Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Low-Rank Tensor Completion with Total Variation for Visual Data Inpainting

Xutao Li, Yunming Ye, Xiaofei Xu
Shenzhen Key Laboratory of Internet Information Collaboration,
Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, 518055, China
Emails: lixutao@hitsz.edu.cn; yym@hitsz.edu.cn;xiaofei@hit.edu.cn

Abstract

With the advance of acquisition techniques, plentiful higher-order tensor data sets are built up in a great variety of fields such as computer vision, neuroscience, remote sensing and recommender systems. The real-world tensors often contain missing values, which makes tensor completion become a prerequisite to utilize them. Previous studies have shown that imposing a low-rank constraint on tensor completion produces impressive performances. In this paper, we argue that low-rank constraint, albeit useful, is not effective enough to exploit the local smooth and piecewise priors of visual data. We propose integrating total variation into low-rank tensor completion (LRTC) to address the drawback. As LRTC can be formulated by both tensor unfolding and tensor decomposition, we develop correspondingly two methods, namely LRTC-TV-I and LRTC-TV-II, and their iterative solvers. Extensive experimental results on color image and medical image inpainting tasks show the effectiveness and superiority of the two methods against state-of-the-art competitors. Our codes are available at https://sites.google.com/site/xutaoli2014

Introduction

As a generalization of matrices and vectors, tensors refer to multiway arrays, which are powerful to represent multidimensional data or interactions related to multiple factors (Kolda and Bader 2009). For example, a color image (a video sequence) can be represented as a third-order tensor, where the three dimensions are *height*, *width* and *color channel* (*time channel*); similarly, magnetic resonance imaging (MRI) data recording pictures of a patient under different frequencies can be also denoted as a tensor; E-commerce data measuring the interactions among users, items and contexts is another example.

Due to loss of information or unacceptable cost to acquire complete data, tensors built up in real-world applications may contain missing values in their entries. Completing the values for missing entries, known as *tensor completion problem*, thus becomes an important research topic, which can serve plentiful applications, for instance, image or video inpainting (Liu et al. 2013; Yao and Kwok 2015; Romera-Paredes and Pontil 2013; Wang, Nie, and Huang 2014; Li et al. 2015), hyperspectral or MRI data recovery (Li and Li 2010; Ji et al. 2016; Shang, Liu, and Cheng 2014), context-aware recommendation (Karatzoglou et al. 2010; Rettinger et al. 2012), among others.

Matrix completion, namely the second-order tensor completion problem, has been intensively studied in the past decades (Candès and Recht 2009; Fazel 2002; Recht 2011; Candès and Tao 2010). Since the problem is ill-posed without any constraints, i.e., there are infinite solutions, a great number of studies assume the matrix to complete is low-rank and try to minimize its rank when filling in missing entries. However, the rank minimization is unfortunately discrete, non-convex and NP-hard. To address the challenge, nuclear norm (trace norm) is often utilized as a surrogate of the rank function, which is continuous, convex and easy to optimize. In fact, it has been shown theoretically to be the tightest convex approximation of a matrix's rank (Fazel 2002). Excellent performances in various domains demonstrate that low-rank regularization with nuclear norm is very effective for matrix completion.

Recently, low-rank constraint has also been imposed to recover higher-order tensors from partial observations. Different from matrix, the rank of a tensor is not well-defined. Theoretically, it is defined as the minimum number of $rank$-1 CANDECOMP/PARAFAC (CP) decomposition components. However, the definition is rarely used, as it is hard to determine, or even estimate, the rank of a tensor when adopting the definition in practice. Hence, many approaches extend and leverage the definition of matrix's rank for low-rank tensor completion (LRTC). Roughly speaking, they can be broken down into two directions. One line of methods utilizes linear combinations of the rank of unfolded matrix along each mode as low-rank regularization (Gandy, Recht, and Yamada 2011; Liu et al. 2013; Signoretto et al. 2011; Tomioka, Hayashi, and Kashima 2010; Signoretto et al. 2014), while the other line performs CP or Tucker decomposition and try to make the decomposition factors be low-rank (Chen, Hsu, and Liao 2014; Liu et al. 2014a; Filipović and Jukić 2015; Zhao, Zhang, and Cichocki 2015; Liu et al. 2014b). Previous studies have shown that the two types of methods yield impressive performances.

In this paper, we argue that low-rank constraint, albeit useful, is not effective enough to exploit some underlying local structures of tensors for completion. The point is in par-

ticular obvious for visual data inpainting. As well-known, visual data often exhibits smooth and piecewise structures in spatial dimension, due to objects or edges therein. Without special considerations on such kind of local structures, the inpainting results may be unsatisfactory. Total variation (TV), a well-known norm to preserve piecewise smooth prior, has been successfully applied to many image processing applications. As complementary information for low rankness, we propose to incorporate total variation into tensor recovery such that local piecewise smooth structures can be exploited. Specifically, if we expect a piecewise prior for one mode, an anisotropic total variation term on that mode is then introduced. As aforementioned, there are two ways to formulate the low-rank regularizations. Hence, we combine total variation with low-rank constraints, and examine the performance in the two cases. The main contributions of the paper can be summarized as follows:

- We propose to combine total variation and low-rank constraints for tensor completion. In particular, two methods are developed according to different regularization formulations, where one is based on tensor unfolding and the other is on tensor decomposition.
- Two iterative solvers are derived based on alternating direction method of multipliers (ADMM), which deliver the completion results of the two methods above.
- Extensive experiments on color image inpainting and MRI data recovery have been conducted. The results demonstrate that the two proposed methods indeed improve the recovery accuracy compared to the counterparts with only low-rank constraints. Moreover, the method based on tensor decomposition delivers the best performance, and outperforms state-of-the-art competitors.

Related Work

In terms of the way to formulate low-rank regularization, existing LRTC methods can be roughly classified into two categories, namely tensor unfolding formulation and tensor decomposition formulation.

