POPLAR: Parafac2 decOmPosition using auxiLiAry infoRmation

Ekta Gujral
UC Riverside
egujr001@ucr.edu

Georgios Theocharous
Adobe Inc.
theochar@adobe.com

Evangelos E. Papalexakis
UC Riverside
epapalex@cs.ucr.edu

***Abstract*—PARAFAC2 is a powerful method for analyzing multi-modal data consisting of irregular frontal slices. In this work, we propose POPLAR method that imposes graph Laplacians constraints induced by the similarity symmetric tensor as auxiliary information to force decomposition factors to behave similarly and the method is developed using AO-ADMM for 3-way PARAFAC2 tensor decomposition. To the best of our knowledge, POPLAR is the first approach to incorporate graph Laplacians constraints using auxiliary information. We extensively evaluate POPLAR's performance in comparison to state-of-the-art approaches across synthetic and real dataset, and POPLAR clearly exhibits better performance with respect to the Fitness (better $3-8\%$), and F1 score (better $5-20\%$) among the state-of-the-art factorization method. Furthermore, the running time for the method is comparable to the state-of-art method.**

I. Introduction

In real world applications, we often encounter multi-aspect relationships in data. For example, in social network analysis, interactions among various users and their interactions types are the main focus of interest. Tensors (or multi-way arrays) are highly suitable representation for such relationships. Tensor analysis methods have been extensively studied and applied by researchers [7], [18], [22] to many real-world problems. Regardless of recent development of traditional tensor decomposition approaches, there are certain instances [14], [15] wherein time modeling is difficult for the regular tensor factorization methods, due to either data irregularity or time-shifted latent factor appearance as shown in Figure 1. The PARAFAC2 decomposition, proposed by Harshman [12], is another alternative to the traditional model. The PARAFAC2 model is appropriate for 3-mode data that do not follow a perfect trilinear structure and allows one of the modes to vary. It has been extensively used in chemometrics [4], [25], electronic health record [23], natural language processing [10], and social sciences [13].

In many practical cases, we have multi-aspect information represented as tensors (PARAFAC or PARAFAC2), and we can compute information on the objects forming the relationships based on various similarities. For example, in the (user, item, time)-relationships, each user has location/calls/connectivity information, and each item has its product/service information. Therefore, we can consider that we have similarity measures that corresponds to the sets of matrices for non-variable modes of data. Even if recently, a constraint PARAFAC2 (COPA) fitting algorithm was proposed for large and sparse data [2], it cannot incorporate meaningful auxiliary information on the model factors such as: a) user-user interactions, which facilitates model interpretability and understanding, b) product/service similarity that provides relational information among objects.

To handle the these challenges and inspired by the work by Narita et al. [21] which incorporates object similarity into PARAFAC1 tensor factorization, we exploit the auxiliary information given as similarity tensor in a regularization framework for PARAFAC2 tensor factorization. We propose the Parafac2 decOmPosition using auxiLiAry infoRmation method (POPLAR), which introduces graph laplacian constraints/regularization in PARAFAC2 modeling that enables to improve the prediction accuracy of tensor decomposition.

Property	PARAFAC2	COPA	POPLAR
Sparsity	–	✓	✓
Graph Laplacian	–	–	✓
Scalability	–	✓	✓
Handle irregular tensors	✓	✓	✓

TABLE I: Comparison of models.

Our contributions are summarized as follows:

- **Novel Algorithm** We propose POPLAR, a method equipping the PARAFAC2 modeling with Graph Laplacian constraints. While POPLAR incorporates a additional auxiliary tensor for Laplacian constraints, it provides more accuracy than baselines and it is comparable in terms of scalability.
- **Experimental Evaluation** : we show experimental results on both synthetic and real datasets.

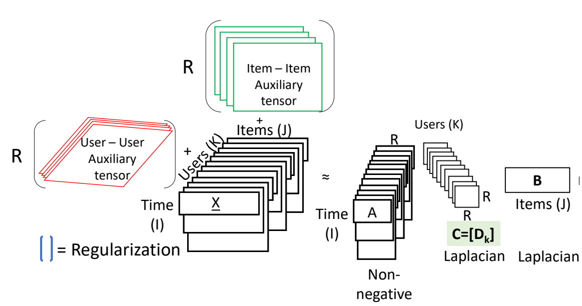

Fig. 1: An illustration of the laplacian constraints imposed by POPLAR on PARAFAC2 tensor decomposition.

