$ is the gamma function. The first moment of Eq. (7.6) is $\hat{x} = \alpha/\beta$.

Bibliography

[1] P. Orbanz and Y. W. Teh, “Bayesian nonparametric models.” *Encyclopedia of machine learning*, no. 1, 2010.

[2] N. Sengupta, M. Sahidullah, and G. Saha, “Lung sound classification using cepstral-based statistical features,” *Computers in biology and medicine*, vol. 75, pp. 118–129, 2016.

[3] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing.* Pearson Education, 2014.

[4] A. Buades, B. Coll, and J. M. Morel, “On image denoising methods,” *CMLA Preprint*, vol. 5, 2004.

[5] 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.

[6] M. L. Uss, B. Vozel, V. V. Lukin, and K. Chehdi, “Local signal-dependent noise variance estimation from hyperspectral textural images,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 469–486, 2011.

[7] S. Pyatykh, J. Hesser, and L. Zheng, “Image noise level estimation by principal component analysis,” *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 687–699, 2013.

[8] X. Fu, K. Huang, N. D. Sidiropoulos, and W.-K. Ma, “Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and

applications," *IEEE Signal Processing Magazine*, vol. 36, pp. 59–80, 2019.

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

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

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.

[25] K. Huang and N. D. Sidiropoulos, “Putting nonnegative matrix factorization to the test: A tutorial derivation of pertinent Cramer–Rao bounds and performance benchmarking,” *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 76–86, 2014.

[26] Y.-X. Wang and Y.-J. Zhang, “Nonnegative matrix factorization: A comprehensive review,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1336–1353, 2013.

[27] N. Gillis, “The why and how of nonnegative matrix factorization,” *Regularization, Optimization, Kernels, and Support Vector Machines*, vol. 12, no. 257, 2014.

[28] G. Zhou, A. Cichocki, Q. Zhao, and S. Xie, “Nonnegative matrix and tensor factorizations: An algorithmic perspective,” *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 54–65, 2014.

[29] P. O. Hoyer, “Non-negative matrix factorization with sparseness constraints,” *Journal of Machine Learning Research*, vol. 5, no. Nov, pp. 1457–1469, 2004.

[30] A. B. Owen, P. O. Perry *et al.*, “Bi-cross-validation of the SVD and the nonnegative matrix factorization,” *The Annals of Applied Statistics*, vol. 3, no. 2, pp. 564–594, 2009.

[31] J. Josse and F. Husson, “Selecting the number of components in principal component analysis using cross-validation approximations,” *Computational Statistics & Data Analysis*, vol. 56, no. 6, pp. 1869–1879, 2012.

[32] M. Gavish and D. L. Donoho, “The optimal hard threshold for singular values is $4/\sqrt{3}$,” *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 5040–5053, 2014.

[33] Q. Guo, C. Zhang, Y. Zhang, and H. Liu, “An efficient SVD-based method for image denoising,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 5, pp. 868–880, 2016.

[34] Y. Zhang, J. Liu, M. Li, and Z. Guo, “Joint image denoising using adaptive principal component analysis and self-similarity,” *Information Sciences*, vol. 259, pp. 128–141, 2014.

[35] 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.

[36] E. J. Candes, C. A. Sing-Long, and J. D. Trzasko, “Unbiased risk estimates for singular value thresholding and spectral estimators,” *IEEE Transactions on Signal Processing*, vol. 61, no. 19, pp. 4643–4657, 2013.

[37] W. Dong, G. Shi, and X. Li, “Nonlocal image restoration with bilateral variance estimation: a low-rank approach,” *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 700–711, 2013.

[38] 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.

[39] S. F. Yeganli, H. Demirel, and R. Yu, “Noise removal from MR images via iterative regularization based on higher-order singular value decomposition,” *Signal, Image and Video Processing*, vol. 11, no. 8, pp. 1477–1484, 2017.

[40] Z. Huang, Q. Li, H. Fang, T. Zhang, and N. Sang, “Iterative weighted nuclear norm for X-ray cardiovascular angiogram image denoising,” *Signal, Image and Video Processing*, vol. 11, no. 8, pp. 1445–1452, 2017.

[41] X. M. Luo, Z. Y. Suo, Q. G. Liu, and X. F. Wang, "Efficient noise reduction for interferometric phase image via non-local non-convex low-rank regularisation," *IET Signal Processing*, vol. 10, no. 7, pp. 815–824, 2016.

[42] Y. Xie, S. Gu, Y. Liu, W. Zuo, W. Zhang, and L. Zhang, "Weighted Schatten p-norm minimization for image denoising and background subtraction," *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4842–4857, 2016.

