TENSOR COMPLETION VIA FUNCTIONAL SMOOTH COMPONENT DEFLATION

Tatsuya Yokota[1] *and Andrzej Cichocki*[1,2]

[1] Brain Science Institute, RIKEN, Saitama, Japan
[2] System Research Institute, Polish Academy of Science (PAN), Poland
email: {yokota,cia}@brain.riken.jp

ABSTRACT

For the matrix/tensor completion problem with very high missing ratio, the standard local (e.g., patch, probabilistic, and smoothness) and global (e.g., low-rank) structure-based methods do not work well. To address this issue, we proposed to use local and global data structures at the same time by applying a novel functional smooth PARAFAC decomposition model for the tensor completion. This decomposition model is constructed as a sum of the outer product of functional smooth component vectors, which are represented by linear combinations of smooth basis functions. A new algorithm was developed by applying greedy deflation and smooth rank-one tensor decomposition. Our extensive experiments demonstrated the high performance and advantages of our algorithm in comparison to existing state-of-the-art methods.

Index Terms— Tensor completion, greedy deflation, smooth component analysis, cosine basis, B-spline basis

1. INTRODUCTION

Tensor is a general name for multi-dimensional arrays such as vectors, matrices, and higher order ones. When we have some incomplete data tensor with missing elements, a 'completion' technique that estimates the missing values by using available values is often used for the recovery or prediction. Image inpainting [1] and recommender system [2] are typical applications of matrix/tensor completion.

In a technical sense, completion is impossible without any assumptions about the relationships between available elements and missing elements. Low-rank-based matrix/tensor completion has been extensively studied, and efficient algorithms have been developed [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. However, when the ratio of missing elements is very high and the data include some noise, using only low-rank assumption is not sufficient to achieve a good performance. In recent years, completion techniques using the low-rank tensor decomposition models with some additional constraints have been proposed to improve performance [15, 16, 17].

Chen et al. [15] proposed a tensor completion algorithm based on the Tucker decomposition model by minimizing the squared errors, nuclear norm of factor matrices, and regularization terms for factor prior. Zhao et al. [16] proposed a tensor completion algorithm based on a PARAFAC decomposition (PD) model based on a Bayesian framework with mixture prior assumption for the factor matrices. The prior assumptions in both methods are similar, which consider the similarity between the individual component vectors in factor matrices. Both methods succeeded in recovering incomplete visual data with relatively high missing ratio in their experiments.

In this paper, we assume that the original tensor is decomposed by the PD model, and its individual component vectors are smooth and represented by linear combinations of a small number of basis functions. In our model, the input data tensor is reconstructed as the sum of several rank-one tensors, which are given by the outer product of smooth component vectors. Since the visual data are locally smooth, the smoothness constraint is really important and helpful for completion in some specific cases. Furthermore, we enforce different levels of smoothness for the different rank-one tensors adaptively, which allows us to enforce strong smoothness into the background and weaker smoothness into the foreground. To implement our model in practice, we employed a greedy deflation approach for the optimization algorithm and applied discrete cosine transform basis and B-spline basis to the basis functions.

2. PROPOSED METHOD

2.1. Functional PARAFAC decomposition model

Basically, the optimization problem of tensor completion can be solved by minimizing $||\boldsymbol{\mathcal{P}}_\Omega \circledast (\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Z}})||_F^2$, where $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is an input data tensor, $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is a parametric model, and $\boldsymbol{\mathcal{P}}_\Omega \in \{0,1\}^{I_1 \times I_2 \times \cdots \times I_N}$ represents missing and available elements by 0 and 1, respectively. We denote $\boldsymbol{\mathcal{X}}_\Omega := \boldsymbol{\mathcal{P}}_\Omega \circledast \boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Z}}_\Omega := \boldsymbol{\mathcal{P}}_\Omega \circledast \boldsymbol{\mathcal{Z}}$ in some cases, where $\circledast$ is the Hadamard product. The functional PARAFAC decomposition (FPD) model is given by

$$\boldsymbol{\mathcal{Z}} = \sum_{r=1}^{R} g_r \boldsymbol{u}_r^{(1)} \circ \boldsymbol{u}_r^{(2)} \circ \cdots \circ \boldsymbol{u}_r^{(N)}, \tag{1}$$

