Tensor RPCA by Bayesian CP Factorization with Complex Noise

Qiong Luo[1,2], Zhi Han[1]*, Xi'ai Chen[1,2], Yao Wang[3], Deyu Meng[3], Dong Liang[3], Yandong Tang[1]
[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences;
[2]University of Chinese Academy of Sciences; [3]Xi'an Jiaotong University
{luoqiong, hanzhi, chenxiai, ytang}@sia.cn, {dymeng, liangdong}@mail.xjtu.edu.cn, yao.s.wang@gmail.com

Abstract

The RPCA model has achieved good performances in various applications. However, two defects limit its effectiveness. Firstly, it is designed for dealing with data in matrix form, which fails to exploit the structure information of higher order tensor data in some pratical situations. Secondly, it adopts L_1-norm to tackle noise part which makes it only valid for sparse noise. In this paper, we propose a tensor RPCA model based on CP decomposition and model data noise by Mixture of Gaussians (MoG). The use of tensor structure to raw data allows us to make full use of the inherent structure priors, and MoG is a general approximator to any blends of consecutive distributions, which makes our approach capable of regaining the low dimensional linear subspace from a wide range of noises or their mixture. The model is solved by a new proposed algorithm inferred under a variational Bayesian framework. The superiority of our approach over the existing state-of-the-art approaches is demonstrated by extensive experiments on both of synthetic and real data.

1. Introduction

In the fields of data analysis, principal component analysis (PCA) has been a classical and prevalent tool and has extensive applications [16]. Originally, PCA aims to find the best L_2-norm low-rank approximation of a specified matrix due to its smoothness and has many fast numerical solvers [9, 24, 25, 26, 35, 41]. But L_2-norm is only suitable for Gaussian noise and too susceptible to outliers and gross noise. To increase the robustness of PCA, a series of works have been conducted in recent years [12, 17, 13, 19].

Inspired by the improvement of low-rank matrix analysis [4, 5, 30], the robust principal component analysis (RPCA) [40] has been proposed for remedying the deficiency of traditional PCA, in which, a high dimensional observation matrix is assumed to consist of a low-rank component and a sparse component. Specifically, let $Y \in R^{m\times n}$ be the observation data matrix, $X \in R^{m\times n}$ be the low-rank matrix, $E \in R^{m\times n}$ be the sparse noise matrix, and then we can describe the RPCA as the following optimization problem:

$$\min_{\mathrm{X,E}} \|X\|_* + \lambda\|E\|_1 \quad s.t. \quad Y = X + E, \tag{1}$$

where $\|X\|_* = \sum_r \sigma_r(X)$ denotes the nuclear norm of X, $\sigma_r(\mathrm{X})\,(r = 1, 2, ..., \min(m, n))$ is the r^{th} singular value of X, $\|E\|_1 = \sum_{ij} |e_{ij}|$ denotes the L_1-norm of E, and e_{ij} is the element in the i^{th} row and j^{th} column of E. It has been proved that if L and S satisfy a certain incoherence condition, the RPCA can uniquely extract X and E from Y [6]. RPCA has played an important role in handling various problems, including robust matrix recovery [40], face alignment [27], subspace segmentation [21] and so forth.

Recently, it has been noticed that more and more modern applications contain data with a higher order tensor structure, such as background extraction [7], face recognition and representation [40, 34, 38, 2], structure from motion [36], object recognition [37] and motion segmentation [39].

Matrices can be viewed as second order tensors, however, moving from matrices to higher order tensors presents significant new challenges. A direct way to address these challenges is to unfold tensors to matrices and then directly apply the matrix RPCA model. Unfortunately, as recently pointed out by [7], the multilinear structure is lost in such matricization and as a result, methods constructed based on these techniques often lead to suboptimal results. As such, it is helpful to handle such raw data by using a direct tensor representation, and several researches have been made in the literatures [11, 20].

Moreover, L_1-norm and L_2-norm can characterize specific Laplace and Gaussian distributions, respectively, but the real noise is generally not of a particular kind of noise configurations, as already shown in [42]. Mixture of Gaussians (MoG) is capable to commonly approximate wider range of distributions due to its universal approximation capability, and Laplacian and Gaussian are regarded as a special case of MoG [3]. It has been demonstrated that MoG

*Corresponding author.

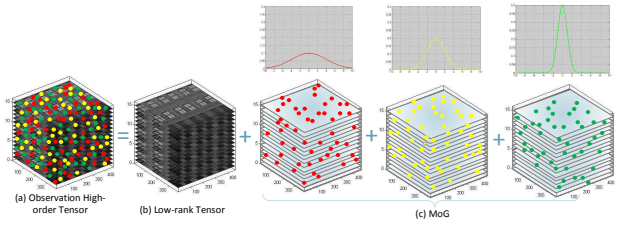

Figure 1. TenRPCA-MoG sketch map. (a) is the observation tensor, (b) is the low-rank tensor, (c) represents the complex noise.

can deal with complex noise in multiple computer vision tasks, like image denoising and recovery [23, 42, 10].

In this paper, we propose a new tensor based RPCA model with noise modeling by MoG, which is named as TenRPCA-MoG. As shown in Fig. 1, the new model divides the observation (noisy data) into a low-rank tensor component (clean data) and the residue (noise), and models them separately. It has the following contributions: firstly, it treats raw high-order data as a tensor to reserve the complete structure information, and uses CP tensor factorization method to replace existing matrix factorization method to extract low-rank structure in tensor data; secondly, it adopts MoG to model noise which makes it have the ability to fit a wide range of noises rather than Gaussian or Laplacian noise; thirdly, we formulate the problem as a generative model under the Bayesian framework and propose an algorithm based on the variational inference theory to infer the posterior and effectively solve the problem.

2. Related work

In order to make full use of high-order data structure information, Cao et al [7] extended the RPCA to the tensor form based on Tucker decomposition and successfully applied it to the background extraction. Compared with other methods, this model can achieve good extraction results at a very low sampling rate. However, it is designed to deal with only two types of noises, i.e., Gaussian noise and impulse noise, such that it is inadequate for more complex noise in real scenarios.

Meng and De la Torre [23] firstly applied the MoG to low-rank matrix factorization (LRMF) for adapting to unknown noise. Consequently, Zhao et al [42] proposed a RPCA-MoG model which used MoG to model RPCA noise. Benefiting from powerful approximation capability of MoG, they successfully applied it to the face modeling and background subtraction. However, these methods are designed based on matrix techniques and fail to take the advantage of structure prior of original data. In order to overcome such defect, Chen et al [10] developed a low-rank tensor factorization (MoG-WLRTF) model with MoG and got a good result on image denoising.