[43] J. Josse and S. Sardy, "Adaptive shrinkage of singular values," *Statistics and Computing*, vol. 26, no. 3, pp. 715–724, 2016.

[44] M. Verbanck, J. Josse, and F. Husson, "Regularised PCA to denoise and visualise data," *Statistics and Computing*, vol. 25, no. 2, pp. 471–486, 2015.

[45] X. Jia, X. Feng, and W. Wang, "Rank constrained nuclear norm minimization with application to image denoising," *Signal Processing*, vol. 129, pp. 1–11, 2016.

[46] M. Nejati, S. Samavi, H. Derksen, and K. Najarian, "Denoising by low-rank and sparse representations," *Journal of Visual Communication and Image Representation*, vol. 36, pp. 28–39, 2016.

[47] W. He, H. Zhang, L. Zhang, and H. Shen, "Hyperspectral image denoising via noise-adjusted iterative low-rank matrix approximation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 3050–3061, 2015.

[48] C. Zhang, W. Hu, T. Jin, and Z. Mei, "Nonlocal image denoising via adaptive tensor nuclear norm minimization," *Neural Computing and Applications*, pp. 1–17, 2015.

[49] X. Liu, X.-Y. Jing, G. Tang, F. Wu, and Q. Ge, "Image denoising using

weighted nuclear norm minimization with multiple strategies," *Signal Processing*, vol. 135, pp. 239–252, 2017.

[50] Z. Wu, Q. Wang, J. Jin, and Y. Shen, "Structure tensor total variation-regularized weighted nuclear norm minimization for hyperspectral image mixed denoising," *Signal Processing*, vol. 131, pp. 202–219, 2017.

[51] P. D. Hoff, "Model averaging and dimension selection for the singular value decomposition," *Journal of the American Statistical Association*, vol. 102, no. 478, pp. 674–685, 2007.

[52] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[53] M. O. Ulfarsson and V. Solo, "Dimension estimation in noisy PCA with SURE and random matrix theory," *IEEE Transactions on Signal Processing*, vol. 56, no. 12, pp. 5804–5816, 2008.

[54] ——, "Selecting the number of principal components with SURE," *IEEE Signal Processing Letters*, vol. 22, no. 2, pp. 239–243, 2015.

[55] N. R. Hansen, "On Steins unbiased risk estimate for reduced rank estimators," *Statistics & Probability Letters*, vol. 135, pp. 76–82, 2018.

[56] S. Ramani, Z. Liu, J. Rosen, J.-F. Nielsen, and J. A. Fessler, "Regularization parameter selection for nonlinear iterative image restoration and MRI reconstruction using GCV and SURE-based methods," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3659–3672, 2012.

[57] T. Qiu, A. Wang, N. Yu, and A. Song, "LLSURE: local linear SURE-based edge-preserving image filtering," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 80–90, 2013.

[58] V. Šmídl and A. Quinn, "On Bayesian principal component analysis," *Computational Statistics & Data Analysis*, vol. 51, no. 9, pp. 4101–4123, 2007.

[59] N. Dobigeon and J.-Y. Tourneret, "Bayesian orthogonal component analysis for sparse representation," *IEEE Transactions on Signal Processing*, vol. 58, no. 5, pp. 2675–2685, 2010.

[60] Y. J. Lim and Y. W. Teh, "Variational Bayesian approach to movie rating prediction," in *Proceedings of KDD cup and workshop*, vol. 7, 2007, pp. 15–21.

[61] A. Holbrook, A. Vandenberg-Rodes, and B. Shahbaba, "Bayesian inference on matrix manifolds for linear dimensionality reduction," *arXiv preprint arXiv:1606.04478*, 2016.

[62] W. E. Zhang, M. Tan, Q. Z. Sheng, L. Yao, and Q. Shi, "Efficient orthogonal non-negative matrix factorization over Stiefel manifold," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management.* ACM, 2016, pp. 1743–1752.

[63] F. Pompili, N. Gillis, P.-A. Absil, and F. Glineur, "Two algorithms for orthogonal nonnegative matrix factorization with application to clustering," *Neurocomputing*, vol. 141, pp. 15–25, 2014.

[64] S. Choi, "Algorithms for orthogonal nonnegative matrix factorization," *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1828–1832, 2008.

[65] D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[66] D.-G. Kim and Z. H. Shamsi, "Enhanced residual noise estimation of low rank approximation for image denoising," *Neurocomputing*, vol. 293, pp. 1–11, 2018.