Thanks to JSPS KAKENHI (grant no. 15K16067) for funding.

where $\boldsymbol{u}_r^{(n)} = \boldsymbol{\Phi}^{(n)}\boldsymbol{w}_r^{(n)} \in \mathbb{R}^{I_n}$ are the smooth component vectors, $\boldsymbol{\Phi}^{(n)} = [\boldsymbol{\phi}_1^{(n)}, \boldsymbol{\phi}_2^{(n)}, ..., \boldsymbol{\phi}_{L_n}^{(n)}] \in \mathbb{R}^{I_n \times L_n}$ are the basis functions, $||\boldsymbol{u}_r^{(n)}||_2 = ||\boldsymbol{\phi}_l^{(n)}||_2 = 1$, and $\boldsymbol{w}_r^{(n)} \in \mathbb{R}^{L_n}$ are the weight parameter vectors. We denote unit rank-one tensors by $\boldsymbol{\mathcal{U}}_r := \boldsymbol{u}_r^{(1)} \circ \boldsymbol{u}_r^{(2)} \circ \cdots \circ \boldsymbol{u}_r^{(N)}$.

2.2. Optimization problem

Let us consider the following novel optimization problem:

$$\min_{g_r, \boldsymbol{w}_r^{(n)}} \frac{1}{2}||\boldsymbol{\mathcal{X}}_\Omega - \boldsymbol{\mathcal{Z}}_\Omega||_F^2 + \sum_{r=1}^{R} \frac{g_r^2}{2} \sum_{n=1}^{N} \boldsymbol{w}_r^{(n)T} \boldsymbol{\Lambda}^{(n)} \boldsymbol{w}_r^{(n)}, \tag{2}$$

$$\text{s.t. } \boldsymbol{\mathcal{Z}} = \sum_{r=1}^{R} g_r \boldsymbol{\mathcal{U}}_r, \ \ ||\boldsymbol{\Phi}^{(n)}\boldsymbol{w}_r^{(n)}||_2 = 1,$$

where $\boldsymbol{\Lambda}^{(n)} \in \mathbb{R}^{L_n \times L_n}$ is a suitably designed constraint matrix. The concept of optimization strategy is to fit the FPC model to the data by minimizing the squared error with smoothing regularization terms $\boldsymbol{w}_r^{(n)T}\boldsymbol{\Lambda}^{(n)}\boldsymbol{w}_r^{(n)}$. The challenge in this optimization problem is to find the optimal number of components R. Assuming a decomposition of observed tensor $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{Z}} + \boldsymbol{\mathcal{E}}$ exists and noise tensor $\boldsymbol{\mathcal{E}}$ is independent from the FPD model, we can estimate R as

$$\widehat{R} = \underset{R}{\operatorname{argmin}} f(R) := ||\boldsymbol{\mathcal{X}}_\Omega - \boldsymbol{\mathcal{Z}}_\Omega||_F^2. \tag{3}$$

Since we generally have $f(k+1) \leq f(k)$ for any natural number k, if $f(k) - f(k+1) \leq \varepsilon$ for some small enough value of ε, then such k is an approximately appropriate value for $\widehat{R}$. Thus, we propose a greedy method in Algorithm 1. In this algorithm, we iterate to solve the suboptimization problem as follows:

$$\min_{g, \boldsymbol{w}^{(n)}} \frac{1}{2}||\boldsymbol{\mathcal{E}}_\Omega - g\boldsymbol{\mathcal{U}}_\Omega||_F^2 + \frac{g^2}{2} \sum_{n=1}^{N} \boldsymbol{w}^{(n)T} \boldsymbol{\Lambda} \boldsymbol{w}^{(n)}, \tag{4}$$

$$\text{s.t. } ||\boldsymbol{\Phi}^{(n)}\boldsymbol{w}^{(n)}|| = 1, \ \forall n \in \{1, 2, ..., N\}.$$

while deflating the residual tensor by $\boldsymbol{\mathcal{E}} \leftarrow \boldsymbol{\mathcal{E}} - g\boldsymbol{\mathcal{U}}$.