The research is partially supported by a gift by Adobe Inc. and by the Department of the Navy, Naval Engineering Education Consortium under award no. N00174-17-1-0005. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties.

Table [I] provides our contributions in the terms of existing works.

II. Related Work

There are multiple studies to include auxiliary information into matrix factorization. Li et al. [19] proposed a regularizer by a Laplacian graph for one of the factor matrices depending on the data distribution. A similar approach is introduced by Cai et al. [8]. Lu et al. [20] proposed using the graph Laplacian and Kalman filter to incorporate both spatial and temporal information. In order to incorporate side information and enforce smoothness on factors, Adams et al. [1] extended the probabilistic matrix factorization. Some work use the auxiliary information as bias variables added to model parameters [27] and not as regularization term. Narita et al. [21] incorporates auxiliary information in form of similarity matrices as regularizer into PARAFAC1 tensor factorization. To best of our knowledge, our work is the first attempt to incorporate auxiliary information (tensor format) into PARAFAC2 tensor factorization.

III. Preliminaries and Notation

A. Tensors and Tensor Operations

A tensor is a higher order generalization of a matrix. An N-mode tensor is essentially indexed by N variables. In particular, regular matrix with two variables i.e. I and J is 2-mode tensor, whereas data cube (I, J and K) is viewed as a 3-mode tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$. The number of modes is also called "order". Matricization converts the tensor into a matrix representation without altering its values. Table II contains the symbols used throughout the paper.

TABLE II: Symbols and definitions

Symbol	Definition
$\mathcal{X}, \mathbf{X}, \mathbf{x}, x$	Tensor, Matrix, vector, scalar
$\mathbf{X}(:, \mathbf{i})$	Spans the entire i^{th} column of $\mathbf{X}$ (same for tensors)
$\mathbf{X}(\mathbf{i}, :)$	Spans the entire i^{th} row of $\mathbf{X}$ (same for tensors)
$diag(\mathbf{X})$	Diagonal of matrix $\mathbf{X}$
$\mathbf{X}_k$	k^{th} frontal slice of tensor $\mathcal{X}$
$\{\mathbf{X}_k\}$	the set of $\mathcal{X}_k$
$\circ, *$	Outer and Element-wise (Hadamard) product
$\otimes, \odot$	Kronecker and Khatri-Rao product

B. Classic PARAFAC2 Decomposition

PARAFAC2 model [12] differs from CPD/PARAFAC [5], [9], [11] where a low-rank trilinear model is not required. The CP decomposition applies the same factors across all the different modes, whereas PARAFAC2 allows for non-linearities such that variation across the values and/or the size of one mode as shown in Fig 1. PARAFAC2 with factors $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ can be written w.r.t. the frontal slices of the tensor $\mathcal{X}$ as:

$$\mathcal{X}_k = \mathbf{A}_k \mathbf{D}_k \mathbf{B}^T \tag{1}$$

where $k = 1, \ldots, K$, $\mathbf{A}_k \in \mathbb{R}^{I_k \times R}$, $\mathbf{D}_k = diag(C(k,:)) \in \mathbb{R}^{R\times R}$ is diagonal matrix, and $\mathbf{B} \in \mathbb{R}^{J\times R}$. Given the above modeling, the standard algorithm to solve PARAFAC2 for data $\mathcal{X}$ tackles the following optimization problem:

$$\min_{\{\mathbf{A_k}\},\{\mathbf{D_k}\},\mathbf{B}} \sum_{k=1}^{K} \|\mathcal{X}_k - \mathbf{A}_k\mathbf{D}_k\mathbf{B}^T\|_F^2 \tag{2}$$

subject to $\mathbf{A}_k = \mathbf{Q}_k\mathbf{H}$, $\mathbf{Q}_k^T\mathbf{Q}_k = \mathbf{I}$, and $\mathbf{D}_k$ is diagonal matrix. The $\mathbf{A}_k$ decomposed into two matrices, $\mathbf{Q}_k$ that has orthonormal columns and $\mathbf{H}$ which is invariant regardless of k. To solve Eq (2), most common method is Alternating Least Square (ALS) that updates $\mathbf{Q}_k$ by fixing other factor matrices i.e $\mathbf{H}, \mathbf{C}$, and $\mathbf{B}$. The orthogonal coupling matrix $\mathbf{Q}_k$ can be obtained by Singular Value decomposition (SVD) of $(\mathbf{HCB}^T\underline{\mathbf{X}}_k^T) = [\mathbf{P}_n, \Sigma_n, \mathbf{Z}_n^T]$. With $\mathbf{Q}_k^T = \mathbf{P}_n\mathbf{Z}_n^T$ fixed, the rest of factors can be obtained as:

$$\begin{aligned} \mathcal{L} = \min_{\mathbf{H,C,B}} \frac{1}{2}||\mathbf{Q}_k\underline{\mathbf{X}}_k - \mathbf{HCB}^T||_2^F s.t. \mathbf{Q}_k\mathbf{Q}_k^T = \mathbf{I}_r \\ \min_{\mathbf{H,C,B}} \frac{1}{2}||\underline{\mathbf{Y}} - \mathbf{HCB}^T||_2^F \end{aligned} \tag{3}$$

The Eq. (3) is equivalent of solving CP decomposition [17], [23] of $\underline{\mathbf{Y}}$ using ALS method and the constrained version of Eq. (3) can be written as:

$$\mathcal{L} = \min_{\mathbf{H,C,B}} \frac{1}{2}||\underline{\mathbf{Y}} - \mathbf{HCB}^T||_2^F + \frac{\alpha}{2}(\mathcal{R}(\underline{\mathbf{Y}}; \mathbf{H}, \mathbf{D}_k, \mathbf{B})) \tag{4}$$

C. AO-ADMM

A very well-suited optimization framework, that has shown promise in other, simpler tensor models [2], [16], [26], is the Alternating Method of Multipliers (ADMM) [6], applied in an alternating optimization (AO) fashion. In the next subsection we derive and describe our optimization method in detail.

D. Problem Definition

We consider exploiting auxiliary information for improving the prediction accuracy by PARAFAC2 decomposition, especially for sparse observations. The problem that we focus on in this paper is summarized as follows.

Given: A third-order PARAFAC2 tensor $\mathcal{X} \in \mathbb{R}^{I_k\times J}$ and symmetric similarity tensors $\mathcal{M} \in \mathbb{R}^{J\times J\times M}$ and $\mathcal{N} \in \mathbb{R}^{K\times K\times N}$, each corresponding to one of the regular modes of $\mathcal{X}$.
Find: A decomposition $\mathcal{X}$ defined by Eq.(2) and Eq (4).

IV. Proposed method: POPLAR

We propose Graph Laplacians regularization for PARAFAC2 tensors induced by the similarity tensor as auxiliary information to force factors to behave similarly. This a natural extension of the method proposed by Narita et al. [21] for tensor factorization using matrix as an auxiliary information. Consider 3-mode PARAFAC2 tensor with auxiliary tensors $\underline{\mathbf{M}}$ and $\underline{\mathbf{N}}$ on its fixed (2^{nd} and 3^{rd}) modes. The simple way to obtain these tensors is using various standard similarity methods like cosine similarity, Euclidean distance, Jaccard and ABC similarity etc. The regularization term we propose regularizes factor matrices of PARAFAC2 for its static modes using the similarity matrices (e.g user-user or item-item matrix). For simplicity, we explain process for 3^{rd} mode only. Thus, regularization term for 3^{rd} mode $\mathbf{C}$ is defined as:

$$\mathcal{R}(\mathbf{C}) = \sum_{n=1}^{N} Tr(\mathbf{C}^T \mathbf{L}_n \mathbf{C}) \quad (5)$$

where $\mathbf{L}$ is the Laplacian Matrix induced from the similarity tensor $\underline{\mathbf{M}}$ for the factor $\mathbf{C}$. Thus the objective function can be written as:

$$\mathcal{L} = min_{\{\mathbf{A_k}\},\{\mathbf{D_k}\},\mathbf{B}} \sum_{k=1}^{K} \|\mathcal{X}_k - \mathbf{A}_k \mathbf{D}_k \mathbf{B}\|_F^2 + \frac{\alpha}{2}(\mathcal{R}(\underline{\mathbf{M}}) + \mathcal{R}(\underline{\mathbf{N}})) \quad (6)$$