The differences and improvements of our model against Chen [10] and Cao [7] are specified as follows: Chen's work is a LRTF model, while our work is a RPCA model under the Bayesian framework, which has better adaptivity to various problems and the rank can be automatically confirmed by the algorithm itself; compared with our model, Cao's model lacks a universal noise modeling ability and is only suitable for background extraction application.

3. Notations

Throughout the paper, lowercase letters $(a, b, \cdots)$ denote scalars and bold lowercase letters denote vectors $(\mathbf{a}, \mathbf{b}, \cdots)$ with elements $(a_i, b_j, \cdots)$. Uppercase letters $(A, B, \cdots)$ denote the matrices with column vectors $(a_{:i}, b_{:j}, \cdots)$ and elements $(a_{ij}, b_{ij}, \cdots)$. High-order tensors are represented by calligraphic letters $(\mathcal{A}, \mathcal{B}, \cdots)$. A K-mode tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$ is rank-one tensor if it can be written as the out product of K vectors, i.e., $\mathcal{X} = a^1 \circ a^2 \circ \cdots \circ a^K$.

4. TenRPCA-MoG model

The traditional RPCA model is formulated as Eq. 2. Tensor RPCA has the similiar form as:

$$\min_{\mathcal{X},\mathcal{E}} \|\mathcal{X}\|_* + \lambda \|\mathcal{E}\|_1 \quad s.t. \quad \mathcal{Y} = \mathcal{X} + \mathcal{E}, \tag{2}$$

where $\mathcal{Y} \in \mathbb{R}^{f \times g \times m}$ is the observation data tensor, $\mathcal{X} \in \mathbb{R}^{f \times g \times m}$ is the low-rank tensor and $\mathcal{E} \in \mathbb{R}^{f \times g \times m}$ is the noise tensor. Here, the L_1 norm is specifically set for dealing with noise under sparse assumption, Laplacian noise, for example. However, as we introduced before, the real noise is generally much more complex rather than a simple Laplacian noise. In order to improve the robustness to complex noise of tensor RPCA, we introduce MoG for noise modeling and obtain TenRPCA-MoG as:

$$\min_{\mathcal{X},\mathcal{E}} \|\mathcal{X}\|_* + \lambda \|\mathcal{E}\|_M \quad s.t. \quad \mathcal{Y} = \mathcal{X} + \mathcal{E}, \tag{3}$$

where symbol $\|\mathcal{E}\|_M$ means that $\mathcal{E}$ is modeled with MoG. As shown in Fig. 1 as a simulation, the first and the second Gaussians are for describing dense noise, while the third Gaussian is for approximating Laplacian noise.

In our model, the low- rank factorization of tensor $\mathcal{X}$ is obtained by CP decomposition. Hence, the detailed introduction of our model starts from CP decomposition with its helpful properties.

4.1. CP decomposition

There are two general tensor decomposition frameworks, Tucker and CANDECOMP/PARAFAC (CP). CP decomposition can be considered as a higher-order generalization of the matrix singular value decomposition [8]. It decomposes a tensor into a sum of rank-one component tensors [18].

A K-mode tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$, with the integer I_k symbolizing the dimension of $\mathcal{X}$ along the K-th model, can be represented in the CP decomposition form as:

$$\mathcal{X} = \sum_{d=1}^{D} U^d \circ V^d \circ \cdots \circ T^d. \tag{4}$$

Here mark '$\circ$' denotes the vector outer product, D is regarded as the rank of the tensor $\mathcal{X}$, and U, V, T represent the matrices of the corresponding vectors in the rank-one tensors, which are called factor matrix, for example,

$$U = [u_1, u_2, \cdots u_d]. \tag{5}$$

Using the factor matrix, the CP decomposition of a third-order tensor can be written as an unfolding form:

$$\begin{aligned} \mathcal{X}_{(1)} &= U(T \odot V) \\ \mathcal{X}_{(2)} &= V(T \odot U) \\ \mathcal{X}_{(3)} &= T(V \odot U), \end{aligned} \tag{6}$$

where $\odot$ denotes the Khatri-Rao product. The single element of tensor can be written as:

$$x_{ij\cdots k} = \sum_{d=1}^{D} U_i^d V_j^d \cdots T_k^d. \tag{7}$$

In addition, by assuming $U \in \mathbb{R}^{a \times c}$ and $V \in \mathbb{R}^{b \times c}$, the Khatri-Rao product has the following property:

$$(U \odot V)^T (U \odot V) = (U^T U) * (V^T V), \tag{8}$$

where $*$ indicates the dot product.

4.2. Tensor RPCA low-rank component modeling

There are several approaches for solving CP decomposition, such as nuclear norm based method [33, 15] and probabilistic based method [32]. However, these methods are either prone to overfitting due to an inaccurate tensor rank and point estimations of latent factors or computationally expensive, because the tensor rank needs to be predefined by tuning parameter or cross-validations [43]. Variational Bayesian method also has been widely applied to CP decomposition [28, 29]. It is a commonly used approximation method which employs more global criteria and has definite solution of posterior [3]. It overcomes the aforementioned defects. In this paper, we follow the thoughts of Variational Bayesian method for solving the proposed TenRPCA-MoG model.

In Eq. 4, U is a $f \times D$ matrix, V is a $g \times D$ matrix and T is a $m \times D$ matrix. D is the rank of tensor $\mathcal{X}$. One way to model the low-rank component $\mathcal{X}$ is to apply Laplacian prior to the factor matrix. The other way is to add the beta-Bernoulli priors on the factor matrix [14]. Here we introduce the automatic relevance determination (ARD) to model the low-rank component of $\mathcal{X}$ [1], because of its high computational efficiency.

Our goal is to achieve column sparsity in U, V and T, such that most of columns in U, V and T will approach zeros, which makes the $\mathcal{X}$ low-rank. We assume that the columns of U, V and T have the following priors:

$$\begin{aligned} \mathbf{u}_{.d} &\sim \mathcal{N}(\mathbf{u}_{.d}|0, \gamma_d^{-1}\mathbf{I}_f) \\ \mathbf{v}_{.d} &\sim \mathcal{N}(\mathbf{v}_{.d}|0, \gamma_d^{-1}\mathbf{I}_g) \\ \mathbf{t}_{.d} &\sim \mathcal{N}(\mathbf{t}_{.d}|0, \gamma_d^{-1}\mathbf{I}_m), \end{aligned} \tag{9}$$

where $\mathbf{I}_m$ denotes the $m \times m$ identity matrix. Precision variable γ_d follows a conjugate prior as:

$$\gamma_d \sim Gam(\gamma_d|a_0, b_0). \tag{10}$$

Such assumptions make the columns of U, V and T have identical sparsity outline enforced by the common presisions γ_d. Such model has been demonstrated to have the ability of making the γ_d very large during the inference, thus reduces the rank estimation of $\mathcal{X}$ [1].