2.3. Suboptimization algorithm

In this section, we explain how to solve the rank-one FPD problem (4) in Algorithm 1. The error term of the objective function in (4) can be transformed to the following forms:

$$\frac{1}{2}||\boldsymbol{\mathcal{P}}_\Omega \circledast (\boldsymbol{\mathcal{E}} - g\boldsymbol{\Phi}^{(1)}\boldsymbol{w}^{(1)} \circ \boldsymbol{\Phi}^{(n)}\boldsymbol{w}^{(2)} \circ \cdots \circ \boldsymbol{\Phi}^{(N)}\boldsymbol{w}^{(N)})||_F^2$$

$$= \frac{1}{2}||\boldsymbol{\mathcal{E}}_\Omega||_F^2 - g\langle \boldsymbol{\mathcal{E}}_\Omega, \boldsymbol{\mathcal{U}} \rangle + \frac{1}{2}g^2 \langle \boldsymbol{\mathcal{P}}_\Omega, \boldsymbol{\mathcal{U}} \circledast \boldsymbol{\mathcal{U}} \rangle \tag{5}$$

$$= \frac{1}{2}||\boldsymbol{\mathcal{E}}_\Omega||_F^2 - g\boldsymbol{w}^{(n)T}\boldsymbol{\Phi}^{(n)T}[\boldsymbol{E}_\Omega]_{(n)}\boldsymbol{v}^{(n)} \tag{6}$$

$$+ \frac{g^2}{2}\boldsymbol{w}^{(n)T}\boldsymbol{\Phi}^{(n)T}\text{diag}\left([\boldsymbol{P}_\Omega]_{(n)}(\boldsymbol{v}^{(n)} \circledast \boldsymbol{v}^{(n)})\right)\boldsymbol{\Phi}^{(n)}\boldsymbol{w}^{(n)},$$

Algorithm 1 Functional PARAFAC tensor Completion (FPC)

1: **Input**: $\boldsymbol{\mathcal{X}}$, Ω, ε, $R_{\max}$, $\boldsymbol{\Phi}^{(n)}$, $\boldsymbol{\Lambda}^{(n)}$
2: $\boldsymbol{\mathcal{E}}_\Omega \leftarrow \boldsymbol{\mathcal{X}}_\Omega$;
3: $\boldsymbol{\mathcal{Z}} \leftarrow \boldsymbol{0}$;
4: $f(0) \leftarrow ||\boldsymbol{\mathcal{E}}_\Omega||_F^2$;
5: **for** $k = 1, 2, ..., R_{\max}$ **do**
6: Obtain $g\boldsymbol{\mathcal{U}}$ by solving the rank-one FPD optimization (4):
7: $\boldsymbol{\mathcal{E}} \leftarrow \boldsymbol{\mathcal{E}} - g\boldsymbol{\mathcal{U}}$;
8: $\boldsymbol{\mathcal{Z}} \leftarrow \boldsymbol{\mathcal{Z}} + g\boldsymbol{\mathcal{U}}$;
9: $f(k) \leftarrow ||\boldsymbol{\mathcal{E}}_\Omega||_F^2$;
10: **if** $f(k-1) - f(k) \leq \varepsilon$, **then** break; **endif**
11: **end for**
12: **Output**: $\boldsymbol{\mathcal{Z}}$

where $\boldsymbol{\mathcal{E}}_\Omega := \boldsymbol{\mathcal{P}}_\Omega \circledast \boldsymbol{\mathcal{E}}$, $\boldsymbol{v}^{(n)} := \boldsymbol{u}^{(1)} \otimes \boldsymbol{u}^{(2)} \otimes \cdots \otimes \boldsymbol{u}^{(n-1)} \otimes \boldsymbol{u}^{(n+1)} \otimes \cdots \otimes \boldsymbol{u}^{(N)}$ and $\otimes$ denotes the Kronecker product. Note that (5) and (6) are quadratic functions with respect to g and $\boldsymbol{w}^{(n)}$.