Unfolding Formulation. The low-rank completion by tensor unfolding is first introduced in (Liu et al. 2009). In the work, they define the trace norm of a tensor as an average of all its unfolded matrices' trace norms. Low-rank completion is thus accomplished by minimizing the trace norm of the recovered tensor. However, as the unfolded matrix in each mode shares the same entries, their trace norms are interdependent and the defined tensor trace norm is difficult to minimize. Hence, an auxiliary matrix is introduced in each mode to split the interdependent terms for optimization. In a subsequent work, two enhanced methods, i.e. FaLRTC and HaLRTC, are developed, which result in state-of-the-art tensor completion performance (Liu et al. 2013). It has been shown that FaLRTC is faster than HaLRTC, but HaLRTC obtains higher recovery accuracy. With the unfolding formulation, two LRTC solvers are also developed by using Douglas-Rachford splitting technique and ADMM, respectively (Gandy, Recht, and Yamada 2011). In addition, Tomioka et al. develop three relaxations to estimate low-rank tensors, which are "tensor as a matrix", "constraint" and "mixture" models (Tomioka, Hayashi, and Kashima 2010). In (Signoretto et al. 2014), a general transductive learning framework is put forward under the unfolding formulation, which not only is able to complete missing values in feature tensors, but also predict their labels.

Decomposition Formulation. There are two ways to achieve LRTC with tensor decomposition formulation. On the one hand, standard tensor decomposition models CP and Tucker can be extended to handle missing data, where low-rank is obtained by specifying tensor ranks. For example, weighted CP models are introduced by performing decomposition with known values only (Tomasi and Bro 2005; Acar et al. 2011), which can predict values for missing entries with the decomposition result. However, the two algorithms require users to manually specify the tensor rank, which is a challenge. To address the drawback, a fully Bayesian tensor factorization model is developed under CP framework (FBCP) (Zhao, Zhang, and Cichocki 2015). By imposing a sparsity-inducing hierarchical prior on decomposition factors, FBCP can automatically estimate the tensor rank and produce state-of-the-art performance for visual data recovery. Similarly, a weighted Tucker model (WTuker), is introduced for LRTC (Filipović and Jukić 2015). Though WTucker also needs tensor ranks as input, numerical results manifest that good performance can be obtained by overestimating the ranks.

On the other hand, trace norm constraints can be imposed on decomposition factors of CP or Tucker for LRTC. For example, Liu et al. introduce trace norm regularization to the factor matrices of CP (Liu et al. 2014a). In another work, they propose gHOI method, which puts a tensor trace norm constraint on core tensor of higher-order orthogonal iteration and develops a rank-increasing scheme for LRTC (Liu et al. 2014b). Due to the scheme, gHOI is quite efficient. Chen et al. develop a method called simultaneous tensor decomposition and completion (STDC), which employs trace norm minimization for the factors of Tucker decomposition (Chen, Hsu, and Liao 2014). Moreover, to characterize the underlying structure between factors, a graph-Laplacian term is also utilized in STDC. STDC has shown state-of-the-art performance in visual data inpainting.

Tensors and TV. Although the notion of piecewise priors has been utilized for visual tensor data in two recent studies (Shi et al. 2015; Guo and Ma 2015), our focus differs from theirs. Shi et al. study image super-resolution instead of inpainting (Shi et al. 2015). Guo and Ma concentrate on developing a generalized TV-norm, which can model the inhomogeneity and multi-directionality of visual tensors (Guo and Ma 2015). However, our paper aims at combining TV and low-rankness to recover visual tensors, where TV accounts for local piecewise priors and low-rankness exploits regular global patterns. In (Ji et al. 2016), a tensor recovery method is proposed by considering both TV and low-rankness. Specifically, the method performs low-rank matrix factorizations to all-mode matricizations of the tensor, and then imposes an isotropic TV on the factor matrices. Compared with our models, the method has three drawbacks: (i) imposing TV on factor matrices is hard to explain; (ii) the rank of each mode needs to be specified for low-rank factor-

izations, which is difficult; (iii) isotropic TV used is inflexible to model piecewise priors on arbitrary modes.

Low-Rank Tensor Completion with TV

In this section, we propose to incorporate total variation constraint for LRTC. Since LRTC can be formulated by tensor unfolding and tensor decomposition, two TV methods will be developed.

Method 1. Given an incomplete tensor $\mathcal{Y} \in R^{J_1 \times J_2 \times \cdots \times J_N}$ with Ω indicating the set of indices of observations, we propose the following objective function to recover it:

$$\min_{\mathcal{Z}} \lambda \sum_{n=1}^{N} \beta_n \left|\mathbf{F}_n \mathbf{Z}_{(n)}\right| + \frac{1}{N} \sum_{n=1}^{N} \left\|\mathbf{Z}_{(n)}\right\|_* \quad s.t. \quad [\mathcal{Z}]_\Omega = [\mathcal{Y}]_\Omega \tag{1}$$

Here tensor $\mathcal{Z}$ represents the recovery result; $\mathbf{Z}_{(n)}$ denotes its mode-n unfolding matrix; $\mathbf{F}_n$ is a $(J_n - 1)$-by-J_n matrix, where $[\mathbf{F}_n]_{i,i} = 1$, $[\mathbf{F}_n]_{i,i+1} = -1$ and the other entries are zeros; the operator $|\cdot|$, defined as $|\mathbf{A}| = \sum_{i=1}\sum_{j=1} |[\mathbf{A}]_{i,j}|$, is utilized to develop our TV regularizer, and $\|\cdot\|_*$ denotes trace norm of a matrix, which is for fulfilling the low-rank constraint; λ is a tunable parameter; $\beta_1, \cdots, \beta_n$ is 0 or 1, which indicates whether we have a smooth and piecewise prior on the n-th mode of recovered tensor. The settings of $\beta_1, \beta_2, \cdots, \beta_N$ are domain dependent. For example, when $\mathcal{Y}$ is a tensor of color image, we set $\beta_1 = \beta_2 = 1$ and $\beta_3 = 0$, because only spatial dimensions are expected to have smooth and piecewise priors.

The objective function in Eq. (1) comprises of two terms, where the second term models the low-rank constraint, and the first one stands for our TV regularizer. The TV regularizer follows an anisotropic version. We adopt it for two reasons: it is (i) easier to optimize; and (ii) more flexible to impose piecewise smooth priors on different modes. Note, without the TV term, our model degenerates into HaLRTC.