The regularization is imposed by using the Graph Laplacians (GL) method on the similarity tensors ($\underline{\mathbf{M}}$ and $\underline{\mathbf{N}}$) that helps to direct two similar objects in 2^{nd} and 3^{rd} mode to behave similarly. Mathematically, Equ. (6) can be re-written as :

$$\begin{aligned}\mathcal{L} = \min_{\{\mathbf{A_k}\},\{\mathbf{D_k}\},\mathbf{B}} \sum_{k=1}^{K} \|\mathcal{X}_k - \mathbf{A}_k \mathbf{D}_k \mathbf{B}\|_F^2 + \\ \frac{\alpha}{2}(Tr(\sum_{m=1}^{M} \mathbf{B}^T \mathbf{L}_m \mathbf{B} + \sum_{n=1}^{N} \mathbf{C}^T \mathbf{L}_n \mathbf{C})\end{aligned} \quad (7)$$

where $\mathbf{L}_m$ and $\mathbf{L}_n$ are the Graph Laplacian matrices obtained from the slices of similarity tensors $\underline{\mathbf{M}}$ and $\underline{\mathbf{N}}$, respectively. The Graph Laplacian matrix can be computed as follows:

$$\mathbf{L}_n = \mathbf{Deg}_n - \mathbf{N}_n$$

where $\mathbf{Deg}$ is degree matrix and it i^{th} diagonal element is the sum of all of the elements in the i^{th} row similarity matrix and computed as:

$$\mathbf{Deg}_{n_{i,j}} = \begin{cases} \sum_{j=1}^{N} \mathbf{N}_n(i,j), & \text{i = j.} \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

The regularization term can be simply interpreted as:

$$Tr(\sum_{n=1}^{N} \mathbf{C}^T \mathbf{L}_n \mathbf{C}) = \sum_{n=1}^{N} (\sum_{i,j=1}^{K} \mathbf{N}_{n_{i,j}} \sum_{r=1}^{R} (\mathbf{C}_{i,r} - \mathbf{C}_{j,r})^2) \quad (9)$$

This term implies that, if two objects are similar to each other, the corresponding factor vectors should be similar to each other. Thus, when using AO-ADMM the update rule for $\mathbf{C}$ is:

$$\mathbf{C}^T := ((\mathbf{H}^T \mathbf{H} * \mathbf{B}^T \mathbf{B}) + \rho \mathbf{I})^{-1} (\underline{\mathbf{X}}_{(3)} (\mathbf{B} \odot \mathbf{H}) + \rho(\overline{\mathbf{C}} + \mathbf{M}_{\mathbf{C}^T}))^T$$

$$\overline{\mathbf{C}} := \min_{\overline{\mathbf{C}}} \sum_{n=1}^{N} Tr(\overline{\mathbf{C}}^T \mathcal{L}_n \overline{\mathbf{C}}) + \rho ||\overline{\mathbf{C}} - \mathbf{C}^T + \mathbf{M}_{\mathbf{C}^T}||_F^2$$

$$\mathbf{M}_{\mathbf{C}^T} := \mathbf{M}_{\mathbf{C}^T} - \mathbf{C}^T + \overline{\mathbf{C}} \text{ s.t. } \overline{\mathbf{C}} = \mathbf{C}$$

where $\mathbf{M}_{\mathbf{C}^T}$ is a dual variable and the Lagrange multiplier ρ is a step size related to $\mathbf{C}^T$ factor matrix and set to minimum value between 10^{-3} and $Tr(\mathbf{H}^T \mathbf{H} * \mathbf{B}^T \mathbf{B})/R$ to yield good performance.

Adapting AO-ADMM to solve PARAFAC2 with Laplacians constraints has following benefits: (1) AO-ADMM is more general than other methods in the sense that the loss function doesn't need to be differentiable; (2) Simple to implement, parallelize and easy to incorporate a wide variety of constraints that can obtained using simple element-wise operations; (3) computational savings gained by using the Cholesky decomposition and (4) The splitting scheme can be applied to large-scale data. Data can be distributed across different machines and optimize objective locally with communication on the primal, auxiliary and dual variables between the machines.

V. Experimental Results

A. *Data Set Description*

*1) **Auxiliary Tensor Creation**:* We created Auxiliary tensors using ABC similarity [24], Pearson correlation [24] , K-NN similarity [3], Jacard similarity and Edit distance.