Update rules for g and $\boldsymbol{w}^{(n)}$ are derived in Sections 2.3.1 and 2.3.2, and the new optimization scheme is summarized in Algorithm 2.

2.3.1. Update rule of g

From (4) and (5), the optimization problem for scalar g is an unconstrained quadratic problem. Thus, the update rule can be given as

$$g \leftarrow \frac{\langle \boldsymbol{\mathcal{E}}_\Omega, \boldsymbol{\mathcal{U}} \rangle}{\langle \boldsymbol{\mathcal{P}}_\Omega, \boldsymbol{\mathcal{U}} \circledast \boldsymbol{\mathcal{U}} \rangle + \sum_{n=1}^{N} \boldsymbol{w}^{(n)T}\boldsymbol{\Lambda}\boldsymbol{w}^{(n)}}; \tag{7}$$

2.3.2. Update rule of $\boldsymbol{w}^{(n)}$

To update $\boldsymbol{w}^{(n)}$, we formulate the following optimization problem:

$$\min_{\boldsymbol{w} \in \mathbb{R}^{L_n}} h(\boldsymbol{w}) := \frac{1}{2}\boldsymbol{w}^T\boldsymbol{H}^{(n)}\boldsymbol{w} - \boldsymbol{w}^T\boldsymbol{c}^{(n)},$$

$$\text{s.t. } ||\boldsymbol{\Phi}^{(n)}\boldsymbol{w}|| = 1, \tag{8}$$

where $\boldsymbol{H}^{(n)} := g^2\big(\boldsymbol{\Phi}^{(n)T}\text{diag}\left([\boldsymbol{P}_\Omega]_{(n)}(\boldsymbol{v}^{(n)} \circledast \boldsymbol{v}^{(n)})\right)\boldsymbol{\Phi}^{(n)} + \boldsymbol{\Lambda}\big)$, and $\boldsymbol{c}^{(n)} := g\boldsymbol{\Phi}^{(n)T}[\boldsymbol{E}_\Omega]_{(n)}\boldsymbol{v}^{(n)}$. Please note that the problem is symmetric with respect to $\boldsymbol{w}^{(n)}$ for any n, and therefore it can be solved by any technique for the unit-norm constrained quadratic optimization. By employing the gradient-based coefficient normalization method [18], we can solve (8) by applying the following update rules:

$$\boldsymbol{w}_{k+1} \leftarrow \boldsymbol{w}_k - \alpha(\boldsymbol{H}^{(n)}\boldsymbol{w}_k - \boldsymbol{c}^{(n)}); \tag{9}$$

$$\boldsymbol{w}_{k+1} \leftarrow \boldsymbol{w}_{k+1}/||\boldsymbol{\Phi}^{(n)}\boldsymbol{w}_{k+1}||_2; \tag{10}$$

where $\alpha > 0$ is a step-size parameter that is selected so that $h(\boldsymbol{w}_k) \leq h(\boldsymbol{w}_{k+1})$. We repeat iterations (9) and (10) until $h(\boldsymbol{w}_k) - h(\boldsymbol{w}_{k+1}) \leq \varepsilon$.

Algorithm 2 Rank-One Functional PARAFAC Decomposition (FPD)

Input: $\boldsymbol{\mathcal{E}}$, Ω, ε, $\boldsymbol{\Phi}^{(n)}$, $\boldsymbol{\Lambda}^{(n)}$
Initialize $\boldsymbol{w}^{(n)}$ for $n = 1, 2, ..., N$, randomly; k = 0;
Normalize $\boldsymbol{w}^{(n)}$ as unit vectors; calculate g by (7);
$y(1) \leftarrow \frac{1}{2}||\boldsymbol{\mathcal{E}}_\Omega - g\boldsymbol{\mathcal{U}}||_F^2 + \frac{1}{2}g^2 \sum_{n=1}^N \boldsymbol{w}^{(n)T}\boldsymbol{\Lambda}\boldsymbol{w}^{(n)}$;
repeat
  $k \leftarrow k + 1$;
  Update $\boldsymbol{w}^{(n)}$ by solving (8) for all $n \in \{1, 2, ..., N\}$;
  Update g by (7);
  $y(k+1) \leftarrow \frac{1}{2}||\boldsymbol{\mathcal{E}}_\Omega - g\boldsymbol{\mathcal{U}}||_F^2 + \frac{1}{2}g^2 \sum_{n=1}^N \boldsymbol{w}^{(n)T}\boldsymbol{\Lambda}\boldsymbol{w}^{(n)}$;
until $y(k) - y(k+1) \leq \varepsilon$
Output: $g\boldsymbol{\mathcal{U}}$