4.3. Tensor RPCA noise component modeling

Following the statement in Eq. 3, noise $\mathcal{E}$ is described by the MoG as:

$$e_{ijk} \sim \sum\nolimits_{n=1}^{N} \pi_n \mathcal{N}(e_{ijk}|\mu_n, \tau_n^{-1}), \tag{11}$$

where π_n is the mixing proportion with $\pi_n \geq 0$ and $\sum_{n=1}^{N} \pi_n = 1$, N is the Gaussian components number and $\mathcal{N}(e|\mu, \tau^{-1})$ denotes the Gaussian distribution with mean μ and precision τ. A latent binary random variable z_{ijkn} is introduced to express Eq. 11 as a two-level generative model via Eq. 12 and Eq. 13:

$$e_{ijk} \sim \prod\nolimits_{n=1}^{N} \mathcal{N}(e_{ijk}|\mu_n, {\tau_n}^{-1})^{z_{ijkn}}, \tag{12}$$

$$\mathbf{z}_{ijk} \sim Multinomia(\mathbf{z}_{ijk}|\boldsymbol{\pi}), \tag{13}$$

where $\mathbf{z_{ijk}} = (z_{ijk1}, \cdots, z_{ijkN}) \in \{0, 1\}^N$ and $\sum_{n=1}^{N} z_{ijkn} = 1$. The marginal distribution of $\mathcal{Z}$ abide by a multinomial distribution in terms of the mixing proportion π_n as Eq. 13, where $0 \leq \pi_n \leq 1$ and $\sum_{n=1}^{N} \pi_n = 1$. In addition, we apply conjugate priors on the mixing coefficient $\boldsymbol{\pi}$ and Gaussian parameters μ_n, τ_n.

$$\mu_n, \tau_n \sim \mathcal{N}(\mu_n|\mu_0, (\beta_0\tau_n)^{-1}) Gam(\tau_n|c_0, d_0), \tag{14}$$

$$\boldsymbol{\pi} \sim Dir(\boldsymbol{\pi}|\boldsymbol{\alpha}_0), \tag{15}$$

where $Dir(\boldsymbol{\pi}|\boldsymbol{\alpha}_0)$ is the Dirichlet distribution parameterized by $\boldsymbol{\alpha}_0 = (\alpha_{01}, \cdots, \alpha_{0n})$, and $Gam(\tau|c_0, d_0)$ denotes the Gamma distribution with c_0 and d_0.

Based on Eq. 3 and Eq. 9-15, the full Bayesian model of tensor RPCA with MoG (TenRPCA-MoG), can be constructed as:

$$p(U, V, T, Z, \mu, \tau, \pi, \gamma|\mathcal{Y}), \tag{16}$$

where $\mathcal{Z} = \{z_{ijk}\}$, $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_n)$, $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_n)$, and $\boldsymbol{\gamma} = (\gamma_1, \cdots, \gamma_d)$.

5. TenRPCA-MoG Inference Algorithm

5.1. Variational Inference

Considering the following problem: there is an observed data D and we have already known the form of the model, we aim to draw the posterior $p(l|D)$ of all the involved parameters and latent variables. Generally, the form of posterior is intractable. Variational Bayesian (VB) method deals with the problem by finding a more tractable and easier mathematical form $q(l)$ to approximate the true posterior $p(l|D)$ [3]. Naturally, we need a dissimilarity function $d(q; p)$ to measure the difference between $q(l)$ and $p(l|D)$. Hence, the inference is performed by selecting the distribution $q(l)$ and minimizing $d(q; p)$ to find the approximation distribution $q(l|D)$. In this paper, we choose the Kullback-Leibler divergence (KL-devergence) as the dissimilarity function:

$$\min_{q \in C} KL(q \parallel p) = -\int q(l) \ln \left\{ \frac{p(l|D)}{q(l)} \right\} dl, \tag{17}$$

where C denotes the set of probability densities with certain restrictions to make the minimization tractable, $KL(q \parallel p)$ denotes the KL divergence. The variational distribution $q(l)$ is usually assumed to factorize over some partition of the latent variables, known as mean-field variational Bayesian, $q(l) = \prod_i q_i(l_i)$. By minimizing the KL devergence, $q(l)$ has the closed-form solution as:

$$q_j^*(l_j) = \frac{\exp\left\{ \langle \ln p(l, D) \rangle_{l/l_j} \right\}}{\int \exp\left\{ \langle \ln p(l, D) \rangle_{l/l_j} \right\} dl_j}, \tag{18}$$

where $\langle . \rangle_{l/l_j}$ is the expectation of the logarithm of the joint probability of the data and latent variables without l_j. In this paper, the posterior of Eq. 16 will be replaced by factorized form based on mean-field variational Bayesian theory as:

$$\begin{aligned} &q(U, V, T, \mathcal{Z}, \boldsymbol{\mu}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\gamma}) = \textstyle\prod_i q(\boldsymbol{u}_{i.}) \prod_j q(\boldsymbol{v}_{j.}) \prod_k q(\boldsymbol{t}_{k.}) \\ &\textstyle\prod_{ijk} q(\boldsymbol{z}_{ijk}) \prod_n q(\mu_n, \tau_n) q(\boldsymbol{\pi}) \prod_d q(\gamma_{d.}), \end{aligned} \tag{19}$$

where $\boldsymbol{u}_{i.}$ denotes the the i-th row of U. The next section will give the approximation distribution in Eq. 19.

5.2. Low-rank component estimation

U, V, T and $\boldsymbol{\gamma}$ are updated during the low-rank component estimation. From Eq. 18, the row $\mathbf{u}_{i.}$ of U has the following inference result:

$$q(\mathbf{u}_{i.}) = \mathcal{N}(\mathbf{u}_{i.}|\mu_{\mathbf{u}_{i.}}, \textstyle\sum_{\mathbf{u}_{i.}}). \tag{20}$$

Here $\mathbf{u}_{i.}$ and $\sum_{\mathbf{u}_{i.}}$ denote the mean and covariance of the Gaussian distribution respectively. The closed form of updating is given by

$$\begin{aligned} \mu_{\mathbf{u}_{i.}}^T = &\textstyle\sum_{\mathbf{u}_{i.}} * \Big\{ \sum_n \langle \tau_n \rangle \sum_{jk} \langle z_{ijkn} \rangle (y_{ijk} - \langle \mu_n \rangle) \\ &\langle \mathbf{t}_{k.} \odot \mathbf{v}_{j.} \rangle \Big\}^T, \end{aligned}$$

$$\textstyle\sum_{\mathbf{u}_{i.}} = \Big\{ \sum_n \tau_n \sum_{jk} \langle z_{ijk} \rangle \Big\langle (\mathbf{t}_{k.} \odot \mathbf{v}_{j.})^T (\mathbf{t}_{k.} \odot \mathbf{v}_{j.}) \Big\rangle + \mathbf{\Gamma} \Big\}^{-1}.$$