*2) **Synthetic Data**:* : Table III provides summary statistics regarding all datasets used. For all synthetic data we use rank $R = 10$. The entries of loading matrix $\mathbf{B}$ and $\mathbf{C}$ are Gaussian with unit variance, and orthogonality is imposed on factors $\mathbf{H}$ and $\mathbf{Q}$, then a few entries are clipped to zero randomly to create a sparse PARAFAC2 tensor. The labels are created by selecting highest value index for each entry in matrix $\mathbf{C}$.

Dataset	K	J	max(I_k)	$\#nnz$	R
SYN-I	25k	5k	1k	2.1 Mil.	10
SYN-II	50k	10k	2k	5.4 Mil.	10
ADOBE	80k	1.7k	17k	3 Mil.	10, 40

TABLE III: Summary statistics for the datasets of our experiments. K is the number of users, J is the number of items, I_k is the number of time observations for the k^{th} subject, $\#nnz$ corresponds to the total number of non-zeros in tensor $\underline{\mathbf{X}}$ and R is rank of tensor $\underline{\mathbf{X}}$.

*3) **Real Data**:* : Adobe dataset is sequential data and it consists of tutorial sequence 7 million anonymized users. The data is structured as sequence-by-tutorial-by-user. We selected users who watched at least unique 10 tutorials. Thus, the dataset is of dimension [max 17k $\times$ 1.7k $\times$ 80k]. We compute user-user tensor similarity tensors. We have semi synthetic ground truth values for this dataset and we assigned each user to class based on the type of tutorial watched.

B. *Baselines*

In this experiment, two baselines have been used as the competitors to evaluate the performance.

- **PARAFAC2** [17] : an implementation of standard fitting algorithm PARAFAC2 with random initialization.
- **COPA** [2] : a scalable constrained PARAFAC2 fitting algorithm was proposed for large and sparse data.

C. *Evaluation Measures*

We evaluate POPLAR and the baselines using three quantitative criteria: Fitness [0-1], F1-score [0-1] and CPU-Time (in seconds). These measures provide a quantitative way to compare the performance of our method. For Fitness and F1-score, higher is better.

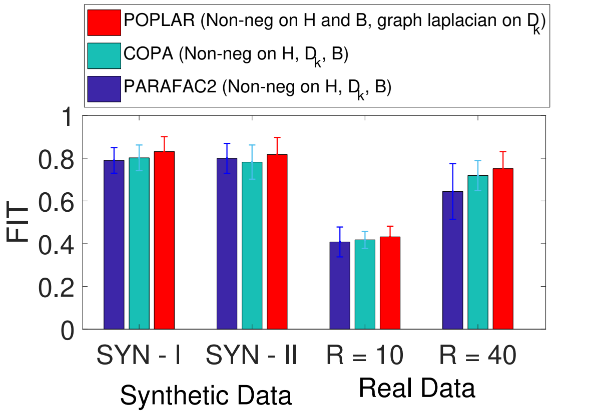

Fig. 2: Comparison of FIT for different approaches on synthetic data with rank R = 10 and two target ranks R = 10 and R = 40 on real world dataset. Overall, POPLAR shows comparable fitness to baseline while supporting additional graph laplacian constraint.

D. Results

1) Fitness and Accuracy: We run each method for 3 different random initialization and provide the average and standard deviation of FIT as shown in Figure 2. This Figure illustrates the impact of proposed constraint on the FIT values across both synthetic for fixed rank R = 10 and real datasets for two different target ranks (R=10, 40). Overall,POPLAR achieves average $3 - 8\%$ improvement.

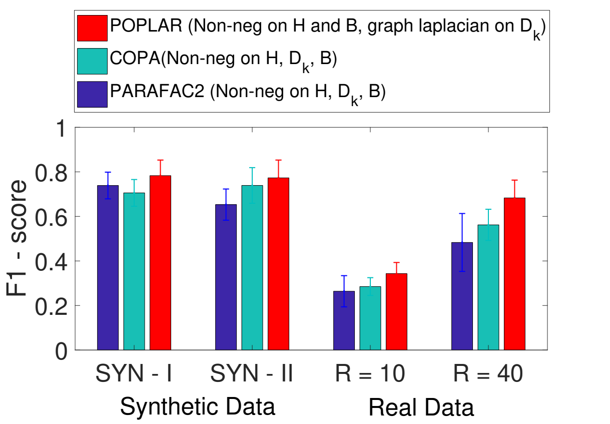

Fig. 3: Comparison of F1-score for different approaches on synthetic data with rank R = 10 and two target ranks R = 15 and R = 40 on real world dataset. Overall, POPLAR achieves better F1 score to baseline.

