Low-rank and Smooth Tensor Recovery on Cartesian Product Graphs

***Abstract*—Data science research has found great success with algorithms that leverage the structure of the topological space that the high-dimensional data lies on. In particular, low-rank tensor models which represent low-dimensional latent factors in a succinct and parsimonious way have become indispensable tools. These low-rank models have been utilized in a variety of applications including tensor completion from corrupted or missing entries. In the standard tensor completion problem, the different modes of the tensor are assumed to be completely independent of each other. However, in many real-world problems such as those involving spatio-temporal data, there exist relationships between the different modes. This information can be encoded in terms of graphs which can bring additional structure to the tensor completion problem. In this paper, we introduce methods for structured tensor completion where both the low-rank and smoothness of tensor are incorporated into the optimization problem. In particular, we model tensor data as graph signals on Cartesian product graphs and use the Dirichlet energy to quantify the smoothness of tensor data with respect to the graph. We evaluate the performance of this tensor recovery approach for different types of data, i.e. low-rank, smooth and low-rank plus smooth, and compare with existing methods.**

I. Introduction

Higher-order tensors preserve the complicated innate multi-way structural properties of data [1]. Tensor completion, which is a missing data imputation task based on the observed data, has attracted significant amounts of research in areas such as image processing and machine learning [2], [3].

Since tensor completion is an underdetermined problem, it is usually necessary to constrain the problem by imposing some structure. The most commonly used structural constraint is the low rank assumption which restricts the degree of freedoms of the missing entries. Since a tensor has different types of rank definitions, various low rank tensor completion approaches have been developed corresponding to the different norm definitions [4], [2], [5], [6]. The resulting low-rank minimization problems have been extended using tools from robust data analysis which plays an instrumental role in dealing with outliers in the completion problem [7]. Drawing upon the advances in robust PCA analysis [8], robust tensor completion is to complete a tensor $\mathcal{Y}$ by separating it into a low-rank part, $\mathcal{X}$, plus a sparse part, $\mathcal{S}$, to capture the different noise patterns. The completion problem could be addressed by the joint minimization of trace norm and ℓ_1 norm.

While the robust low-rank tensor representations capture the global structure of tensor data, they do not preserve the local geometric structure and capture the relationship between modes. Manifold learning addresses this issue and has been successfully implemented for tensors [9], [10]. However, current manifold learning approaches typically focus on only one mode of the data. Yet for many data matrices and tensors, correlations exist across all modalities. Several recent papers [11], [12], [13], [14], [15], [16], [17] exploit this coupled relationship to co-organize matrices and infer underlying row and column embeddings.

In this paper, we introduce a robust tensor recovery approach that simultaneously takes both the low-rank and geometric structure into account. While the low-rank structure is captured through nuclear norm minimization, the geometric structure is captured by minimizing the total variation norm (Dirichlet energy) of the tensor data with respect to the underlying product graph structure. Taking graph structure into account while processing data has well-documented merits for several signal processing and machine learning tasks like sampling, denoising and clustering [18], [19]. The graph structure can help us to take into account any auxiliary information available such as the location of sensors in a sensor network. In this paper, we present the optimization problems corresponding to the different assumptions, i.e. low-rank, smooth and low-rank+smooth tensor recovery. The resulting algorithms are evaluated on simulated and real data.

II. Background

A. Tensor Notation and Algebra

A multidimensional array with N modes $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is called a tensor, where $x_{i_1,i_2,\ldots,i_N}$ denotes the $(i_1, i_2, \ldots, i_N)$th element of the tensor $\mathcal{X}$. Vectors obtained by fixing all indices of the tensor except the one that corresponds to nth mode are called mode-n fibers.

Tensor unfolding: Process of reordering the elements of the tensor into a matrix is known as matricization or unfolding. The mode-n matricization of tensor $\mathcal{X}$ is denoted as $X_{(n)}$ and is obtained by arranging mode-n fibers to be the columns of the resulting matrix.