2.4. Efficient regularization matrix

In this section, we consider the regularization matrix $\boldsymbol{\Lambda}^{(n)}$. When we use the proposed two types of basis functions, the smoothness constraint of $\boldsymbol{u}^{(n)}$ may still be too weak. For example, fitting using only low-frequency cosine functions produces fringe patterns in the reconstructed images and the boundary of the B-spline basis is not continuous when we include zero- or first-order spline functions. To avoid this problem, we propose to apply a quadratic variation (QV) constraint for smoothing as follows:

$$\sum_{i=1}^{I_n-1} |u^{(n)}(i) - u^{(n)}(i+1)|^2 = ||\boldsymbol{L}^{(n)}\boldsymbol{u}^{(n)}||_2^2 = \boldsymbol{w}^{(n)T}\boldsymbol{\Phi}^{(n)T}\boldsymbol{L}^{(n)T}\boldsymbol{L}^{(n)}\boldsymbol{\Phi}^{(n)}\boldsymbol{w}^{(n)}, \quad (11)$$

where a matrix $\boldsymbol{L}^{(n)} \in \mathbb{R}^{(I_n-1)\times I_n}$ is a smoothness constraint matrix typically defined as

$$\boldsymbol{L}^{(n)} := \begin{pmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{pmatrix}. \quad (12)$$

Hence, if we employ the QV constraint, the regularization matrix is given by $\boldsymbol{\Lambda}^{(n)} = \rho\boldsymbol{\Phi}^{(n)T}\boldsymbol{L}^{(n)T}\boldsymbol{L}^{(n)}\boldsymbol{\Phi}^{(n)}$, where ρ is a regularization parameter.

2.5. Improved algorithm

Note that, when applying the proposed greedy deflating algorithm, the resulting output tensor $\boldsymbol{\mathcal{Z}}$ is constructed by adding the $g\boldsymbol{\mathcal{U}}$ term iteratively. Since the values of the missing elements are not included in the objective function, the values of $g\boldsymbol{\mathcal{U}}_{\bar{\Omega}}$ affect the result directly. When some very wrong values are added to $\boldsymbol{\mathcal{Z}}$ during the early stage of deflation, it is not easy to reduce the error in the later stage of deflation because the scale g_r of the latter rank-one tensor could be too small. To prevent such a scenario, we propose to set the regularization parameter ρ to have a large value in the early stage and relatively small values in the later stages. We set ρ_0 and $\rho_{\min}$, and update $\rho \leftarrow \max(\rho_{\min}, \nu\rho)$ in each deflation, where $0 < \nu < 1$ (typically 0.95) controls the rate of decreasing ρ. To implement this idea, we just insert the following update between the fifth and the sixth row in Algorithm 1:

$$\boldsymbol{\Lambda}^{(n)} \leftarrow \max(\nu^{k-1}\rho_0, \rho_{\min})\boldsymbol{\Phi}^{(n)T}\boldsymbol{L}^{(n)T}\boldsymbol{L}^{(n)}\boldsymbol{\Phi}^{(n)}; \quad (13)$$

for all $n \in \{1, 2, ..., N\}$. We call the proposed method as 'robust FPC' (RFPC).