Here $\mathbf{\Gamma}$ denotes $diag(\langle \boldsymbol{\gamma} \rangle)$. However, the mean of $(\mathbf{t}_{k.} \odot \mathbf{v}_{j.})^T (\mathbf{t}_{k.} \odot \mathbf{v}_{j.})$ cannot be solved directly, for which the Eq. 8 is introduced and the following result is obtained

$$\textstyle\sum_{\mathbf{u}_{i.}} = \Big\{ \sum_n \tau_n \sum_{jk} \langle z_{ijk} \rangle \big\langle \big(\mathbf{t}_{k.}{}^T \mathbf{t}_{k.}\big) * \big(\mathbf{v}_{j.}{}^T \mathbf{v}_{j.}\big) \big\rangle + \mathbf{\Gamma} \Big\}^{-1}.$$

The updates of V and T have similar froms.

$$q(\mathbf{v}_{j.}) = \mathcal{N}(\mathbf{v}_{j.}|\mu_{\mathbf{v}_{j.}}, \textstyle\sum_{\mathbf{v}_{j.}}), \tag{21}$$

$$\begin{aligned} \mu_{\mathbf{v}_{j.}}^T = &\textstyle\sum_{\mathbf{v}_{i.}} * \Big\{ \sum_n \langle \tau_n \rangle \sum_{jk} \langle z_{ijkn} \rangle (y_{ijk} - \langle \mu_n \rangle) \\ &\langle \mathbf{t}_{k.} \odot \mathbf{u}_{i.} \rangle \Big\}^T, \end{aligned}$$

$$\textstyle\sum_{\mathbf{v}_{j.}} = \Big\{ \sum_n \tau_n \sum_{ik} \langle z_{ijk} \rangle \big\langle \big(\mathbf{t}_{k.}{}^T \mathbf{t}_{k.}\big) * \big(\mathbf{u}_{i.}{}^T \mathbf{u}_{i.}\big) \big\rangle + \mathbf{\Gamma} \Big\}^{-1}.$$

$$q(\mathbf{t}_{k.}) = \mathcal{N}(\mathbf{t}_{k.}|\mu_{\mathbf{t}_{k.}}, \textstyle\sum_{\mathbf{t}_{k.}}), \tag{22}$$

$$\begin{aligned} \mu_{\mathbf{t}_{k.}}^T = &\textstyle\sum_{\mathbf{t}_{k.}} * \Big\{ \sum_n \langle \tau_n \rangle \sum_{ij} \langle z_{ijkn} \rangle (y_{ijk} - \langle \mu_n \rangle) \\ &\langle \mathbf{v}_{j.} \odot \mathbf{u}_{i.} \rangle \Big\}^T, \end{aligned}$$

$$\textstyle\sum_{\mathbf{t}_{k.}} = \Big\{ \sum_n \tau_n \sum_{ij} \langle z_{ijk} \rangle \big\langle \big(\mathbf{v}_{j.}{}^T \mathbf{v}_{j.}\big) * \big(\mathbf{u}_{i.}{}^T \mathbf{u}_{i.}\big) \big\rangle + \mathbf{\Gamma} \Big\}^{-1}.$$

The parameters $\boldsymbol{\gamma}$ has following update:

$$q(\gamma_d) = Gam(\gamma_d|a_d, b_d), \tag{23}$$

where

$$a_d = a_0 + \frac{f + g + m}{2},$$

$$b_d = b_0 + \frac{1}{2} \left(\langle \mathbf{u}_{.d}^T \mathbf{u}_{.d} \rangle + \langle \mathbf{v}_{.d}^T \mathbf{v}_{.d} \rangle + \langle \mathbf{t}_{.d}^T \mathbf{t}_{.d} \rangle \right).$$

5.3. Noise component estimation

Combining Eq. 14-15 and its conjugate characteristic, μ_n and τ_n have the following inference results:

$$q(\mu_n, \tau_n) = \mathcal{N}(m_n, (\beta_n \tau_n)^{-1}) Gam(\tau_n | c_n, d_n), \quad (24)$$

here

$$\beta_n = \beta_0 + \sum\nolimits_{ijk} \langle z_{ijkn} \rangle,$$

$$m_n = \frac{1}{\beta_n}(\beta_0 \mu_0 + \sum\nolimits_{ijk} \langle z_{ijkn} \rangle (y_{ijk} - \langle u_{i.} \rangle \langle v_{j.} \odot t_{k.} \rangle^T),$$

$$c_n = c_0 + \frac{1}{2} \sum\nolimits_{ijk} \langle z_{ijkn} \rangle,$$

$$d_n = d_0 + \frac{1}{2} \left\{ \sum\nolimits_{ijk} \langle z_{ijkn} \rangle \left\langle \left(y_{ijk} - \langle u_{i.} \rangle \langle v_{j.} \odot t_{k.} \rangle^T \right)^2 \right\rangle + \beta_0 {\mu_0}^2 - \frac{1}{\beta_n} \left(\sum\nolimits_{ijk} \langle z_{ijkn} \rangle (y_{ijk} - \langle u_{i.} \rangle \langle v_{j.} \odot t_{k.} \rangle^T) + \beta_0 \mu_0 \right)^2 \right\}.$$

The mixing coefficient also can be inferred as:

$$q(\boldsymbol{\pi}) = Dir(\boldsymbol{\pi} | \boldsymbol{\alpha}), \quad (25)$$

where

$$\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_n),$$

$$\alpha_n = \alpha_{0n} + \sum\nolimits_{ijk} \langle z_{ijkn} \rangle.$$

The latent binary random variable z_{ijkn} is inferred to have following distribution:

$$q(\boldsymbol{z}_{ijk}) = \prod\nolimits_n {r_{ijkn}}^{z_{jikn}}, \quad (26)$$

where

$$r_{ijkn} = \frac{\rho_{ijkn}}{\sum_n \rho_{ijkn}},$$

$$\ln \rho_{jikn} = \frac{1}{2} \langle \ln \tau_n \rangle - \frac{1}{2} \ln 2\pi - \frac{1}{2} \langle \tau_n \rangle \langle (y_{jik} - \mathbf{u}_{i.} (\mathbf{t}_{k.} \odot \mathbf{v}_{j.}) - \mu_n) \rangle + \langle \ln \pi_n \rangle.$$

Based on Eq. 20-26, the overall optimization process under variational Bayesian framework is shown in Algorithm 1.