Tensor Rank: Unlike matrices, which have a unique definition of rank, there are multiple rank definitions for tensors including *tensor rank* and *tensor n-rank*. In this paper, we will focus on the *n-rank* of $\mathcal{X}$ defined as the collection of ranks of mode matrices $X_{(n)}$:

$$n\text{-rank}(\mathcal{X}) = \left(\text{rank}(X_{(1)}),\ \text{rank}(X_{(2)}), \ldots,\ \text{rank}(X_{(N)})\right).$$

B. Cartesian Product Graphs

Consider N graphs, $G_k = (V_k, E_k, W^k)$, corresponding to each mode of an N-mode tensor each with I_k nodes. The graph Laplacian matrix L_k associated with the graph G_k is

defined as $L_k = diag[W_k \mathbf{1}_{I_k}] - W_k$, where the first term is a diagonal matrix of the degrees of the graph. The Cartesian product graph is defined as $G = G_1 \oplus G_2 \oplus \ldots \oplus G_N$, where $V = V_1 \times V_2 \times \ldots \times V_N$. The Laplacian of this product graph can be written as the Kronecker sum of the individual Laplacians as $L_{G_1 \oplus G_2 \oplus \ldots \oplus G_N} = L_1 \oplus L_2 \oplus \ldots \oplus L_N$ [15].

The smoothness promoting Dirichlet energy of the data tensor, $\mathcal{X}$ with respect to the Laplacian of the product graph can be explicitly expressed in terms of its graph factors as $\sum_{i=1}^{N} tr(X_{(i)}^T L_i X_{(i)})$ [20].

C. Higher Order RPCA

For two-way data, robust PCA (RPCA) was introduced to address the limitations of PCA against outliers and non-Gaussian errors [8]. In this approach, a given data matrix is decomposed into a low-rank plus a sparse model:

$$min_{X,S}\{rank(X) + \lambda ||S||_0 | X + S = Y\}. \tag{1}$$

Goldfarb and Qin [7] have extended RPCA to higher-order data, i.e. tensors. In higher-order RPCA (HoRPCA), (1) is modified by replacing the nuclear rank of a matrix by the Tucker rank (Trank) of a tensor. Similar to RPCA, Trank and l_0 norms are replaced with their convex counterparts CTrank and l_1 norm yielding

$$min_{\mathcal{X},\mathcal{S}}\{CTrank(\mathcal{X}) + \lambda ||\mathcal{S}||_1 | \mathcal{X} + \mathcal{S} = \mathcal{Y}\}. \tag{2}$$

Goldfarb and Qin [7] proposed various models to solve this optimization problem. One such model is the Singleton model, which estimates the nuclear norm of the tensor as the sum of the nuclear norms of the mode-n matricizations of the tensor yielding

$$min_{\mathcal{X},\mathcal{S}}\{\sum_{i=1}^{N} ||X_{(n)}||_* + \lambda ||\mathcal{S}||_1 | \mathcal{X} + \mathcal{S} = \mathcal{Y}\}. \tag{3}$$

III. Low-Rank Plus Smooth Tensor Recovery on Cartesian Product Graphs

A. Smooth Tensor Recovery on Cartesian Product Graphs (SmoothTR)

First, we consider the problem of tensor reconstruction from corrupted observations with possibly missing entries with the assumption that the underlying tensor is smooth or low-frequency with respect to the product graph. The notion of smoothness is quantified by the Dirichlet energy of the tensor. Our goal is to decompose the observed tensor $\mathcal{Y}$ into a smooth (low-frequency) part, $\mathcal{X}$, and a sparse part, $\mathcal{S}$, that corresponds to the outliers. The existence of missing elements in the observed data is taken into account through a projection operator, $\mathcal{P}_\Omega[\mathcal{Y}]$, where $\Omega \in \{1, \ldots, I_1\} \times \{1, \ldots, I_2\} \times \cdots \times \{1, \ldots, I_N\}$ denotes the indices of the observed entries. This goal can be formulated through:

$$\min_{\mathcal{X},\mathcal{S}} \sum_{i=1}^{N} \alpha_i tr(X_{(i)}^\top L_i X_{(i)}) + \lambda \|\mathcal{S}\|_1 \text{ s.t. } \mathcal{P}_\Omega[\mathcal{X} + \mathcal{S}] = \mathcal{P}_\Omega[\mathcal{Y}], \tag{4}$$

where α_i is the weight of the smoothness assigned to each mode and λ is the regularization parameter controlling the sparsity level. This problem can be rewritten by by applying variable-splitting to $\mathcal{X}$ and introducing N auxiliary variables $\mathcal{X}_1, \ldots, \mathcal{X}_N$ and defining another auxiliary variable $\mathcal{Z} = \mathcal{X}_i$.