2.6. Convergence

We have established the local convergence property of our algorithm. First, the update rules for g, $\boldsymbol{w}^{(n)}$ in our method do not increase the objective function of (4) because both update rules are based on convex optimizations. Next, we have $f(k+1) \leq f(k)$ in Algorithm 1 by doing the following: First, we can put $f(k) = ||\boldsymbol{\mathcal{E}}_\Omega||_F^2$ and $f(k+1) = ||\boldsymbol{\mathcal{E}}_\Omega - g\boldsymbol{\mathcal{U}}_\Omega||_F^2$. When the initialization for g is given as zero, the objective value at the starting point in Algorithm 2 is given by $||\boldsymbol{\mathcal{E}}_\Omega||_F^2$. Since the objective function does not increase after updating g and $\boldsymbol{w}^{(n)}$, we have

$$f(k) = ||\boldsymbol{\mathcal{E}}_\Omega||_F^2 \geq ||\boldsymbol{\mathcal{E}}_\Omega - g\boldsymbol{\mathcal{U}}_\Omega||_F^2 + g^2 \sum_{n=1}^N \boldsymbol{w}^{(n)T}\boldsymbol{\Lambda}\boldsymbol{w}^{(n)} \geq ||\boldsymbol{\mathcal{E}}_\Omega - g\boldsymbol{\mathcal{U}}_\Omega||_F^2 = f(k+1), \quad (14)$$

taking into account that $g^2 \sum_{n=1}^N \boldsymbol{w}^{(n)T}\boldsymbol{\Lambda}\boldsymbol{w}^{(n)} \geq 0$.

3. EXPERIMENTS

Ten benchmark (256x256x3) RGB color images shown in Fig. 1(A) were used in our experiments. First, we applied the proposed algorithms of FPC and RFPC to the 'Lena' image with various parameter settings for $\rho \in \{0, 1, 10, 10^2, 10^3, 10^4\}$ and a number of basis functions for #dimension $\in \{2^3, 2^4, 2^5, 2^6, 2^7, 2^8\}$. Fig. 1(B) shows the results with #dimension $= 128$ from the incomplete image with 90% missing ratio, and Fig. 1(C) shows the results of the peak signal-to-noise ratio (PSNR) for all parameter settings. We can see that the performance of RFPC was more accurate than that of FPC with all settings of ρ for all settings of #dimension, where we set $\rho_0 = 1000$, $\rho_{\min} = 1$, and $\nu = 0.95$ in RFPC. Next, we applied the state-of-the-art algorithms for matrix and tensor completion of LTVNN[1] [19], HaLRTC[2] [5], STDC[3] [15], and FBCP-MP[4] [16] for all our benchmark images with various missing ratios {60%, 70%,

[1] Linear Total Variation Approximate Regularized Nuclear Norm
[2] High Accuracy Low-Rank Tensor Completion
[3] Simultaneous Tensor Decomposition and Completion
[4] Fully Bayesian CANDECOMP/PARAFAC tensor completion with Mixture Prior