Because the two terms in our model share the same variable $\mathbf{Z}_{(n)}$, they are interdependent. Furthermore, the variables $\mathbf{Z}_{(1)}, \mathbf{Z}_{(2)}, \cdots, \mathbf{Z}_{(n)}$ are also interdependent in that these unfolded matrices share the same entries of tensor $\mathcal{Z}$. The interdependencies make our model difficult to optimize. Fortunately, ADMM technique provides a splitting scheme to cope with such type of problems. Hence, we adopt it to develop our solver. Specifically, by introducing a set of matrices $\{\mathbf{Q}_n\}_{n=1}^N$, $\{\mathbf{M}_n\}_{n=1}^N$ and $\{\mathbf{R}_n\}_{n=1}^N$ as auxiliary variables, we split the interdependencies and rewrite the optimization problem as

$$\min_{\mathcal{Z}, \{\mathbf{Q}_n, \mathbf{R}_n, \mathbf{M}_n\}_{n=1}^N} \lambda \sum_{n=1}^{N} \beta_n \left|\mathbf{Q}_n\right| + \frac{1}{N} \sum_{n=1}^{N} \left\|\mathbf{M}_n\right\|_* \quad s.t. \quad \{\mathbf{Q}_n = \mathbf{F}_n \mathbf{R}_n, \mathbf{M}_n = \mathbf{Z}_{(n)}, \mathbf{R}_n = \mathbf{Z}_{(n)}\}_{n=1}^N \quad [\mathcal{Z}]_\Omega = [\mathcal{Y}]_\Omega \tag{2}$$

By using the augmented Lagrange formulation (Boyd et al. 2011), the optimization problem is changed into:

$$\begin{aligned}\mathcal{L} = & \sum_{n=1}^{N} \beta_n \cdot \left(\lambda \left|\mathbf{Q}_n\right| + \frac{\rho_1}{2} \left\|\mathbf{Q}_n - \mathbf{F}_n \mathbf{R}_n + \frac{\mathbf{\Lambda}_n}{\rho_1}\right\|_F^2\right) \\ & + \sum_{n=1}^{N} \beta_n \cdot \left(\frac{\rho_2}{2} \left\|\mathbf{R}_n - \mathbf{Z}_{(n)} + \frac{\mathbf{\Phi}_n}{\rho_2}\right\|_F^2\right) \\ & + \sum_{n=1}^{N} \left(\frac{1}{N} \left\|\mathbf{M}_n\right\|_* + \frac{\rho_3}{2} \left\|\mathbf{M}_n - \mathbf{Z}_{(n)} + \frac{\mathbf{\Gamma}_n}{\rho_3}\right\|_F^2\right) \\ s.t. \quad & [\mathcal{Z}]_\Omega = [\mathcal{Y}]_\Omega \end{aligned} \tag{3}$$

where matrices $\{\mathbf{\Lambda}_n\}_{n=1}^N$, $\{\mathbf{\Phi}_n\}_{n=1}^N$ and $\{\mathbf{\Gamma}_n\}_{n=1}^N$ are Lagrange multipliers; $\|\cdot\|_F$ stands for Frobenius norm of a matrix or a tensor. Next, we derive the update formulae of $\{\mathbf{Q}_n\}_{n=1}^N$, $\{\mathbf{M}_n\}_{n=1}^N$, $\{\mathbf{R}_n\}_{n=1}^N$ and $\mathcal{Z}$ to develop our solver.

Updating $\{\mathbf{Q}_n\}_{n=1}^N$: By keeping the other variables fixed, the optimization problem w.r.t. $\{\mathbf{Q}_n\}_{n=1}^N$ is given by:

$$\min_{\{\mathbf{Q}_n\}_{n=1}^N} \sum_{n=1}^{N} \beta_n \cdot \left(\lambda \left|\mathbf{Q}_n\right| + \frac{\rho_1}{2} \left\|\mathbf{Q}_n - \mathbf{F}_n \mathbf{R}_n + \frac{\mathbf{\Lambda}_n}{\rho_1}\right\|_F^2\right) \tag{4}$$

As $\mathbf{Q}_1, \mathbf{Q}_2, \cdots, \mathbf{Q}_N$ are independent in the optimization problem, we can easily derive the update formula of $\mathbf{Q}_n$ as:

$$\mathbf{Q}_n = \beta_n \cdot \text{shrinkage}_{\frac{\lambda}{\rho_1}} \left(\mathbf{F}_n \mathbf{R}_n - \frac{1}{\rho_1} \mathbf{\Lambda}_n\right) \tag{5}$$

where $\text{shrinkage}_\alpha(\cdot)$ is the elementwise shrinkage-thresholding operator of a matrix, i.e.,

$$[\text{shrinkage}_\alpha(\mathbf{A})]_{i,j} = [\mathbf{A}]_{i,j} - \min(\alpha, |[\mathbf{A}]_{i,j}|) \cdot \frac{[\mathbf{A}]_{i,j}}{|[\mathbf{A}]_{i,j}|},$$

and $\frac{[\mathbf{A}]_{i,j}}{|[\mathbf{A}]_{i,j}|}$ is defined as zero when $[\mathbf{A}]_{i,j} = 0$.

Updating $\{\mathbf{M}_n\}_{n=1}^N$: Fixing the other variables, we obtain the following optimization problem w.r.t. $\{\mathbf{M}_n\}_{n=1}^N$:

$$\min_{\{\mathbf{M}_n\}_{n=1}^N} \sum_{n=1}^{N} \frac{1}{N} \left\|\mathbf{M}_n\right\|_* + \frac{\rho_3}{2} \left\|\mathbf{M}_n - \mathbf{Z}_{(n)} + \frac{\mathbf{\Gamma}_n}{\rho_3}\right\|_F^2 \tag{6}$$

Optimizing the problem leads to the solution of $\mathbf{M}_n$ as:

$$\mathbf{M}_n = D_{\frac{1}{N \cdot \rho_3}} \left(\mathbf{Z}_{(n)} - \frac{1}{\rho_3} \mathbf{\Gamma}_n\right) \tag{7}$$

where $D_\alpha(\mathbf{A}) = \mathbf{U}(\text{diag}\{\delta - \alpha\})_+ \mathbf{V}^T$ is a singular value thresholding operator, the SVD of $\mathbf{A}$ is denoted by $\mathbf{U}(\text{diag}(\{\delta_\text{i}\}_{0 \le \text{i} \le \text{rank}(\mathbf{A})}))\mathbf{V}^T$, and $t_+ = \max(0, t)$.