$$\begin{aligned} \min_{\mathcal{X}_i,\mathcal{Z},\mathcal{S}} \sum_{i=1}^{N} \alpha_i tr(X_{i_{(i)}}^\top L_i X_{i_{(i)}}) + \lambda \|\mathcal{S}\|_1, \\ \text{s.t. } \mathcal{P}_\Omega[\mathcal{Z} + \mathcal{S}] = \mathcal{P}_\Omega[\mathcal{Y}] \\ \mathcal{Z} = \mathcal{X}_i, i = 1, 2, \ldots, N. \end{aligned} \tag{5}$$

This problem can be solved through two block ADMM algorithm, where the Augmented Lagrangian of the problem is,

$$\mathcal{L}(\mathcal{Z}, \mathcal{S}, \mathcal{X}_i, \Lambda, \Lambda_i) = \sum_{i=1}^{N} \alpha_i tr(X_{i_{(i)}}^\top L_i X_{i_{(i)}}) + \lambda \|S\|_1$$
$$+ \frac{\rho}{2} \|\mathcal{P}_\Omega[\mathcal{Z} + \mathcal{S} - \mathcal{Y} + \frac{\Lambda}{\rho}]\|_F^2 + \frac{\rho}{2} \sum_{i=1}^{N} \|\mathcal{Z} - \mathcal{X}_i + \frac{\Lambda_i}{\rho}\|_F^2.$$

The update equations for the different variables are then derived as follows.

$\mathcal{X}_i$ **update:**

$$\mathcal{X}_{i_{(i)}}^{k+1} = (\alpha_i L_i + \rho I)^{-1} (\rho \mathcal{Z}_{(i)}^k - \Lambda_{i_{(i)}}^k) \text{ for i} = 1, \ldots, \text{N}.$$

$\mathcal{S}$ **update:**

$$\mathcal{P}_\Omega[\mathcal{S}^{k+1}] = \mathcal{P}_\Omega \left[\mathcal{T}_{\lambda/\rho} \Big(\mathcal{Y} - \mathcal{Z}^k - \frac{\Lambda^k}{\rho} \Big) \right],$$

where $\mathcal{T}$ is the soft-thresholding operator.

$\mathcal{Z}$ **update:**

$$\mathcal{P}_\Omega[\mathcal{Z}^{k+1}] = \mathcal{P}_\Omega \left[\frac{1}{N+1} \Big(\sum_{i=1}^{N} (\mathcal{X}_i^{k+1} + \frac{\Lambda^k}{\rho}) + \mathcal{Y} - \mathcal{S}^{k+1} - \frac{\Lambda^k}{\rho} \Big) \right],$$

$$\mathcal{P}_\Omega^\perp[\mathcal{Z}^{k+1}] = \mathcal{P}_\Omega^\perp \left[\frac{1}{N} \Big(\sum_{i=1}^{N} (\mathcal{X}_i^{k+1} + \frac{\Lambda^k}{\rho}) \Big) \right].$$

Dual Updates:

$$\Lambda^{k+1} = \Lambda^k + \rho(\mathcal{Z} + \mathcal{S} - \mathcal{Y}),$$
$$\Lambda^{k+1} = \Lambda_i^k + \rho(\mathcal{Z} - \mathcal{X}_i) \text{ for i} = 1, ..., \text{N}.$$

B. Smooth and Low-rank Tensor Recovery on Cartesian Product Graphs (GRHoRPCA)

In the second problem, we focus on simultaneous robust low-rank and smooth tensor recovery. This goal can be achieved by minimizing the following optimization problem:

$$\begin{aligned} \min_{\mathcal{X},\mathcal{Y},\mathcal{S}} \sum_{i=1}^{N} \alpha_i tr(X_{(i)}^\top L_i X_{(i)}) + \psi_i \|X_{(i)}\|_* + \lambda_1 \|S\|_1, \\ \mathcal{P}_\Omega[\mathcal{X} + \mathcal{S}] = \mathcal{P}_\Omega[\mathcal{Y}], \end{aligned} \tag{6}$$

where ψ_i is the weight of the nuclear norm across the ith mode. Similar to the formulation in the previous section, we will introduce auxiliary variables for each mode of the tensor,