[67] X. Huang, L. Chen, J. Tian, and X. Zhang, “Blind image noise level estimation using texture-based eigenvalue analysis,” *Multimedia Tools and Applications*, vol. 75, no. 5, pp. 2713–2724, 2016.

[68] X. Liu, M. Tanaka, and M. Okutomi, “Single-image noise level estimation for blind denoising,” *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5226–5237, 2013.

[69] W. Liu and W. Lin, “Additive white Gaussian noise level estimation in SVD domain for images,” *IEEE Transactions on Image Processing*, vol. 22, no. 3, pp. 872–883, 2013.

[70] D. Zoran and Y. Weiss, “Scale invariance and noise in natural images,” in *2009 IEEE 12th International Conference on Computer Vision.*

[71] C. Hage and M. Kleinsteuber, “Robust PCA and subspace tracking from incomplete observations using ℓ_0-surrogates,” *Computational Statistics*, vol. 29, no. 3, pp. 467–487, Jun 2014.

[72] H. Ji, S. Huang, Z. Shen, and Y. Xu, “Robust video restoration by joint sparse and low rank matrix approximation,” *SIAM Journal on Imaging Sciences*, vol. 4, no. 4, pp. 1122–1142, 2011.

[73] Q. Sun, S. Xiang, and J. Ye, “Robust principal component analysis via capped norms,” in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 2013, pp. 311–319.

[74] M. Tao and X. Yuan, “Recovering low-rank and sparse components of matrices from incomplete and noisy observations,” *SIAM Journal on Optimization*, vol. 21, no. 1, pp. 57–81, 2011.

[75] X. Ding, L. He, and L. Carin, “Bayesian robust principal component analysis,” *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3419–3430, 2011.

[76] C. Aicher, "A variational Bayes approach to robust principal component analysis," *REU 2013*, 2013.

[77] N. Wang and D.-Y. Yeung, "Bayesian robust matrix factorization for image and video processing," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1785–1792.

[78] Q. Zhao, D. Meng, Z. Xu, W. Zuo, and Y. Yan, "l_1-norm low-rank matrix factorization by variational Bayesian method," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 4, pp. 825–839, 2015.

[79] Q. Zhao, D. Meng, Z. Xu, W. Zuo, and L. Zhang, "Robust principal component analysis with complex noise," in *International Conference on Machine Learning*, 2014, pp. 55–63.

[80] 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.

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

[82] 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.

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

[84] M. Gönen and S. Kaski, "Kernelized Bayesian matrix factorization." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, p. 2047, 2014.

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

[86] 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.

[87] 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.

[88] 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.

[89] 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.

[90] 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.

[91] 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.

[92] 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.

[93] 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.

[94] 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.

[95] I. Porteous, A. Asuncion, and M. Welling, “Bayesian matrix factorization with side information and Dirichlet process mixtures,” in *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[96] 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.

[97] 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.

[98] 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.

[99] 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.

[100] 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.

[101] 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.

[102] 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.

[103] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, p. 788, 1999.

[104] ——, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, 2001, pp. 556–562.

[105] A. Cichocki, S. Cruces, and S.-i. Amari, "Generalized Alpha-Beta divergences and their application to robust nonnegative matrix factorization," *Entropy*, vol. 13, no. 1, pp. 134–170, 2011.

[106] B. Shen, B.-D. Liu, Q. Wang, and R. Ji, "Robust nonnegative matrix factorization via l_1 norm regularization by multiplicative updating rules," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 5282–5286.

[107] D. Kong, C. Ding, and H. Huang, "Robust nonnegative matrix factorization using $l_{2,1}$ norm," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. ACM, 2011, pp. 673–682.

[108] J. Huang, F. Nie, H. Huang, and C. Ding, "Robust manifold nonnegative matrix factorization," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 3, p. 11, 2014.

[109] L. Du, X. Li, and Y.-D. Shen, “Robust nonnegative matrix factorization via half-quadratic minimization,” in *2012 IEEE 12th International Conference on Data Mining.* IEEE, 2012, pp. 201–210.

[110] X. Shen, X. Zhang, L. Lan, Q. Liao, and Z. Luo, “Another robust NMF: Rethinking the hyperbolic tangent function and locality constraint,” *IEEE Access*, vol. 7, pp. 31 089–31 102, 2019.

[111] R. Schachtner, G. Po, A. M. Tomé, C. G. Puntonet, E. W. Lang *et al.*, “A new Bayesian approach to nonnegative matrix factorization: Uniqueness and model order selection,” *Neurocomputing*, vol. 138, pp. 142–156, 2014.