Updating $\{\mathbf{R}_n\}_{n=1}^N$: By keeping the other variables fixed, the optimization problem w.r.t. $\{\mathbf{R}_n\}_{n=1}^N$ is given by:

$$\begin{aligned}\min_{\{\mathbf{R}_n\}_{n=1}^N} & \sum_{n=1}^{N} \beta_n \frac{\rho_1}{2} \left\|\mathbf{Q}_n - \mathbf{F}_n \mathbf{R}_n + \frac{\mathbf{\Lambda}_n}{\rho_1}\right\|_F^2 \\ & + \sum_{n=1}^{N} \beta_n \frac{\rho_2}{2} \left\|\mathbf{R}_n - \mathbf{Z}_{(n)} + \frac{\mathbf{\Phi}_n}{\rho_2}\right\|_F^2 \end{aligned} \tag{8}$$

Hence, the following update formula is derived by solving the minimization problem:

$$\begin{aligned}\mathbf{R}_n = & \beta_n (\rho_1 \mathbf{F}_n^T \mathbf{F}_n + \rho_2 \mathbf{I})^{-1} \\ & (\mathbf{F}_n^T \mathbf{\Lambda}_n + \rho_1 \mathbf{F}_n^T \mathbf{Q}_n + \rho_2 \mathbf{Z}_{(n)} - \mathbf{\Phi}_n)\end{aligned} \tag{9}$$

where $\mathbf{I}$ stands for the identify matrix.

Updating $\mathcal{Z}$: Similarly, the optimization problem w.r.t. $\mathcal{Z}$ is:

$$\begin{aligned}\min_{\mathcal{Z}} \sum_{n=1}^{N} \beta_n \cdot \left(\frac{\rho_2}{2} \left\| \mathbf{R}_n - \mathbf{Z}_{(n)} + \frac{\mathbf{\Phi}_n}{\rho_2} \right\|_F^2 \right) \\ + \sum_{n=1}^{N} \frac{\rho_3}{2} \left\| \mathbf{M}_n - \mathbf{Z}_{(n)} + \frac{\mathbf{\Gamma}_n}{\rho_3} \right\|_F^2 \\ s.t. \qquad [\mathcal{Z}]_\Omega = [\mathcal{Y}]_\Omega \end{aligned} \tag{10}$$

The update formulae of $\mathcal{Z}$ are computed as:

$$[\mathcal{Z}]_{\bar{\Omega}} = \left[\frac{\sum_{n=1}^{N} (\text{fold}_n(\mathbf{\Gamma}_n + \rho_3 \mathbf{M}_n) + \text{fold}_n(\mathbf{\Phi}_n + \rho_2 \mathbf{R}_n))}{(N\rho_3 + \sum_{n=1}^{N} \beta_n \rho_2)} \right]_{\bar{\Omega}}$$

and

$$[\mathcal{Z}]_\Omega = [\mathcal{Y}]_\Omega \,. \tag{11}$$

Here $\text{fold}_n(\cdot)$ denotes the opposite operation of mode-n unfolding of a tensor, i.e., $\text{fold}_n(\mathbf{A}_{(n)}) = \mathcal{A}$.

With these update formulae, we summarize the solver of method 1, namely LRTC-TV-I, in Algorithm 1. The solver is an iterative algorithm under ADMM framework. In line 3, the auxiliary matrices $\{\mathbf{Q}_n\}_{n=1}^N$, $\{\mathbf{M}_n\}_{n=1}^N$, $\{\mathbf{R}_n\}_{n=1}^N$ and target output variable $\mathcal{Z}$ are updated according to our derivations. In lines 4∼6, we renew the Lagrange multipliers $\mathbf{\Lambda}_n$, $\mathbf{\Phi}_n$ and $\mathbf{M}_n$ as the standard ADMM. $\{\rho_i\}_{i=1}^3$ are adaptively increased in line 7 for a better speed of convergence (Lin, Chen, and Ma 2009). The main computational costs of LRTC-TV-I lie at updates of $\{\mathbf{M}_n\}_{n=1}^N$ as Eq. (7), which needs perform SVD on a matrix of size J_n-by-$\Pi_{i\neq n}^N J_i$. Hence, the time complexity of LRTC-TV-I is $O(K(\sum_{n=1}^N (J_n)^2 (\Pi_{i\neq n}^N J_i) + (\Pi_{i\neq n}^N J_i)^3))$.

Algorithm 1: LRTC-TV-I

input : an incomplete tensor $\mathcal{Y}$, iteration number K, parameters λ, $\{\rho_i\}_{i=1}^3$ and $\mu \in [1.1, 1.5]$
output: a recovery tensor $\mathcal{Z}$

1 Set $[\mathcal{Z}]_\Omega = [\mathcal{Y}]_\Omega$, $[\mathcal{Z}]_{\bar{\Omega}} = 0$, and randomly initialize $\{\mathbf{Q}_n\}_{n=1}^N$, $\{\mathbf{R}_n\}_{n=1}^N$ and $\{\mathbf{M}_n\}_{n=1}^N$;
2 **for** $k = 1$ *to* K **do**
3 Update $\{\mathbf{Q}_n\}_{n=1}^N$, $\{\mathbf{M}_n\}_{n=1}^N$, $\{\mathbf{R}_n\}_{n=1}^N$ and $\mathcal{Z}$ as Eqs. (5), (7), (9), (11);
4 $\mathbf{\Lambda}_n = \mathbf{\Lambda}_n + \rho_1(\mathbf{Q}_n - \mathbf{F}_n\mathbf{R}_n)$;
5 $(n = 1, 2, \cdots, N)$ $\mathbf{\Phi}_n = \mathbf{\Phi}_n + \rho_2(\mathbf{R}_n - \mathbf{Z}_{(n)})$;
6 $(n = 1, 2, \cdots, N)$ $\mathbf{\Gamma}_n = \mathbf{\Gamma}_n + \rho_3(\mathbf{M}_n - \mathbf{Z}_{(n)})$;
7 $(n = 1, 2, \cdots, N)$ $\rho_1 = \mu\rho_1$, $\rho_2 = \mu\rho_2$ and $\rho_3 = \mu\rho_3$;
8 **return** $\mathcal{Z}$.