$\mathcal{Z}$ and $\mathcal{W}$, to disentangle the different terms in the optimization such that $\mathcal{Z} = \mathcal{X}_i = \mathcal{W}_i$ and rewrite the problem as:

$$\min_{\{\mathcal{X}_i,\mathcal{W}_i\},\mathcal{Z},\mathcal{S}} \sum_{i=1}^{N} \alpha_i tr(X_{i_{(i)}}^\top L_i X_{i_{(i)}}) + \psi_i \|\mathcal{W}_{i_{(i)}}\|_* + \lambda_1 \|\mathcal{S}\|_1,$$
$$\mathcal{P}_\Omega[\mathcal{Z} + \mathcal{S}] = \mathcal{P}_\Omega[\mathcal{Y}],$$
$$\mathcal{Z} = \mathcal{X}_i = \mathcal{W}_i. \tag{7}$$

This problem can be solved through two block ADMM algorithm, where the Augmented Lagrangian of the problem is divided into two blocks, i.e., $\{\mathcal{W}_i, \mathcal{X}_i, \mathcal{S}\}$ and $\{\mathcal{Z}\}$, where each variable in a block is separable with dual variables $\Gamma_i, \Lambda_i, \Lambda$ respectively.

$$\mathcal{L}(\mathcal{Z}, \mathcal{S}, \mathcal{X}_i, \mathcal{W}_i, \Lambda, \Lambda_i, \Gamma_i) = \sum_{i=1}^{N} \Big(\alpha_i tr(X_{i_{(i)}}^T L_i X_{i_{(i)}})$$
$$+ \psi_i \|W_{(i)}\|_* + \lambda_1 \|\mathcal{S}\|_1 \Big) + \frac{\rho}{2} \big\| \mathcal{P}_\Omega[\mathcal{Z} + \mathcal{S} - \mathcal{Y} + \frac{\Lambda}{\rho}] \big\|_F^2$$
$$+ \sum_{i=1}^{N} \Big(\frac{\rho}{2} \|\mathcal{Z} - \mathcal{X}_i + \frac{\Lambda_i}{\rho}\|_F^2 + \frac{\rho}{2} \|\mathcal{Z} - \mathcal{W}_i + \frac{\Gamma_i}{\rho}\|_F^2 \Big). \tag{8}$$

The update steps for each variable are,
$\mathcal{W}_i$ **update:**

$$W_{i_{(i)}}^{k+1} = \mathcal{D}_{\psi_i/\rho}(Z_{(i)} + \frac{\Gamma_{i_{(i)}}}{\rho}) \text{ for i} = 1, ..., \text{N},$$

where $\mathcal{D}$ is the proximal operator corresponding to singular value thresholding.
$\mathcal{X}_i$ **update:**

$$X_{i_{(i)}}^{k+1} = (\alpha_i L_i + \rho I)^{-1} (\rho \mathcal{Z}_{(i)}^k - \Lambda_{i_{(i)}}^k) \text{ for i} = 1, ..., \text{N}.$$

$\mathcal{S}$ **update:**

$$\mathcal{P}_\Omega[\mathcal{S}^{k+1}] = \mathcal{P}_\Omega \left[\mathcal{T}_{\lambda_1/\rho}(\mathcal{Y} - \mathcal{Z}^k - \frac{\Lambda^k}{\rho}) \right].$$

$\mathcal{Z}$ **update:**

$$\mathcal{P}_\Omega[\mathcal{Z}^{k+1}] = \mathcal{P}_\Omega \Big[\sum_{i=1}^{N} \Big(\mathcal{X}_i^{k+1} + \mathcal{W}_i^{k+1} - \frac{\Lambda_i^k + \Gamma_i^k}{\rho} \Big)$$
$$+ \mathcal{Y} - \mathcal{S}^{k+1} - \frac{\Lambda^k}{\rho} \Big],$$

$$\mathcal{P}_\Omega^\perp[\mathcal{Z}^{k+1}] = \frac{1}{2N} \mathcal{P}_\Omega^\perp \Big[\sum_{i=1}^{N} (\mathcal{X}_i^{k+1} + \mathcal{W}_i^{k+1} - \frac{\Lambda_i^k + \Gamma_i^k}{\rho}) \Big].$$