[112] A. T. Cemgil, “Bayesian inference for nonnegative matrix factorisation models,” *Computational Intelligence and Neuroscience*, vol. 2009, 2009.

[113] P. Gopalan, F. J. Ruiz, R. Ranganath, and D. Blei, “Bayesian nonparametric poisson factorization for recommendation systems,” in *Artificial Intelligence and Statistics*, 2014, pp. 275–283.

[114] S. Mirzaei, H. Van Hamme, and Y. Norouzi, “Blind audio source separation of stereo mixtures using Bayesian non-negative matrix factorization,” in *2014 22nd European Signal Processing Conference (EUSIPCO).* IEEE, 2014, pp. 621–625.

[115] Y. Bando, K. Itoyama, M. Konyo, S. Tadokoro, K. Nakadai, K. Yoshii, T. Kawahara, and H. G. Okuno, “Speech enhancement based on Bayesian low-rank and sparse decomposition of multichannel magnitude spectrograms,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 215–230, 2018.

[116] V. Renkens *et al.*, “Automatic relevance determination for nonnegative dictionary learning in the Gamma-Poisson model,” *Signal Processing*, vol. 132, pp.

121–133, 2017.

[117] J.-T. Chien and P.-K. Yang, "Bayesian factorization and learning for monaural source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 185–195, 2015.

[118] R. Schachtner, G. Poeppel, A. M. Tomé, and E. W. Lang, "A Bayesian approach to the Lee–Seung update rules for NMF," *Pattern Recognition Letters*, vol. 45, pp. 251–256, 2014.

[119] T. Brouwer, J. Frellsen, and P. Lió, "Comparative study of inference methods for Bayesian nonnegative matrix factorisation," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer, 2017, pp. 513–529.

[120] J. L. Hinrich and M. Mørup, "Probabilistic sparse non-negative matrix factorization," in *International Conference on Latent Variable Analysis and Signal Separation.* Springer, 2018, pp. 488–498.

[121] M. N. Schmidt and H. Laurberg, "Nonnegative matrix factorization with Gaussian process priors," *Computational Intelligence and Neuroscience*, vol. 2008, p. 3, 2008.

[122] T. Virtanen, A. T. Cemgil, and S. Godsill, "Bayesian extensions to non-negative matrix factorisation for audio signal modelling," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing.* IEEE, 2008, pp. 1825–1828.

[123] X. Chen, X. Xu, A. Liu, S. Lee, X. Chena, X. Zhang, M. J. McKeown, and Z. J. Wang, "Removal of muscle artifacts from the EEG: A review and recommendations," *IEEE Sensors Journal*, 2019.

[124] S. Thongpanja, A. Phinyomark, F. Quaine, Y. Laurillau, C. Limsakul, and P. Phukpattaranont, “Probability density functions of stationary surface EMG signals in noisy environments,” *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 7, pp. 1547–1557, 2016.

[125] V. Maz’ya and G. Schmidt, “On approximate approximations using Gaussian kernels,” *IMA Journal of Numerical Analysis*, vol. 16, no. 1, pp. 13–29, 1996.

[126] D. M. Titterington, A. F. Smith, and U. E. Makov, *Statistical analysis of finite mixture distributions.* Wiley,, 1985.

[127] R. A. Redner and H. F. Walker, “Mixture densities, maximum likelihood and the EM algorithm,” *SIAM review*, vol. 26, no. 2, pp. 195–239, 1984.

[128] W. Penny, J. Kilner, and F. Blankenburg, “Robust Bayesian general linear models,” *Neuroimage*, vol. 36, no. 3, pp. 661–671, 2007.

[129] D. M. Blei, M. I. Jordan *et al.*, “Variational inference for Dirichlet process mixtures,” *Bayesian Analysis*, vol. 1, no. 1, pp. 121–143, 2006.

[130] W. Shao, Z. Ge, and Z. Song, “Quality variable prediction for chemical processes based on semisupervised Dirichlet process mixture of Gaussians,” *Chemical Engineering Science*, vol. 193, pp. 394–410, 2019.

[131] J. Ren, K. Li, and C. Chen, “Supervised Dirichlet process mixtures of principal component analysis,” *Neurocomputing*, vol. 305, pp. 15–26, 2018.

[132] J. Xuan, J. Lu, G. Zhang, R. Y. D. Xu, and X. Luo, “Doubly nonparametric sparse nonnegative matrix factorization based on dependent Indian buffet processes,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1835–1849, 2018.

[133] S. Williamson, P. Orbanz, and Z. Ghahramani, “Dependent indian buffet processes,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 924–931.