Method 2. In this method, given an incomplete tensor $\mathcal{Y}$, we consider incorporating TV regularization into tensor decomposition for LRTC. Specifically, the following objective function is proposed and designed:

$$\begin{aligned}\min_{\substack{\mathcal{Z},\mathcal{G},\\ \{\mathbf{U}^{(n)}\}_{n=1}^N}} \lambda_1 \sum_{n=1}^{N} \beta_n \left| \mathbf{F}_n \mathbf{Z}_{(n)} \right| + \frac{1}{N} \sum_{n=1}^{N} \left\| \mathbf{U}^{(n)} \right\|_* + \lambda_2 \left\| \mathcal{G} \right\|_F^2 \\ s.t. \qquad \mathcal{Z} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)} \\ [\mathcal{Z}]_\Omega = [\mathcal{Y}]_\Omega \end{aligned} \tag{12}$$

In Eq. (12), we perform Tucker decomposition for $\mathcal{Z}$ and $\mathcal{Z} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)}$, where $\mathcal{G} \in R^{J_1 \times J_2 \times \cdots \times J_N}$ is the core tensor, and $\{\mathbf{U}^{(n)}\}_{n=1}^N$ denote the decomposition factors whose sizes are J_n-by-J_n ($n = 1, 2, \cdots, N$). We choose Tucker, instead of CP decomposition, because Tucker is a more general model and CP can be regarded as one of its special instantiations.

Our objective function in Eq. (12) comprises of three terms. Similar to method 1, the first term still denotes TV-regularization on the recovered tensor $\mathcal{Z}$. Instead of imposing low-rank constraints on the unfolded matrices of tensor $\mathcal{Z}$, we expect its Tucker decomposition factors to be low-rank, which is expressed by the second term in Eq. (12). The last term is utilized to avoid overfitting. If we remove the TV term, our mode degenerates into STDC, but with the difference that the reduced model does not have a graph-Laplacian regularizer on factor matrices. Again, because the three terms in Eq. (12) are interdependent, we adopt ADMM for optimization. In particular, we introduce a set of matrices $\{\mathbf{Q}_n\}_{n=1}^N$, $\{\mathbf{R}_n\}_{n=1}^N$ and $\{\mathbf{V}^{(n)}\}_{n=1}^N$ as auxiliary variables, and change the optimization problem into:

$$\begin{aligned}\min \lambda_1 \sum_{n=1}^{N} \beta_n \left| \mathbf{Q}_{(n)} \right| + \frac{1}{N} \sum_{n=1}^{N} \left\| \mathbf{U}^{(n)} \right\|_* + \lambda_2 \left\| \mathcal{G} \right\|_F^2 \\ s.t. \quad \{\mathbf{Q}_n = \mathbf{F}_n\mathbf{R}_n, \mathbf{R}_n = \mathbf{Z}_{(n)}, \mathbf{V}^{(n)} = \mathbf{U}^{(n)}\}_{n=1}^N \\ \mathcal{Z} = \mathcal{G} \times_1 \mathbf{V}^{(1)} \times_2 \mathbf{V}^{(2)} \cdots \times_N \mathbf{V}^{(N)} \\ [\mathcal{Z}]_\Omega = [\mathcal{Y}]_\Omega \end{aligned} \tag{13}$$

By using the augmented Lagrange formulation, the following optimization problem is obtained:

$$\begin{aligned}\mathcal{L} = \sum_{n=1}^{N} \beta_n \cdot \left(\lambda_1 \left| \mathbf{Q}_n \right| + \frac{\rho_1}{2} \left\| \mathbf{Q}_n - \mathbf{F}_n\mathbf{R}_n + \frac{\mathbf{\Lambda}_n}{\rho_1} \right\|_F^2 \right) \\ + \sum_{n=1}^{N} \beta_n \cdot \left(\frac{\rho_2}{2} \left\| \mathbf{R}_n - \mathbf{Z}_{(n)} + \frac{\mathbf{\Phi}_n}{\rho_2} \right\|_F^2 \right) \\ + \frac{1}{N} \sum_{n=1}^{N} \left\| \mathbf{U}^{(n)} \right\|_* + \sum_{n=1}^{N} \frac{\rho_3}{2} \left\| \mathbf{V}^{(n)} - \mathbf{U}^{(n)} + \frac{\mathbf{\Gamma}_n}{\rho_3} \right\|_F^2 \\ + \lambda_2 \left\| \mathcal{G} \right\|_F^2 + \frac{\rho_4}{2} \left\| \mathcal{Z} - \mathcal{G} \times_1 \mathbf{V}^{(1)} \cdots \times_N \mathbf{V}^{(N)} + \frac{\mathcal{W}}{\rho_4} \right\|_F^2 \\ s.t. \qquad [\mathcal{Z}]_\Omega = [\mathcal{Y}]_\Omega \end{aligned} \tag{14}$$

Here matrices $\{\mathbf{\Lambda}_n\}_{n=1}^N$, $\{\mathbf{\Phi}_n\}_{n=1}^N$, $\{\mathbf{\Gamma}_n\}_{n=1}^N$ and tensor $\mathcal{W}$ are Lagrange multipliers. Next, we present some update formulae to finalize the solver of method 2, which are derived in the same way as in method 1.