Similarly, we evaluate accuracy in terms of predicting correct labels using F1-score. We provide the average and standard deviation of F1 score as shown in Figure 3. Overall, POPLAR achieves significant improvement $5 - 20\%$ over baselines.

2) Computation time: Finally, we briefly discuss the computation time of our method. Although using auxiliary tensor as constraints slightly increases the time complexity of the PARAFAC2 decomposition method, the actual computation time is almost as same as that for baseline methods as shown in Figure 4. This is partially because 1) POPLAR converges (tolerance = 10^{-7}) faster than baselines 2) we used medium sized datasets in the experiments, and further investigation with larger datasets (K $> 10^5$) should be made in future work.

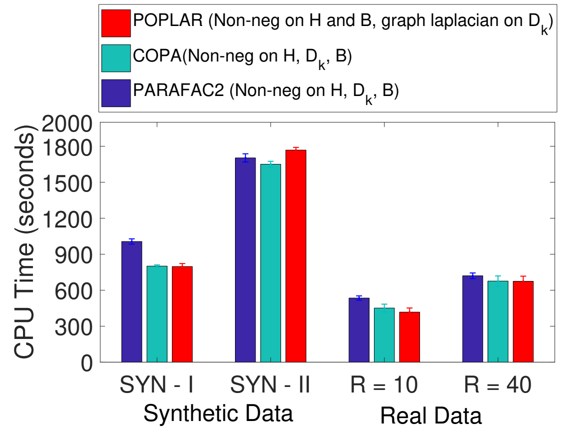

Fig. 4: The CPU Time comparison (average and standard deviation) in seconds for non-negative version of PARAFAC2 & COPA and POPLAR for 3 different random initialization on synthetic data with rank R = 10 and two target ranks R = 10 and R = 40 on real world dataset. Note that even with additional processing of auxiliary tensor POPLAR performs just slightly slower than COPA for SYN-II, which does not support such graph laplacian constraints.

3) Scalability: We also evaluate the scalability of our algorithm on synthetic dataset. A PARAFAC2 tensors $\underline{\mathbf{X}} \in \mathbb{R}^{max 1000 \times 1000 \times [5k - 100k]}$ are decomposed with increasing target rank. The time needed by POPLAR increases very moderately. Our proposed method, successfully decomposed the tensor in reasonable time as shown in Figure 5.

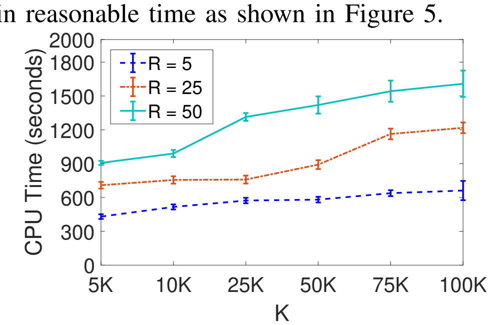

Fig. 5: The average time in seconds for varying target rank.

VI. Conclusions

This paper outlined our vision on exploring the graph laplacian regularization on PARAFAC2 tensor decomposition using auxiliary information to improve accuracy of factorization. We propose POPLAR, a AO-ADMM-based framework that is able to offer interpretable results, and we provide a experimental analysis on synthetic as well as real world dataset. Furthermore, this paper outlines a set of interesting future research directions:

- How can we couple one of auxiliary tensor with PARAFAC2 tensor to obtain better approximation?
- What other constraints, other than graph laplacian or non-negative, for the PARAFAC2 decomposition are well suited for various application and have potential to offer more accurate results?
- How can we incorporate cross mode graph laplacian regularization for PARAFAC2 decomposition?

REFERENCES

[1] Ryan Prescott Adams, George E Dahl, and Iain Murray. Incorporating side information in probabilistic matrix factorization with gaussian processes. *arXiv preprint arXiv:1003.4944*, 2010.

[2] Ardavan Afshar, Ioakeim Perros, Evangelos E Papalexakis, Elizabeth Searles, Joyce Ho, and Jimeng Sun. Copa: Constrained parafac2 for sparse & large datasets. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 793–802. ACM, 2018.