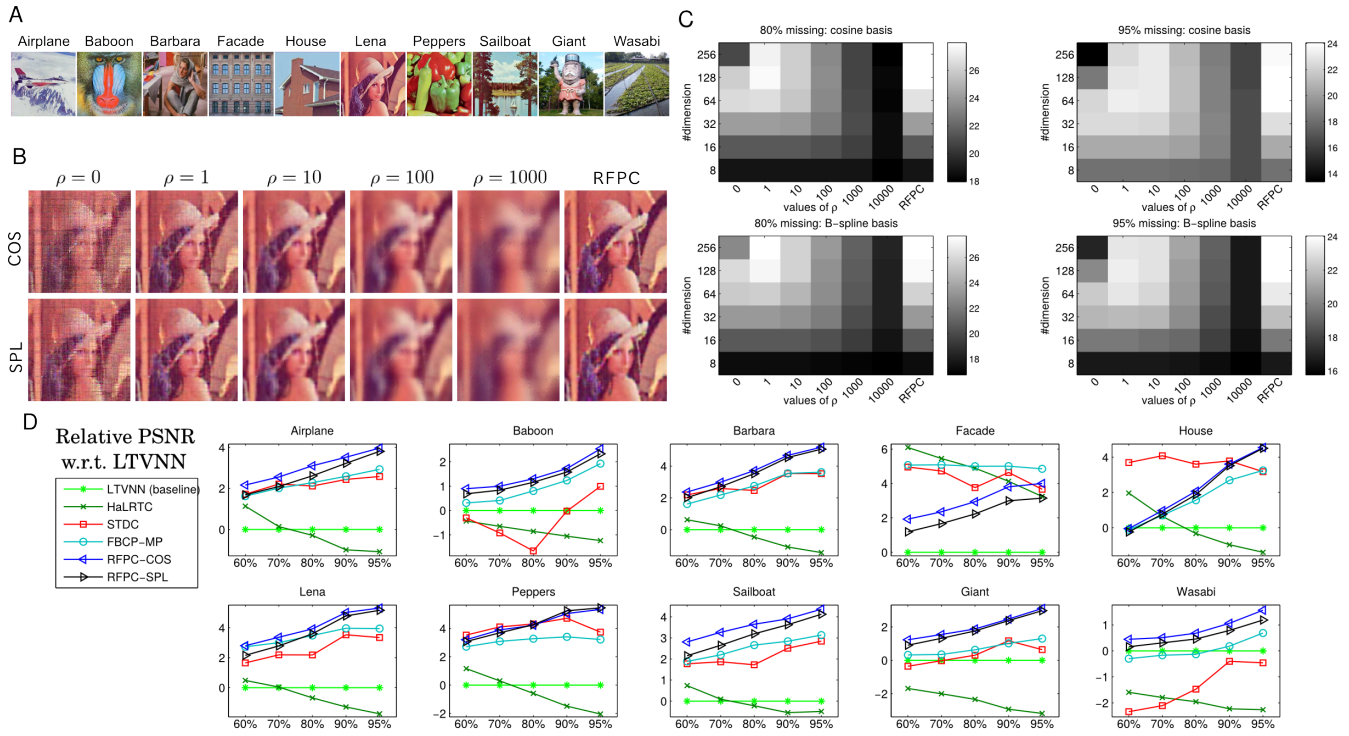

Fig. 1. Summary of experimental results: (A) test images, (B) completion results of various ρ's and RFPC, (C) PSNR evaluation of the proposed methods for various settings, and (D) comparison with existing state-of-the-art methods.

80%, 90%, 95%} (See Fig. 1(D)). Since 'Facade' and 'House' contain some non-smooth parts, the smoothness constraints produced some small errors of over-smoothing; however, the proposed algorithms outperformed all the existing methods significantly in the other eight images by a large margin.

4. DISCUSSION: NOVELTY AND EFFECTIVENESS

In Refs. [20, 21], models of FPD with Fourier series and B-spline basis functions were proposed. A penalized smooth PARAFAC decomposition model was proposed in [22, 23]. However, these algorithms cannot be applied for tensor completion with missing elements. In contrast, our model is designed for the completion problem to enforce the smoothness constraints by combining the functional and penalized approaches at the same time. A functional model reduces the number of parameters of a PD model, and the penalty term guarantees the smoothness of the component vectors, depending on the value of ρ. This allows us to adjust adaptively the smoothness levels by multiplying the scaling parameter g_r^2 and ρ into the smoothness constraint terms.

The LTVNN algorithm [19] imposes the total variation penalty into an 'output matrix itself.' In Refs. [15, 16], some penalties based on prior information are applied; however, they are based on a different kind of assumptions, i.e., the individual component vectors are similar. For these reasons, our model and optimization algorithms are quite different from those described in the literature.

Our methods have two kinds of efficiency. First, our FPD model allows us to reduce the number of model parameters by L_n/I_n. This property is really helpful for dealing with a very large-scale data tensor. Second, the adaptive smoothness of our model plays a very important role in outperforming the other existing methods. In the case of completion with very high missing ratio, the uniqueness of the solution would be quite low. Since the values of the missing elements are constructed by adding the corresponding values of $g\boldsymbol{\mathcal{U}}$, the first term $g\boldsymbol{\mathcal{U}}$ with a large g affects the results significantly. Note that an FPD model with a large ρ gives a good overview of the approximation (see Fig. 1(B)); it is therefore a good idea to extract the smooth components first.