Algorithm 1: Variational Bayesian algorithm for TenRPCA-MoG

Input: $\mathcal{X} \in \mathbb{R}^{f \times g \times m}$, each image with size $f \times g$, and m denotes the number of images.
Output: U, V, T by CP decomposition
1: Initialize $\mathcal{Z}, \boldsymbol{\mu}, \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\gamma}$, d, MoG number N, small threshold ε
2: **Repeat**
3: **Low-Rank update**
  Updating U via Eq. 20
  Updating V via Eq. 21
  Updating T via Eq. 22
  Updating $\boldsymbol{\gamma}$ via Eq. 23
4: **MoG update**
  Updating $\boldsymbol{\mu}$ and $\boldsymbol{\tau}$ via Eq. 24
  Updating $\boldsymbol{\pi}$ via Eq. 25
  Updating $\mathcal{Z}$ via Eq. 26
5: **Until** converge

6. Complexity

We take 3-order tensor of the size $U \times V \times T$ as an example. The rank is set as R, and N is the number of Gaussians. In our algorithm, only simple computations are involved in the variational inference of parameters, except that inferring each of u_i, v_j and t_k needs to invert an $R \times R$ matrix, leading to $\mathrm{O}((U+V+Y))R^3$ cost. The cost of the sum of z_{ijk} and τ_n in the mean in Eq. 20 is $\mathrm{O}(UVTN)$, and the rest of the mean in Eq. 20 takes no more than $\mathrm{O}(UVTR)$ cost. For the variance in Eq. 20, it requires additional no more than $\mathrm{O}((UVT + UV + VT + UT)R^2$ computational cost. In all, the complexity of our algorithm is $\mathrm{O}((U+V+T)R^3 + (UVT + UV + VT + UT)R^2 + UVTR + UVTN)$ per iteration. The cost of our method is thus linear in both data dimensionality and size.

7. Experiments

In this section, we carry out a series of experiments on synthetic data, benchmark RGB image, Columbia Multispectral Image Database[1], real hyperspectral images and video sequences. We compare the proposed TenRPCA-MoG with some competitive models including: VBRPCA [1], RegL1ALM [44], PARAFAC [22], MoG-RPCA [42], MoG-LRMF [23], LRTA [31], MoG-WLRTF [10].

7.1. Parameter settings

We adopt a non-informative way to deal with hyperparameters of TenRPCA-MoG, which can reduce the impact on the posterior distributions [3]. In the following experiments, we set μ_0=0, and $\alpha_{01}, \cdots, \alpha_{0N}$, β_0, a_0, b_0, c_0, d_0 as 10^{-6}. For the number of Gaussian component, we just empirically set N=3 throughout all our experiments.

7.2. Synthetic Experiments

The test synthetic tensor is produced as follows: 1) we randomly generate three matrices denoting as U, V, T with size 100×5, 100×5, 10×5, respectively, where each column vector of the matrices follows the standard normal distribution $\mathcal{N}(0, 1)$; 2) we utilize inner product and Khatri-Rao product as Eq. 6 to get the unfolding form of ground truth tensor with the size of 100×100×10 and rank D = 5. 3) we add different types of noises to the ground truth tensor. The types of noises are specified as follows: (1) Sparse noise: 10% elements are corrupted by the uniform noise between [-25,25]; (2) Gaussian noise: all elements are corrupted with Gaussian noise $\mathcal{N}(0, 0.01)$; (3) Mixture noise: 10% of elements mix with uniform noise within [-25,25], 20% of elements mix with Gaussian noise $\mathcal{N}(0, 1)$

[1] http://www1.cs.columbia.edu/CAVE/databases/multispectral

Table 1. Reconstruction performance of diffirent methods with diffirent noises.

	VBRPCA	RegL1-ALM	PARAFAC	MoG-RPCA	MoG-LRMF	LRTA	MoG-WLRTF	TenRPCA-MoG
Gaussian Noise								
RRE	0.2483	0.1772	0.0337	0.1006	0.1643	**0.0222**	0.0547	0.0261
MSE	0.01329	0.00727	0.00027	0.00263	0.00713	**0.00011**	0.00084	0.00020
PSNR	18.76	21.37	35.72	25.78	21.46	**39.57**	31.29	38.06
SSIM	0.2042	0.7080	0.9624	0.7977	0.6714	**0.9809**	0.8594	0.9563
Sparse Noise								
RRE	**0.0204**	6.2697	1.9773	0.4458	4.2433	0.6089	0.0532	0.0419
MSE	**0.00010**	10.681	0.89897	0.05554	4.3628	0.10559	0.000728	0.000727
PSNR	**39.88**	-10.28	0.46	12.55	-6.39	9.76	31.77	36.31
SSIM	**0.9943**	0.0027	0.0016	0.8062	0.5822	0.0386	0.8604	0.9375
Mixture Noise								
RRE	0.2088	6.4358	2.392	0.1340	2.6342	0.6509	0.0642	**0.0634**
MSE	0.01	10.6232	1.0739	0.0049	1.7949	0.1060	0.00109	**0.00107**
PSNR	19.84	-10.26	-0.30	23.02	-2.54	9.74	29.85	**30.44**
SSIM	0.1950	0.0030	0.0018	0.5853	0.1520	0.0718	0.7911	**0.7955**