Dual updates:

$$\mathcal{P}_\Omega[\Lambda^{k+1}] = \mathcal{P}_\Omega[\Lambda^k + \rho(\mathcal{Z}^{k+1} + \mathcal{S}^{k+1} - \mathcal{Y})],$$

$$\Lambda_i^{k+1} = \Lambda_i^k + \rho(\mathcal{Z}^{k+1} - \mathcal{X}_i^{k+1}) \text{ for i} = 1, ..., \text{N},$$

$$\Gamma_i^{k+1} = \Gamma_i^k + \rho(\mathcal{Z}^{k+1} - \mathcal{W}_i^{k+1}) \text{ for i} = 1, ..., \text{N}.$$

IV. Results

A. Simulated Data

The performance of the proposed optimization algorithms was first evaluated for different types of simulated tensor data with respect to HoRPCA.

Within each experimental setting, we generated three different types of data: Smooth Data, Low-rank and Smooth Data, Low-rank Data. Smooth tensor data is generated by randomly generating three factor graphs, G_1, G_2, G_3 with number of nodes equal to n_1, n_2, and n_3, respectively, corresponding to the three modes of the tensor data. We then constructed the a Cartesian product , G_{PG}, whose Laplacian matrix $L_{PG} \in \mathbb{R}^{n_1 n_2 n_3 \times n_1 n_2 n_3}$ is used to construct a filter $h(L_{PG})$ with heat kernel as in [21]. The smooth tensor data $\mathcal{Y}$ in its vectorized form is $vec(\mathcal{Y}) = h(L_{PG}) x_0 \in \mathbb{R}^{n_1 n_2 n_3}$ where $x_0 \in \mathbb{R}^{n_1 n_2 n_3}$ is a random graph signal with entries from standard normal distribution. The filter $h(L)$ is defined as $h(L) = U[\sum_{i=1}^{n_1 n_2 n_3} exp(-\alpha \lambda_i)] U^\top$ where U is the eigenvector matrix of L and λ_is are the eigenvalues.

Low-rank tensors are constructed by randomly generating three orthonormal mode factor matrices $A_i \in \mathbb{R}^{n_i \times r_i}$ and a random core tensor $\mathcal{C} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ whose entries are drawn from the standard normal distribution with dimensions corresponding to the n-rank of the data. Low-rank data is then generated by mode-n products as $\mathcal{Y} = \mathcal{C} \times_1 A_1 \times_2 A_2 \times_3 A_3$.

Low-rank and smooth data is generated by combining these two concepts. We first generate a low-rank tensor and filter it with respect to the Cartesian product graph as described above.

Experiment 1: In the first experiment, we created three-mode tensors $\mathcal{Y} \in \mathbb{R}^{12 \times 12 \times 12}$ that are low-rank, approximately low-rank and smooth and only smooth with $r_1 = r_2 = r_3 = 5$. In this first setting, we set $\mathcal{S} = 0$, i.e., there are no outliers added to the tensor. We varied the ratio of observed entries from 10% to 90%. In order to generate smooth tensor data and approximately low-rank smooth data, we generated three connected random factor graphs using Erdős-Rényi model with edge probability $p = 0.15$ each with 12 nodes. Factor graphs are then used to produce a Cartesian product graph, which is used as a graph filter with smoothness regularization parameter $\alpha = 20$. The optimal values of the hyperparameters are found by searching over the grid for the best relative error. This experiment was repeated for 10 random product graphs and the mean relative error for the three data types and three methods are shown in Fig. 1. From this figure, it can be seen that when the data is purely low-rank HoRPCA performs the best with graph regularized HoRPCA peforming similar to it. When the data is smooth, HoRPCA does not perform well and GRHoRPCA and SmoothTR perform better as expected.

B. Real Data

COIL-20 dataset consists of grayscale pictures of 20 different objects taken from different views by rotating the objects every $5°$. The original pictures were downsampled to 48×48 and sampled every every four views. From this downsampled