5. CONCLUSIONS

In this paper, we proposed an efficient algorithm for smooth tensor completion by applying functional and adaptive penalized smoothness. This algorithm has the advantages of computational cost and higher performance at the same time and also guarantees local convergence. The proposed RFPC algorithm outperformed the existing state-of-the-art algorithms by a large margin, which was confirmed by our extensive experiments.

6. REFERENCES

[1] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[2] J.D.M. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proceedings of the 22nd International Conference on Machine Learning*. ACM, 2005, pp. 713–719.

[3] E. J. Candes and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[4] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Problems*, vol. 27, no. 2, 2011.

[5] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2013.

[6] J.A. Bazerque, G. Mateos, and G. Giannakis, "Rank regularization and Bayesian inference for tensor completion and extrapolation," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5689–5703, 2013.

[7] B. Romera-Paredes and M. Pontil, "A new convex relaxation for tensor completion," in *Advances in Neural Information Processing Systems*, 2013, pp. 2967–2975.

[8] H. Tan, B. Cheng, W. Wang, Y.-J. Zhang, and B. Ran, "Tensor completion via a multi-linear low-n-rank factorization model," *Neurocomputing*, vol. 133, pp. 161–169, 2014.

[9] M. Mardani, G. Mateos, and G.B. Giannakis, "Subspace learning and imputation for streaming big data matrices and tensors," *arXiv preprint arXiv:1404.4667*, 2014.

[10] D. Kressner, M. Steinlechner, and B. Vandereycken, "Low-rank tensor completion by Riemannian optimization," *BIT Numerical Mathematics*, vol. 54, no. 2, pp. 447–468, 2014.

[11] Y. Liu, F. Shang, W. Fan, J. Cheng, and H. Cheng, "Generalized higher-order orthogonal iteration for tensor decomposition and completion," in *Advances in Neural Information Processing Systems*, 2014, pp. 1763–1771.

[12] M. Filipović and A. Jukić, "Tucker factorization with missing data with application to low-n-rank tensor completion," *Multidimensional Systems and Signal Processing*, pp. 1–16, 2014.

[13] N. Vervliet, O. Debals, L. Sorber, and L.D. Lathauwer, "Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 71–79, 2014.

[14] Y. Yang, Y. Feng, and J.A.K. Suykens, "A rank-one tensor updating algorithm for tensor completion," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1633–1637, 2015.

[15] Y.L. Chen, C.T. Hsu, and H.Y.M. Liao, "Simultaneous tensor decomposition and completion using factor priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 577–591, 2014.

[16] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian CP factorization of incomplete tensors with automatic rank determination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1751–1763, 2015.

[17] T. Yokota, Q. Zhao, and A. Cichocki, "Smooth PARAFAC decomposition for tensor completion," *arXiv preprint arXiv:1505.06611*, 2015.

[18] S.C. Douglas, S. Amari, and S.Y. Kung, "On gradient adaptation with unit-norm constraints," *IEEE Transactions on Signal Processing*, vol. 48, no. 6, pp. 1843–1847, 2000.

[19] X. Han, J. Wu, L. Wang, Y. Chen, L. Senhadji, and H. Shu, "Linear total variation approximate regularized nuclear norm optimization for matrix completion," *Abstract and Applied Analysis*, vol. ID 765782, 2014.

[20] M.M. Reis and M. Ferreira, "PARAFAC with splines: A case study," *Journal of Chemometrics*, vol. 16, no. 8-10, pp. 444–450, 2002.

[21] M.E. Timmerman and H.A.L. Kiers, "Three-way component analysis with smoothness constraints," *Computational Statistics & Data Analysis*, vol. 40, no. 3, pp. 447–470, 2002.

[22] E. Martinez-Montes, J.M. Sanchez-Bornot, and P.A. Valdes-Sosa, "Penalized PARAFAC analysis of spontaneous EEG recordings," *Statistica Sinica*, vol. 18, pp. 1449–1464, 2008.

[23] G. Allen, "Multi-way functional principal components analysis," in *IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 2013, pp. 220–223.