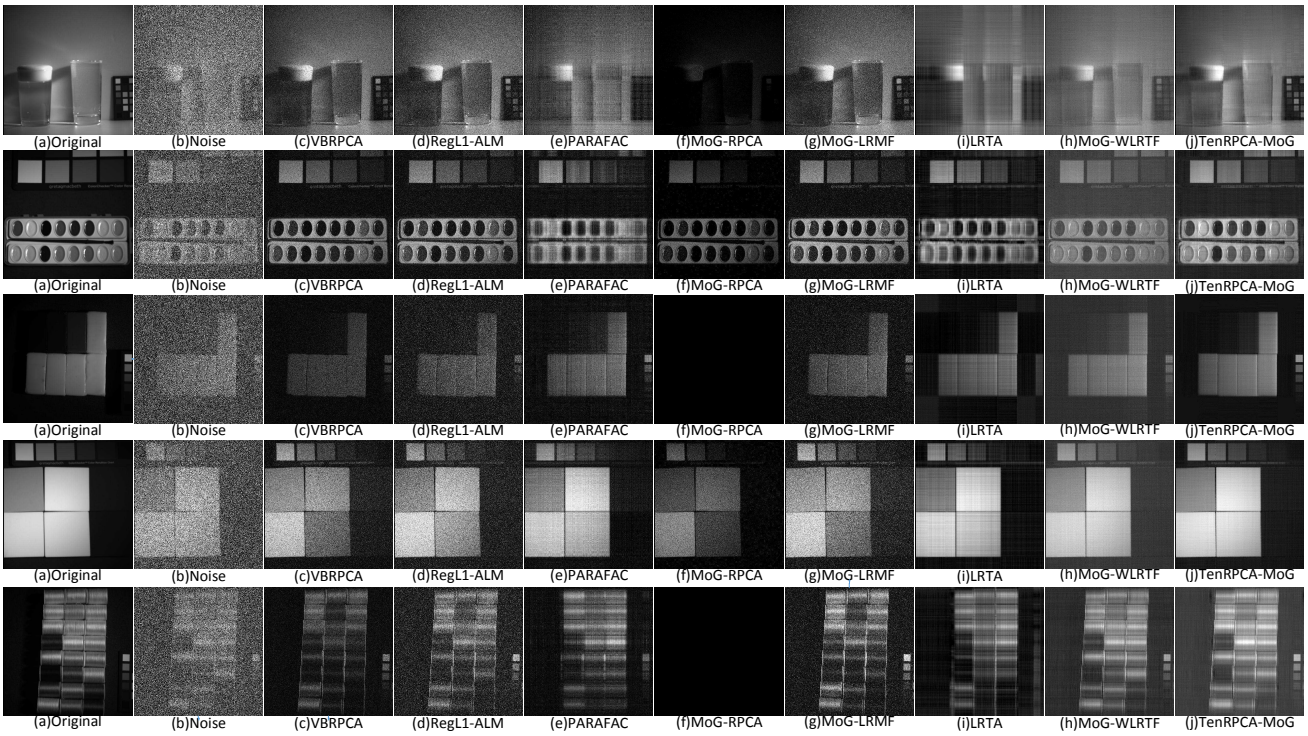

Figure 2. The 31st band of multispectral images. (a) Original image; (b) Noisy image; (c)-(j) Recovered image

and 70% of elements mix with Gaussian noise $\mathcal{N}(0, 0.01)$. The results are derived from the average of 10 trials.

We adopt four criteria to quantitatively evaluate the performances. (1) Relative reconstruction error (RRE): $\|X - M\|_F / \|M\|_F$, where M represents the ground truth tensor and X represents the reconstructed low-rank tensor; (2) Mean squared error (MSE); (3) Peak signal-to-noise ratio (PSNR); (4) Structural similarity (SSIM). The smaller RRE, MSE values and larger PSNR, SSIM values imply a better denoising effect. The results are listed in Table 1.

We bold the optimal values and underline the suboptimal values. For the sparse noise case, our approach is superior to other competitive methods, except VBRPCA which is specifically designed for sparse noise. For the Gaussian noise case, all the best results are from tensor based methods (LRTA and our method). In the experiment of mix-

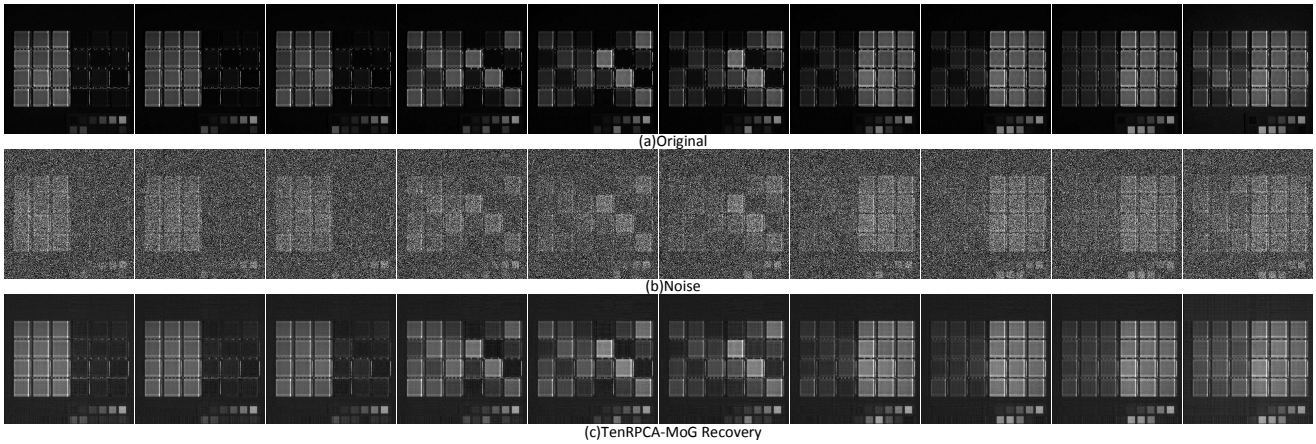

Figure 3. Ten randomly selected bands from glass tiles multispectral images. (a) Original image; (b) Noisy image; (c) Recovered image

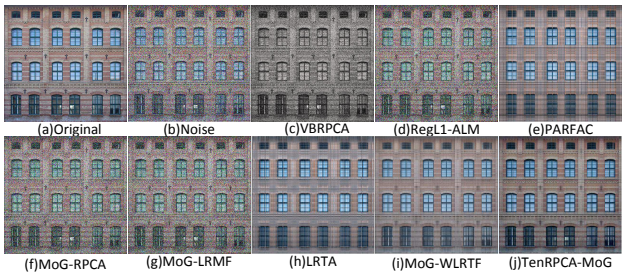

Figure 4. Smaller mixture noise for Facade. (a) Original image; (b) Noisy image; (c)-(j) Recovered image

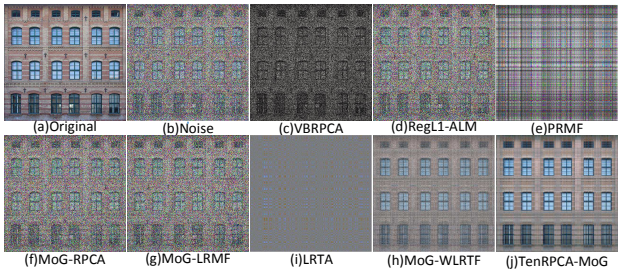

Figure 5. Bigger mixture noise for Facade. (a) Original image; (b) Noisy image; (c)-(j) Recovered image

ture noise, the methods of noise modeling with MoG (our method and MoG-WLRTF) provide better results than others, and our method achieves the best performance overall. These experiments illustrate the merits of tensor representation and MoG modeling, especially for complex noise.