Updating Formulae: It is easy to check that $\{\mathbf{Q}_n\}_{n=1}^N$ and $\{\mathbf{R}_n\}_{n=1}^N$ are still updated as Eqs. (5) and (9), respectively. The formulae of $\{\mathbf{U}^{(n)}\}_{n=1}^N$, $\{\mathbf{V}^{(n)}\}_{n=1}^N$, $\mathcal{Z}$ and $\mathcal{G}$ are given by:

$$\mathbf{U}^{(n)} = D_{\frac{1}{N\rho_3}}\left(\mathbf{V}^{(n)} + \frac{1}{\rho_3}\mathbf{\Gamma}_n\right) \tag{15}$$

$$\mathbf{V}^{(n)} = \left(-\mathbf{\Gamma}_n + \rho_3\mathbf{U}^{(n)} + (\mathbf{W}_{(n)} + \rho_4\mathbf{Z}_{(n)})\mathbf{V}^{(-n)}\mathbf{G}_{(n)}^T\right) \left(\rho_3\mathbf{I} + \rho_4\mathbf{G}_{(n)}\mathbf{V}^{(-n)^T}\mathbf{V}^{(-n)}\mathbf{G}_{(n)}^T\right)^{-1} \tag{16}$$

$$[\mathcal{Z}]_{\bar{\Omega}} = \left[\frac{\sum_{n=1}^N \beta_n\left(-\text{fold}_n(\mathbf{\Phi}_n + \rho_2\mathbf{R}_n)\right) - \mathcal{W} + \rho_4\mathcal{T}}{\sum_{n=1}^N \beta_n\rho_2 + \rho_4}\right]_{\bar{\Omega}} \tag{17}$$

$$\text{vec}(\mathcal{G}) = \left[\mathbf{V}^{(-n)^T}\mathbf{V}^{(-n)} \otimes \rho_4\mathbf{V}^{(n)^T}\mathbf{V}^{(n)} + \lambda_2\mathbf{I}\right]^{-1} \text{vec}\left(\mathbf{V}^{(n)^T}(\mathbf{W}_{(n)} + \rho_4\mathbf{Z}_{(n)})\mathbf{V}^{(-n)}\right) \tag{18}$$

where $\mathbf{V}^{(-n)} = \mathbf{V}^{(1)} \otimes \mathbf{V}^{(2)} \cdots \otimes \mathbf{V}^{(n-1)} \otimes \mathbf{V}^{(n+1)} \cdots \otimes \mathbf{V}^{(N)}$, $\otimes$ is the Kronecker product, in Eq. (17) $\mathcal{T} = \mathcal{G} \times_1 \mathbf{V}^{(1)} \times_2 \mathbf{V}^{(2)} \cdots \times_N \mathbf{V}^{(N)}$, and $\text{vec}(\mathcal{G})$ in Eq. (18) denotes the vectorization of tensor $\mathcal{G}$, which can be reformed into tensor $\mathcal{G}$ after the calculation.

With the update formulae, we can adapt the algorithm 1 to develop the solver of method 2, namely LRTC-TV-II. Specifically, LRTC-TV-II can be obtained by replacing the update formulae in line 3 with Eqs. (5), (9), (15), (16), (17) and (18), and the updates of Lagrange multipliers in line 6 with $\mathbf{\Gamma}_n = \mathbf{\Gamma}_n + \rho_3(\mathbf{V}^{(n)} - \mathbf{U}^{(n)})$ and $\mathcal{W} = \mathcal{W} + \rho_4(\mathcal{Z} - \mathcal{G} \times_1 \mathbf{V}^{(1)} \cdots \times_n \mathbf{V}^{(n)})$. Moreover, we add one more equation $\rho_4 = \mu\rho_4$ in line 7. The computational bottlenecks of LRTC-TV-II are updates as Eq. (18), which needs compute the inverse of a matrix. Hence, its time complexity is given by $O(K(\Pi_{i=1}^N J_i)^3)$. Due to the space limitation, we do not present the algorithm of LRTC-TV-II.

Experiments

In this section, we present extensive experimental results to examine the performance of the two proposed methods. Two inpainting tasks are considered, where one is to complete RGB-color images from partial observations, and the other is to recover MRI medical images. The HaLRTC, FBCP, WTucker, gHOI and STDC are utilized as baseline methods. The experimental results are evaluated by the relative squared error (RSE) and peak signal-to-noise ratio (PSNR), which are widely used in the tensor-completion and visual data inpainting tasks. The two metrics are defined as:

$$\text{RSE} = \frac{\|\mathcal{Z} - \mathcal{Z}_{\text{true}}\|_F}{\|\mathcal{Z}_{\text{true}}\|_F} \tag{19}$$

$$\text{PSNR} = 10\log_{10}\frac{\hat{Z}_{\text{true}}^2}{\frac{1}{\Pi_{n=1}^N J_n}\|\mathcal{Z} - \mathcal{Z}_{\text{true}}\|_F^2} \tag{20}$$

where $\mathcal{Z}$, $\mathcal{Z}_{\text{true}}$ and $\hat{Z}_{\text{true}}$ represent the recovered tensor, ground-truth tensor, and the maximum value in the ground-truth tensor, respectively. Smaller RSE and larger PSNR indicate better recovery performance.

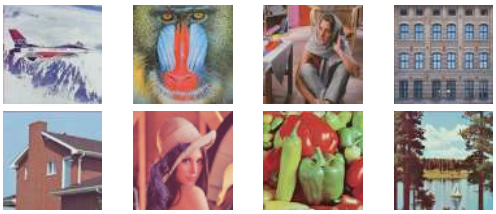
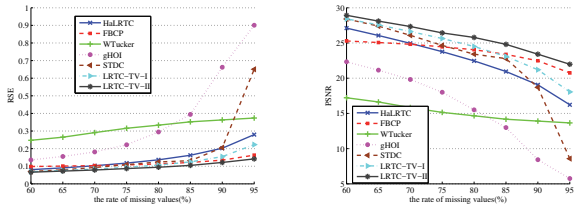


Figure 1: Ground truth of eight benchmark color images.



(a) RSE Metric (b) PSNR Metric

Figure 2: Results on color image inpainting.

Parameter Settings. In both inpainting tasks, tensors of third order are considered, whose first two modes stand for spatial dimensions and third mode denotes channel information. Hence, we set $\beta_1 = \beta_2 = 1$ and $\beta_3 = 0$ for LRTC-TV-I and LRTC-TV-II in both tasks. In LRTC-TV-I, the parameter λ is set to be 2.0×10^{-2}. In LRTC-TV-II, we utilize $\lambda_1 = 5.0 \times 10^{-1}$ and $\lambda_2 = 1.0 \times 10^3$. For both methods, we set $K = 300$, $\mu = 1.1$ and $\rho_1 = \rho_2 = \rho_3(= \rho_4) = 1.0 \times 10^{-2}$. The parameters of baseline approaches are specified optimally by following the papers in which these approaches were developed and proposed.