[3] Saba A Al-Sayouri, Ekta Gujral, Danai Koutra, Evangelos E Papalexakis, and Sarah S Lam. t-pine: Tensor-based predictable and interpretable node embeddings. *arXiv preprint arXiv:1805.01889*, 2018.

[4] José Manuel Amigo, Marta J Popielarz, Raquel M Callejón, Maria L Morales, Ana M Troncoso, Mikael A Petersen, and Torben B Toldam-Andersen. Comprehensive analysis of chromatographic data by using parafac2 and principal components analysis. *Journal of Chromatography a*, 1217:4422–4429, 2010.

[5] Brett W Bader, Tamara G Kolda, et al. Matlab tensor toolbox version 2.6, available online, february 2015, 2015.

[6] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, pages 1–122, 2011.

[7] Rasmus Bro. Parafac. tutorial and applications. *Chemometrics and intelligent laboratory systems*, 38, 1997.

[8] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1548–1560, 2010.

[9] J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika*, pages 283–319, 1970.

[10] Peter A Chew, Brett W Bader, Tamara G Kolda, and Ahmed Abdelali. Cross-language information retrieval using parafac2. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 143–152. ACM, 2007.

[11] R.A. Harshman. Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis. 1970.

[12] Richard A Harshman. Parafac2: Mathematical and technical notes. *UCLA working papers in phonetics*, 22(3044):122215, 1972.

[13] Nathaniel E Helwig. Estimating latent trends in multivariate longitudinal data via parafac2 with functional and structural constraints. *Biometrical Journal*, 59:783–803, 2017.

[14] Joyce C Ho, Joydeep Ghosh, Steve R Steinhubl, Walter F Stewart, Joshua C Denny, Bradley A Malin, and Jimeng Sun. Limestone: High-throughput candidate phenotype generation via tensor factorization. *Journal of biomedical informatics*, 52:199–211, 2014.

[15] Joyce C Ho, Joydeep Ghosh, and Jimeng Sun. Marble: high-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization. In *Proc. of the 20th ACM SIGKDD Int. Conf on Knowledge discovery and data mining*. ACM, 2014.

[16] Kejun Huang, Nicholas D Sidiropoulos, and Athanasios P Liavas. A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. *IEEE Transactions on Signal Processing*, pages 5052–5065, 2016.

[17] Henk AL Kiers, Jos MF Ten Berge, and Rasmus Bro. Parafac2—part i. a direct fitting algorithm for the parafac2 model. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 13(3-4):275–294, 1999.

[18] T.G. Kolda and B.W. Bader. Tensor decompositions and applications. *SIAM review*, 2009.

[19] Wu-Jun Li and Dit-Yan Yeung. Relation regularized matrix factorization. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[20] Zhengdong Lu, Deepak Agarwal, and Inderjit S Dhillon. A spatio-temporal approach to collaborative filtering. In *Proceedings of the third ACM conference on Recommender systems*, pages 13–20. ACM, 2009.

[21] Atsuhiro Narita, Kohei Hayashi, Ryota Tomioka, and Hisashi Kashima. Tensor factorization using auxiliary information. *Data Mining and Knowledge Discovery*, 25(2):298–324, 2012.

[22] Evangelos E Papalexakis, Christos Faloutsos, and Nicholas D Sidiropoulos. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2016.

[23] Ioakeim Perros, Evangelos E Papalexakis, Fei Wang, Richard Vuduc, Elizabeth Searles, Michael Thompson, and Jimeng Sun. Spartan: Scalable parafac2 for large & sparse data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 375–384. ACM, 2017.

[24] Tara Safavi, Chandra Sripada, and Danai Koutra. Scalable hashing-based network discovery. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 405–414. IEEE, 2017.

[25] Thomas Skov, Davide Ballabio, and Rasmus Bro. Multiblock variance partitioning: A new approach for comparing variation in multiple data blocks. *Analytica chimica acta*, 615:18–29, 2008.

[26] Shaden Smith, Alec Beri, and George Karypis. Constrained tensor factorization with accelerated ao-admm. In *2017 46th International Conference on Parallel Processing (ICPP)*, pages 111–120. IEEE, 2017.

[27] Kai Yu, John Lafferty, Shenghuo Zhu, and Yihong Gong. Large-scale collaborative prediction using a nonparametric random effects model. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1185–1192. ACM, 2009.