7.3. Multispectral Image Restoration

Multispectral Image database includes image sets of various scenes, in which, each set includes 31 images of different bands with the size of 512×512. The mixed noises we add here are 10% of the elements mixed with uniform noise within [-5, 5], 70% of elements with Gaussian noise $\mathcal{N}(0, 1)$ and 20% of elements with Gaussian noise $\mathcal{N}(0, 0.01)$. We input 31 bands at a time and show the results of Band 31 in Fig. 2 for comparison. All the methods have effects on denoising to different extents. MoG-WLRTF performs much better than other competitive methods, but compared with our method, it loses more details and gets intensity deviations. In Fig. 3, we randomly choose 10 bands from the glass tiles multispectral set and show the denoising results by our method. All the bands are well recovered and preserve the differences between different bands as well.

7.4. RGB Image Restoration

Single RGB image denoising is a more challenging problem, especially for matrix based method. It is because that by vectoring and aligning the R, G, B channels, there are only 3 linearly related dimensions such that it is harder to seek a meaningful low dimensional subspace. We carry out two experiments on colorful building facade image by adding noises of different levels. In the first experiment, the added noises are: 10% of elements mixed with uniform noise within [-2.5,2.5], 20% of elements mixed with Gaussian noise $\mathcal{N}(0, 1)$ and 70% of elements mixed with Gaussian noise $\mathcal{N}(0, 0.01)$. As shown in Fig. 4, all the matrix based methods lose efficiency, while the tensor based methods perform much better. Compared with other effective methods, our method recovers more details and the colors are more consistent with the original image. In the second experiment, the added noises are bigger: 10% of elements mixed with uniform noise within [-25,25], 70% of elements mixed with Gaussian noise $\mathcal{N}(0, 1)$, and 20% of elements mixed with Gaussian noise $\mathcal{N}(0, 0.01)$. As shown in Fig. 5, almost all the competitive methods fail to recovery, as well as MoG-WLRTF. Although our method loses more details compared with the first experiment, the noise is well eliminated and the recovery is satisfactory.

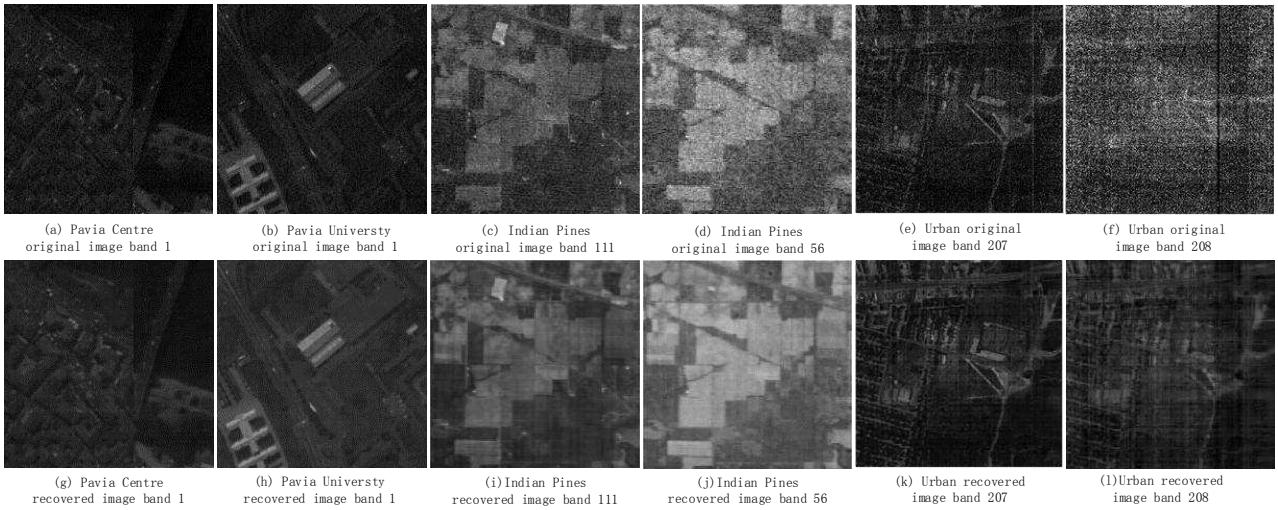

Figure 6. Test on real hyperspectral images from various datasets. (a)-(f) Original image; (g)-(l) Recovered image

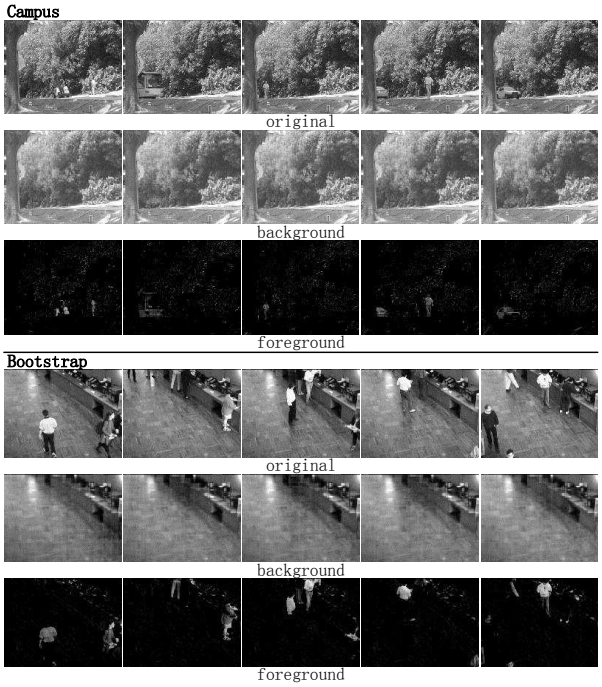

Figure 7. Background subtraction experiments on Campus and Bootstrap

7.5. Real Hyperspectral Image Restoration

To verify our method on dealing with real noise, we do the test on four different real hyperspectral images datasets, including Pavia Centre, Pavia Universty, Indian Pines and Urban. All of them are earth observation images taken from airbornes or satellites. Some bands are badly contaminated by the air stream or signal transmission loss. Fig. 6 shows the original images (the upper row) and the recovered images (the lower row) by our method. The results illustrate that our method can handle real unknown complex noise, even the gross one as shown in Urban band 208.

7.6. Background Substraction

We apply the proposed model to background subtraction experiments on general test video sequences Campus and Bootstrap. The results for randomly sample frames are shown in Fig. 7. We can see that our model is also effective for background extraction application.