Color Image Inpainting

We show in Figure 1 the ground-truth of eight images used for the experiment. Each image is with the resolution of 256-by-256 pixels and three color channels, which thus can be represented as a 256-by-256-by-3 tensor. To test the inpainting performance, we mask off 60%, 65%, 70%, 75%, 80%, 85%, 90% and 95% of entries in each image randomly, and regard them as missing values. The remaining points, making up the incomplete tensor $\mathcal{Y}$, are leveraged to recover the original tensor. We record the RSE and PSNR of the eight images, and compute their averages respectively as final results.

Figure 2 presents the performance of all the methods. The following three observations can be made from the figure. First, the proposed method LRTC-TV-II performs the best among all the approaches, including LRTC-TV-I. The observation demonstrates that incorporating TV into tensor completion is helpful; and a tensor decomposition based nonconvex formulation (LRTC-TV-II) works better than an unfolding based convex formulation (LRTC-TV-I), especially for sparse observations, which is in accordance with the theoretical result that nonconvex recovery formulations often require fewer observations than convex ones (Oymak et al. 2015). Second, LRTC-TV-I produces better results than

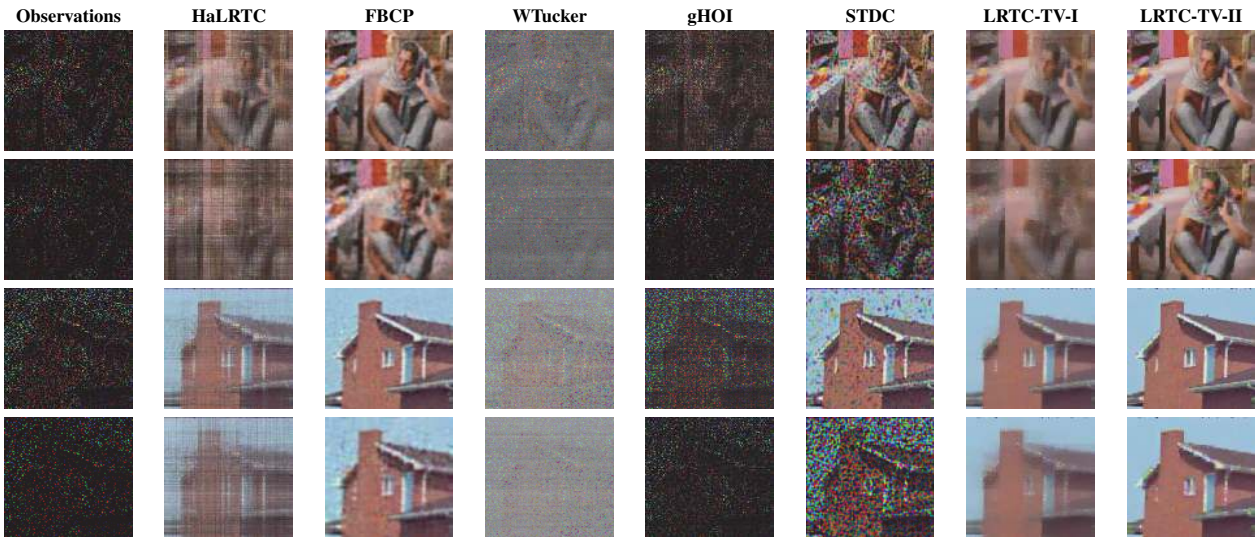
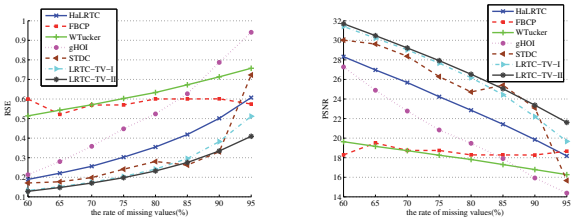


Figure 3: Visual effect comparison of different methods on color image inpainting. From top to bottom, each row denotes the results of different methods on barbara and house with missing rates being 90% and 95%, respectively.



(a) RSE Metric (b) PSNR Metric

Figure 4: Results on MRI medial data recovery.

all the other algorithms with only one exception, namely FBCP. The performance of LRTV-TV-I is first superior to that of FBCP and then becomes inferior when we increase the missing rate. This is because LRTV-TV-I fulfills the low-rank constraints under unfolding formulation, which may be less effective to characterize the low-rank structure than decomposition formulation when the observations are extremely sparse (Chen, Hsu, and Liao 2014). Third, WTucker and gHOI perform the worst. For WTucker, it overestimates the tensor ranks and does not really impose low-rank constraints. The gHOI utilizes a rank increasing scheme for better efficiency, which may hurt the accuracy when observations are very sparse.

To make a further investigation and comparison, we visualize the recovered results of some examples in Figure 3. We can see that the visual effects of WTucker and gHOI are unsatisfactory when the missing rate is at 90% or 95%. HaLRTC can roughly recover the images. However, the objects in the original images are not well-inpainted. Moreover, the recovered images are nonsmooth, which contain vertical or horizontal noisy lines. This is because HaLRTC neglects the local smooth and piecewise property of visual data. LRTC-TV-I, as an enhanced model of HaLRTC with TV-regularization, obtains more clear objects and smoother pictures. STDC can basically recover the images, while the results contain many isolated noisy points, especially when the missing rate is at 95%. The reason may be that STDC builds a big graph to model the underlying interactions between its decomposition factors. However, the graph construction relies on the completion result of HaLRTC, which can be less qualified as we observe. FBCP delivers good visual effects but with minor blurring parts. Its good performance is in that FBCP automatically estimates the rank of a tensor. However, as we will see in next section, its performance can be pretty bad when applied to MRI data recovery. LRTC-TV-II yields the best visual effects among all the methods. The inpainting results are smooth and with well-depicted edges of objects. The excellent performance is attributed to the TV constraints imposed on the recovered tensors.