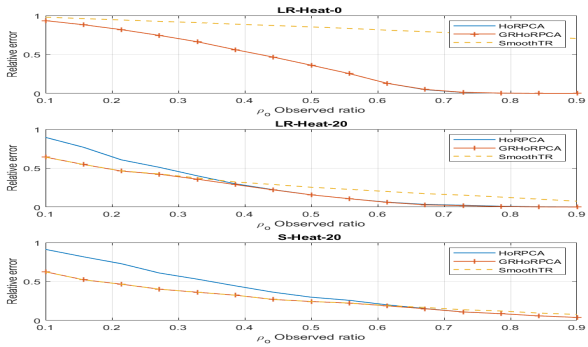

Fig. 1. Relative Reconstruction Error vs. Observed Ratio: (Top) Low-Rank Tensor; (Middle) Approximately Low-Rank and Smooth Tensor; (Bottom) Smooth Tensor

data, we construct a tensor $\mathcal{Y} \in \mathbb{R}^{20\times18\times48\times48}$, where 20 is the number of objects, 18 is the number of views and the last two modes correspond to the size of the images. We construct k-NN graphs across views, rows and columns. The similarity graph for the views is constructed using a 3-NN graph with Gaussian kernel, where σ is chosen proportionately to the Frobenius norm of $\mathcal{Y}$. The row and column graphs are constructed with 2-NN graphs with binary edges using Euclidean distance. In this experiment, we compare the reconstruction performances of the GRHoRPCA, HoRPCA and SmoothTR algorithms by varying the percentage of observed entries from 10% to 100%.

The smoothness parameters $(\alpha_2, \alpha_3, \alpha_4)$ are set as $\gamma \cdot (0.05, 1, 1)$. The weights for the nuclear norms are set to $\psi_i = p \cdot tr(cov(Y_{(i)})$ where p is selected such that $max(\{\psi_i\}) = 1$. For a fair comparison, we ran grid search on the hyperparameters γ and λ_1 for all three algorithms and report the results in Fig. 2. It can be seen from this figure, similar to the simulated smooth tensor, the performances of GRHoRPCA and SmoothTR are close to each other with HoRPCA performing much worse. This is due to the fact that most real data are not exactly low-rank across all modes of the tensor. In

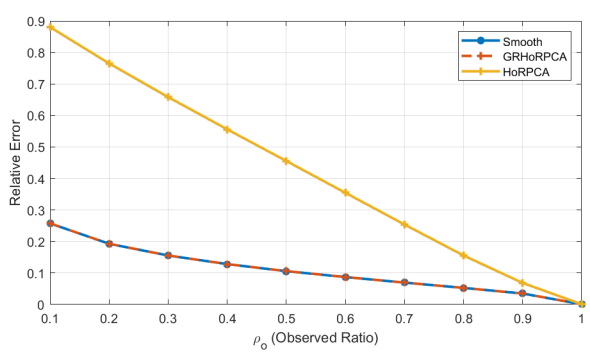

Fig. 2. Reconstruction error vs. observed entry ratio for Coil-20

Fig. 3, we illustrate a reconstructed object from COIL-20 with 60% missing data and impulsive noise. It can be seen that the quality of the reconstructed image is better for SmoothTR and GRHoRPCA compared to HoRPCA. While SmoothTR and GRHoRPCA perform similar to each other, the run time for SmoothTR (48 sec) is less than the run time for GRHoRPCA (129 sec).

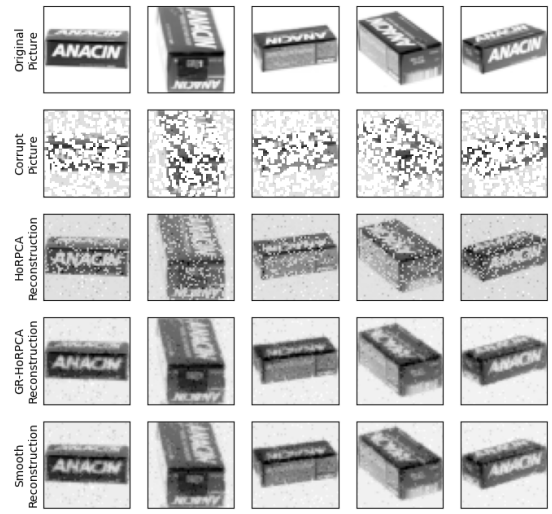

Fig. 3. An example of a reconstructed object from COIL-20 dataset with 60% of its entries missing and with impulsive noise drawn from i.i.d. uniform distribution $U(-70, 70)$ added to randomly selected 10% of the pixels.