8. Conclusion

In this paper, we propose a TenRPCA-MoG model for image denoising. Compared with the existing models, our model directly deals with the third-order tensor instead of two-dimensional matrix, which better preserves the original data structure. Simultaneously, it introduces MoG to model noise, which overcomes the disadvantages of traditional R-PCA model or tensor factorization methods only for specific type of noise. In addition, we design an algorithm under Bayesian framework for solving the model. Synthetic and real data experiments demonstrate the effectiveness of our method for complex noise.

9. Acknowledgement

We thank Qian Zhao for the helpful discussion on variational Bayesian method. This work was supported by the National Natural Science Foundation of China (Grant No. 61303168, 11501440, 61603293). The authors also thank the support by Youth Innovation Promotion Association CAS.

References

[1] S. Babacan, M. Luessi, R. Molina, and A. Katsaggelos. Sparse bayesian methods for low-rank matrix estimation. *IEEE Transactions on Signal Processing*, 60(8):3964–3977, 2012.

[2] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[3] C. M. Bishop. Pattern recognition and machine learning. *Springer New York*, 2006.

[4] E. Candès and R. Benjamin. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[5] E. Candès and T. Tao. The power of convex relaxation: Nearoptimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[6] E. Candèsn, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11:1–11:37, 2011.

[7] W. Cao, Y. Wang, J. Sun, D. Meng, C. Yang, A. Cichocki, and Z. Xu. Total variation regularized tensor rpca for background subtraction from compressive measurements. *IEEE Transactions on Image Processing*, 25(9), 2016.

[8] J. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scal ing via an n-way gener- alization of eckart-young decomposition. *Psychometrika*, 35:283–319, 1970.

[9] P. Chen. Optimization algorithms on subspaces: Revisiting missing data problem in low-rank matrix. *International Journal of Computer Vision*, 80:125142, 2008.

[10] X. Chen, Z. Han, Y. Wang, Q. Zhao, D. Meng, and Y. Tang. Robust tensor factorization with unknown noise. *Computer Vision and Pattern Recognition*, pages 5213–5221, 2016.

[11] E. Chi and T. Kolda. Making tensor factorizations robust to non-gaussian. *arXiv preprint arXiv*, 2010.

[12] F. De la Torre and M. Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1).

[13] C. Ding, D. Zhou, X. He, and H. Zha. R1-pca: Rotational invariant l1-norm principal component analysis for robust subspace factorization. *International Conference on Machine Learning*, pages 281–288, 2006.

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

[15] L. H. e. a. Huang J, Zhang S. Composite splitting algorithms for convex optimization. *Computer Vision and Image Understanding*, 115(12):1610–1622, 2011.

[16] Jolliffe.I.T. Principal component analysis. *Springer series in statistics. Springer, New York, 2nd edition*, 2002.

[17] Q. Ke and T. Kanade. Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming. *Computer Vision and Pattern Recognition*, pages 234–778, 2005.

[18] T. Kolda and B. Bader. Tensor decompositions and applications. *Society for Industrial and Applied Mathematics*, 51(3):455–500, 2009.

[19] N. Kwak. Principal component analysis based on l1-norm maximization. *Pattern Analysis and Machine Intelligence*, 30(9):1672–1680, 2008.

[20] X. Li, Bourennane, and C. Fossati. Tensor completion for estimating missing values in visual data. *Pattern Analysis and Machine Intelligence*, 34(1):208–220, 2013.

[21] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. *International Conference on Machine Learning*, pages 663–670, 2010.

[22] X. Liu, S. Bourennane, and C. Fossati. Denoising of hyperspectral images using the parafac and statistical performance analysis. *Geoscience and Remote Sensing*, 50(10):3717–3724, 2012.

[23] D. Meng and F. De la Torre. Robust matrix factorization with unknown noise. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1337–1344, 2013.

[24] K. Mitra, K. Mitra, and R. Chellappa. Large-scale matrix factorization with missing data under additional constraints. *Advances in Neural Information Processing Systems*, pages 1651–1659, 2010.

[25] T. Okatani and K. Deguchi. On the wiberg algorithm for matrix factorization in the presence of missing components. *International Journal of Computer Vision*, 72:329–337, 2007.

[26] T. Okatani, T. Yoshida, and K. Deguchi. Efficient algorithm for low-rank matrix factorization with missing components and performance comparison of latest algorithms. *Proceedings of the IEEE International Conference on Computer Vision*, 2011.

[27] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *Pattern Analysis and Machine Intelligence*, pages 2233–2246, 2010.

[28] H. Pragarauskas and O. Gross. Temporal collaborative filtering with bayesian probabilistic tensor factorization. *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 211–222, 2010.

[29] P. Rai, Y. Wang, S. Guo, G. Chen, D. Dunson, and L. Carin. Scalable bayesian low-rank decomposition of incomplete multiway tensors. *International Conference on Machine Learning*, pages 1800–1808, 2014.

[30] B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[31] N. Renard, S. Bourennane, and J. Blanc-Talon. Denoising and dimensionality reduction using multilinear tools for hyperspectral images. *Geoscience and Remote Sensing Letters*, 5(2):138–142, 2008.

[32] G. P. e. a. Sheng G A O, Denoyer L. Probabilistic latent tensor factorization model for link pattern prediction in multi-relational networks. *The Journal of China Universities of Posts and Telecommunications*, 20:172–181, 2012.

[33] D. L. L. e. a. Signoretto M, Dinh Q T. Learning with tensors: a framework based on convex optimization and spectral regularization. *Machine Learning*, 94(3):303–351, 2014.

[34] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Josa a*, 4(3):519–524, 1987.

[35] N. Srebro and T. Jaakkola. Weighted low-rank approximations. *International Conference on Machine Learning*, pages 720–727, 2003.

[36] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[37] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuro Science*, 3:71–86, 1991.

[38] M. Turk and A. Pentland. Face recognition using eigenfaces. *Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[39] R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using powerfactorization and gpca. *International Journal of Computer Vision*, 79(1):85–105, 2008.

[40] J. Wright, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted lowrank matrices by convex optimization. *Advances in Neural Information Processing Systems*, pages 2080–2088, 2009.

[41] K. Zhao and Z. Zhang. Successively alternate least square for low-rank matrix factorization with bounded missing data. *Computer Vision and Image Understanding*, 114:1084–1096, 2010.

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

[43] Q. Zhao, L. Zhang, and A. Cichocki. Bayesian cp factorization of incomplete tensors with automatic rank determi- nation. *Pattern Analysis and Machine Intelligence*, 37(9):1751–1763, 2015.

[44] Y. Zheng, G. Liu, S. Sugimoto, S. Yan, and M. Okutomi. Practical low-rank matrix approximation under robust l1-norm. *Computer Vision and Pattern Recognition (CVPR)*, pages 1410–1417, 2012.