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

[135] H. Jeffreys, *The Theory of Probability.* OUP Oxford, 1998.

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

[137] J. Winn and C. M. Bishop, “Variational message passing,” *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 661–694, 2005.

[138] J. W. Miskin, “Ensemble learning for independent component analysis,” in *in Advances in Independent Component Analysis.* Citeseer, 2000.

[139] Z. Ghahramani and M. J. Beal, “Variational inference for Bayesian mixtures of factor analysers,” in *Advances in Neural Information Processing Systems*, 2000, pp. 449–455.

[140] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[141] C. Khatri and K. Mardia, “The Von Mises-Fisher matrix distribution in orientation statistics,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 95–106, 1977.

[142] C. M. Stein, “Estimation of the mean of a multivariate normal distribution,” *The Annals of Statistics*, pp. 1135–1151, 1981.

[143] L. Zhang, W. Dong, D. Zhang, and G. Shi, “Two-stage image denoising by principal component analysis with local pixel grouping,” *Pattern Recognition*, vol. 43, no. 4, pp. 1531–1549, 2010.

[144] T. Dai, Z. Xu, H. Liang, K. Gu, Q. Tang, Y. Wang, W. Lu, and S.-T. Xia, “A generic denoising framework via guided principal component analysis,” *Journal of Visual Communication and Image Representation*, vol. 48, pp. 340–352, 2017.

[145] C. M. Bishop, *Pattern Recognition and Machine Learning.* springer, 2006.

[146] D. Van De Ville and M. Kocher, “Sure-based non-local means,” *IEEE Signal Processing Letters*, vol. 16, no. 11, pp. 973–976, 2009.

[147] 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.

[148] M. Hashemi and S. Beheshti, “Adaptive noise variance estimation in BayesShrink,” *IEEE Signal Processing Letters*, vol. 17, no. 1, pp. 12–15, 2010.

[149] T. Furnival, R. K. Leary, and P. A. Midgley, “Denoising time-resolved microscopy image sequences with singular value thresholding,” *Ultramicroscopy*, vol. 178, pp. 112–124, 2017.

[150] D. M. Blei, P. R. Cook, and M. Hoffman, “Bayesian nonparametric matrix factorization for recorded music,” in *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 439–446.

[151] F. Luisier, T. Blu, and M. Unser, “Image denoising in mixed Poisson–Gaussian noise,” *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 696–708, 2011.

[152] 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.

[153] 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.

[154] 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.

[155] 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.

[156] 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.

[157] 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.

[158] 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.

[159] 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.

[160] 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.

[161] A. Sobral, S. Javed, S. Ki Jung, T. Bouwmans, and E.-h. Zahzah, "Online stochastic tensor decomposition for background subtraction in multispectral video sequences," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 106–113.

[162] N. Guan, D. Tao, Z. Luo, and J. Shawe-Taylor, "Mahnmf: Manhattan non-negative matrix factorization," *ArXiv*, vol. abs/1207.3438, 2012.

[163] M. Beyeler, E. L. Rounds, K. D. Carlson, N. Dutt, and J. L. Krichmar, "Neural correlates of sparse coding and dimensionality reduction," *PLOS Computational Biology*, vol. 15, no. 6, p. e1006908, 2019.

[164] A. Ebied, E. Kinney-Lang, L. Spyrou, and J. Escudero, "Evaluation of matrix factorisation approaches for muscle synergy extraction," *Medical Engineering & Physics*, vol. 57, pp. 51–60, 2018.

[165] M. Atzori, A. Gijsberts, I. Kuzborskij, S. Elsig, A.-G. M. Hager, O. Deriaz, C. Castellini, H. Müller, and B. Caputo, "Characterization of a benchmark database for myoelectric movement classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 1, pp. 73–83, 2014.

[166] B. Blankertz, K. . Muller, G. Curio, T. M. Vaughan, G. Schalk, J. R. Wolpaw, A. Schlogl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schroder, and N. Birbaumer, "The BCI competition 2003: progress and perspectives in detection and discrimination of EEG single trials," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1044–1051, June 2004.

[167] W. Zeng, X. Fu, C. Hu, and Y. Du, “Wavelet denoising with generalized bivariate prior model,” *Multimedia Tools and Applications*, vol. 77, no. 16, pp. 20 863–20 887, 2018.

[168] S. Lemm, C. Schafer, and G. Curio, “BCI competition 2003-data set III: probabilistic modeling of sensorimotor μ rhythms for classification of imaginary hand movements,” *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1077–1080, June 2004.