V. Conclusions

In this paper, we introduced two tensor recovery algorithms incorporating the geometric structure of tensors. The first algorithm recovers tensors with the assumption that the tensor is smooth with respect to the underlying product graph. This assumption is equivalent to low pass graph filtering the data with respect to the underlying product graph structure. The second algorithm takes into account both the low-rank and geometric structure of the tensor by minimizing the tensor rank while simultaneously minimizing the Dirichlet energy, resulting in robust tensor recovery on graphs. The proposed algorithms are tested on different types of simulated and real data under different noise and missing data conditions. The results imply that when the underlying tensor is low-rank across all modes, HoRPCA performs the best. However, for real data where the low-rankness across each mode is hard to satisfy, SmoothTR and GRHoRPCA are superior to standard HoRPCA. Moreover, SmoothTR is faster compared to GRHoRPCA and HoRPCA as it does not require nuclear norm minimization.

In the proposed work, we assumed that the product graph underlying the observed tensor data was known a priori or can be constructed using k-NN. While this assumption is valid for certain applications, in a lot of applications the underlying graph structure is not known. For this reason, future work will consider simultaneous graph learning and robust tensor recovery similar to recent work for matrix based data [20].

REFERENCES

[1] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[2] 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, 2012.

[3] Q. Song, H. Ge, J. Caverlee, and X. Hu, "Tensor completion algorithms in big data analytics," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 13, no. 1, pp. 1–48, 2019.

[4] G. Tomasi and R. Bro, "Parafac and missing values," *Chemometrics and Intelligent Laboratory Systems*, vol. 75, no. 2, pp. 163–180, 2005.

[5] J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do, "Efficient tensor completion for color image and video recovery: Low-rank tensor train," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2466–2479, 2017.

[6] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, T.-Y. Ji, and T.-H. Ma, "Tensor n-tubal rank and its convex relaxation for low-rank tensor recovery," *Information Sciences*, vol. 532, pp. 170–189, 2020.

[7] D. Goldfarb and Z. Qin, "Robust low-rank tensor recovery: Models and algorithms," *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 1, pp. 225–253, 2014.

[8] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 1–37, 2011.

[9] Y. Su, X. Bai, W. Li, P. Jing, J. Zhang, and J. Liu, "Graph regularized low-rank tensor representation for feature selection," *Journal of Visual Communication and Image Representation*, vol. 56, pp. 234–244, 2018.

[10] S. E. Sofuoglu and S. Aviyente, "Graph regularized tensor train decomposition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3912–3916.

[11] M. Gavish and R. R. Coifman, "Sampling, denoising and compression of matrices by coherent matrix organization," *Applied and Computational Harmonic Analysis*, vol. 33, no. 3, pp. 354–369, 2012.

[12] N. Shahid, V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, "Robust principal component analysis on graphs," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2812–2820.

[13] V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, "Matrix completion on graphs," *arXiv preprint arXiv:1408.1717*, 2014.

[14] G. Mishne, R. Talmon, I. Cohen, R. R. Coifman, and Y. Kluger, "Data-driven tree transforms and metrics," *IEEE transactions on signal and information processing over networks*, vol. 4, no. 3, pp. 451–466, 2017.

[15] G. Mishne, E. Chi, and R. Coifman, "Co-manifold learning with missing data," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4605–4614.

[16] A. Sharma and M. Ovsjanikov, "Geometric matrix completion: A functional view," *arXiv preprint arXiv:2009.14343*, 2020.

[17] A. Boyarski, S. Vedula, and A. Bronstein, "Deep matrix factorization with spectral geometric regularization," *arXiv preprint arXiv: 1911.07255*, 2019.

[18] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory¡? pub _newline=""?" *IEEE transactions on signal processing*, vol. 63, no. 24, pp. 6510–6523, 2015.

[19] S. P. Chepuri and G. Leus, "Graph sampling for covariance estimation," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 3, pp. 451–466, 2017.

[20] S. K. Kadambari and S. P. Chepuri, "Product graph learning from multi-domain data with sparsity and rank constraints," *IEEE Transactions on Signal Processing*, vol. 69, pp. 5665–5680, 2021.

[21] V. Kalofolias, "How to learn a graph from smooth signals," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Gretton and C. C. Robert, Eds., vol. 51. Cadiz, Spain: PMLR, 09–11 May 2016, pp. 920–929. [Online]. Available: https://proceedings.mlr.press/v51/kalofolias16.html