MRI Medical Data Recovery

In this experiment, we aim to test the performance of different approaches on MRI medical image inpainting task. The BRAINIX data set[1] is utilized, and we build a 288-by-288-by-22 tensor. Again, we randomly remove 60%, 65%, 70%, 75%, 80%, 85%, 90% and 95% points in the tensor as missing values. Figure 4 shows the performance of different approaches. We can see that LRTC-TV-II performs the best, followed by LRTC-TV-I. The result demonstrates the superiority of TV constraints for LRTC. Though STDC produces competitive performance against LRTC-TV-I, its performance becomes much worse when the missing rate is at 95%. Surprisingly, FBCP, performing quite well for color image recovery, fails to deliver good results on this task. The reason may lie at the size difference of tensors in the two tasks. MRI tensor is much larger than color image tensor,

[1] http://www.osirix-viewer.com/datasets/

especially for the third mode. Large tensors may pose FBCP difficulties in accurately estimating the true ranks. Hence, the recovery performance degrades significantly.

Concluding Remarks

In this paper, we propose to integrate TV into LRTC for modeling the local smooth and piecewise priors of visual data. Because LRTC can be formulated by tensor unfolding and tensor decomposition, two methods, namely LRTC-TV-II and LRTC-TV-I, are developed. We derive their solvers under the ADMM framework. Extensive experimental results have demonstrated the superiority of the proposed methods over state-of-the-art techniques.

Acknowledgments

The research was supported in part by NSFC under Grant Nos. 61602132 and 61572158, and Shenzhen Science and Technology Program under Grant Nos. JCYJ20160330163900579 and JSGG20150512145714247.

References

Acar, E.; Dunlavy, D. M.; Kolda, T. G.; and Mørup, M. 2011. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems* 106(1):41–56.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* 3(1):1–122.

Candès, E. J., and Recht, B. 2009. Exact matrix completion via convex optimization. *Foundations of Computational mathematics* 9(6):717–772.

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

Chen, Y.-L.; Hsu, C.-T.; and Liao, H.-Y. M. 2014. Simultaneous tensor decomposition and completion using factor priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(3):577–591.

Fazel, M. 2002. *Matrix rank minimization with applications*. Ph.D. Dissertation.

Filipović, M., and Jukić, A. 2015. Tucker factorization with missing data with application to low-n-rank tensor completion. *Multidimensional Systems and Signal Processing* 26(3):677–692.

Gandy, S.; Recht, B.; and Yamada, I. 2011. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems* 27(2):025010.

Guo, X., and Ma, Y. 2015. Generalized tensor total variation minimization for visual data recovery? In *CVPR*, 3603–3611.

Ji, T.-Y.; Huang, T.-Z.; Zhao, X.-L.; Ma, T.-H.; and Liu, G. 2016. Tensor completion using total variation and low-rank matrix factorization. *Information Sciences* 326:243–257.

Karatzoglou, A.; Amatriain, X.; Baltrunas, L.; and Oliver, N. 2010. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Recsys*, 79–86. ACM.

Kolda, T. G., and Bader, B. W. 2009. Tensor decompositions and applications. *SIAM review* 51(3):455–500.

Li, N., and Li, B. 2010. Tensor completion for on-board compression of hyperspectral images. In *ICIP*, 517–520.

Li, C.; Zhao, Q.; Li, J.; Cichocki, A.; and Guo, L. 2015. Multi-tensor completion with common structures. In *AAAI*.

Lin, Z.; Chen, M.; and Ma, Y. 2009. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Technical Report UILU-ENG-09-2215, UIUC, (arXiv: 1009.5055)*.

Liu, J.; Musialski, P.; Wonka, P.; and Ye, J. 2009. Tensor completion for estimating missing values in visual data. In *ICCV*, 2114–2121.

Liu, J.; Musialski, P.; Wonka, P.; and Ye, J. 2013. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(1):208–220.

Liu, Y.; Shang, F.; Cheng, H.; Cheng, J.; and Tong, H. 2014a. Factor matrix trace norm minimization for low-rank tensor completion. In *SDM*, 866–874. SIAM.

Liu, Y.; Shang, F.; Fan, W.; Cheng, J.; and Cheng, H. 2014b. Generalized higher-order orthogonal iteration for tensor decomposition and completion. In *NIPS*, 1763–1771.

Oymak, S.; Jalali, A.; Fazel, M.; Eldar, Y. C.; and Hassibi, B. 2015. Simultaneously structured models with application to sparse and low-rank matrices. *IEEE Transactions on Information Theory* 61(5):2886–2908.

Recht, B. 2011. A simpler approach to matrix completion. *Journal of Machine Learning Research* 12(Dec):3413–3430.

Rettinger, A.; Wermser, H.; Huang, Y.; and Tresp, V. 2012. Context-aware tensor decomposition for relation prediction in social networks. *Social Network Analysis and Mining* 2(4):373–385.

Romera-Paredes, B., and Pontil, M. 2013. A new convex relaxation for tensor completion. In *NIPS*, 2967–2975.

Shang, F.; Liu, Y.; and Cheng, J. 2014. Generalized higher-order tensor decomposition via parallel admm. In *AAAI*, 1279–1285.

Shi, F.; Cheng, J.; Wang, L.; Yap, P.-T.; and Shen, D. 2015. Lrtv: Mr image super-resolution with low-rank and total variation regularizations. *IEEE Transactions on Medical Imaging* 34(12):2459–2466.

Signoretto, M.; Van de Plas, R.; De Moor, B.; and Suykens, J. A. 2011. Tensor versus matrix completion: a comparison with application to spectral data. *IEEE Signal Processing Letters* 18(7):403–406.

Signoretto, M.; Dinh, Q. T.; De Lathauwer, L.; and Suykens, J. A. 2014. Learning with tensors: a framework based on convex optimization and spectral regularization. *Machine Learning* 94(3):303–351.

Tomasi, G., and Bro, R. 2005. Parafac and missing values. *Chemometrics and Intelligent Laboratory Systems* 75(2):163–180.

Tomioka, R.; Hayashi, K.; and Kashima, H. 2010. Estimation of low-rank tensors via convex optimization. *arXiv preprint arXiv:1010.0789*.

Wang, H.; Nie, F.; and Huang, H. 2014. Low-rank tensor completion with spatio-temporal consistency. In *AAAI*, 2846–2852.

Yao, Q., and Kwok, J. T. 2015. Colorization by patch-based local low-rank matrix completion. In *AAAI*, 1959–1965.

Zhao, Q.; Zhang, L.; and Cichocki, A. 2015. Bayesian cp factorization of incomplete tensors with automatic rank determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(9):1751–1763.