Tensor Factorization Using Auxiliary Information

Atsuhiro Narita[1], Kohei Hayashi[2], Ryota Tomioka[1], and Hisashi Kashima[1,3]

[1] Department of Mathematical Informatics,
The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
{atsuhiro_narita,tomioka,kashima}@mist.i.u-tokyo.ac.jp
[2] Graduate School of Information Science,
Nara Institute of Science and Technology,
8916-5 Takayama, Ikoma, Nara, 630-0192, Japan
kohei-h@is.naist.jp
[3] Basic Research Programs PRESTO,
Synthesis of Knowledge for Information Oriented Society

Abstract. Most of the existing analysis methods for tensors (or multi-way arrays) only assume that tensors to be completed are of low rank. However, for example, when they are applied to tensor completion problems, their prediction accuracy tends to be significantly worse when only limited entries are observed. In this paper, we propose to use relationships among data as auxiliary information in addition to the low-rank assumption to improve the quality of tensor decomposition. We introduce two regularization approaches using graph Laplacians induced from the relationships, and design iterative algorithms for approximate solutions. Numerical experiments on tensor completion using synthetic and benchmark datasets show that the use of auxiliary information improves completion accuracy over the existing methods based only on the low-rank assumption, especially when observations are sparse.

1 Introduction

In real data analysis applications, we often have to face handling multi-object relationships. For example, in on-line marketing scenarios, we analyze relationships among customers, items, and time to capture temporal dynamics of customers' interests and utilize them for recommendation. In social network analysis, interactions among people and their interaction types are the focus of interest. Similar situations arise in bio- and chemo-informatics as protein-protein interactions and drug-target interactions under various conditions. Tensors (or multi-way arrays) [10] are highly suitable representation for such multi-object relationships (Fig. 1). Tensor analysis methods, especially, models and efficient algorithms for low-rank tensor decompositions have been extensively studied and applied to many real-world problems. CANDECOMP/PARAFAC(CP)-decomposition and Tucker decomposition are two widely-used low rank decompositions of tensors (Fig. 2).

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part II, LNAI 6912, pp. 501–516, 2011.

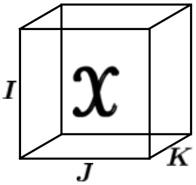
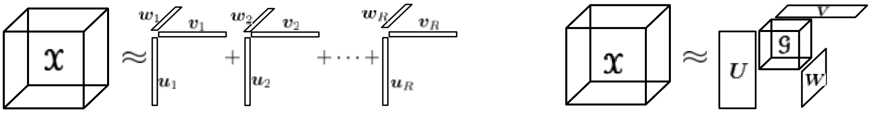


Fig. 1. A third-order tensor (or, a three-way array) $\boldsymbol{\mathcal{X}}$ of size $I \times J \times K$ repesents relationships among three sets of objects, S_1, S_2 and S_3, each of which size is I, J, and K, respectively



(a) CP decomposition.

(b) Tucker decomposition.

Fig. 2. Two widely used low-rank tensor decompositions: CP-decomposition and Tucker decomposition

Tensor completion is one of important applications of tensor analysis methods. Given a tensor with some of its elements being missing, the task is to impute the missing values. In the context of the previous on-line marketing scenario, given the observations for some (customer, item, time)-tuples, we can make recommendations by imputing unobserved combinations of them. Tensor completion is also used for link prediction [6,9] and tag recommendation [15]. Similar to the other tensor analysis methods, low-rank assumption of tensors is often used for imputing missing values. However, when observations are sparse, in other words, the fraction of unobserved elements is high, predictive accuracy of tensor completion methods only with the low-rank assumption tends to be worse. For example, Figure 3 shows the prediction errors by CP-decomposition against the fraction of unobserved elements for a particular dataset. (The experimental setting is the same as that for Fig. 4(c), which will be described in detail in the experimental section.) We can see the accuracy of tensor completion severely degrades when observations are sparse. This fact implies that the low-rank assumption by itself is not sufficient and we need other assumptions to introduce more prior knowledge of subjects.

In many cases, we have not only relational information among objects, but also information on the objects themselves. For example, in the on-line marketing scenarios, each customer has his/her demographic information, and each item has its product information. We consider exploiting these auxiliary information for improving the prediction accuracy of tensor decomposition, especially for sparse cases. Inspired by the work by Li *et al.* [12] which incorporates object similarity into matrix factorization, we exploit the auxiliary information given as similarity matrices in a regularization framework for tensor factorization. We propose two specific regularization methods, one of which we call “within-mode

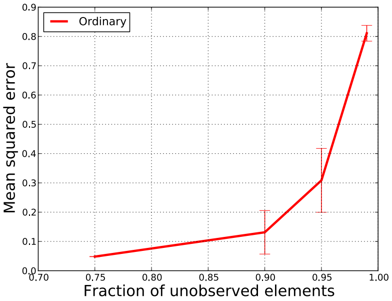


Fig. 3. The tensor completion performance by CP-decomposition for the 'Flow injection' dataset. Prediction accuracy severely degenerates when observations are sparse.

regularization" is a natural extension of the method proposed by Li *et al.* for matrix factorization. It uses the graph Laplacians induced by the similarity matrices to force two similar objects in each mode to behave similarly, in other words, to have similar factors. The second method we call "cross-mode regularization" exploits the similarity information more aggressively to address extremely sparse cases. We apply the two proposed regularization methods to each of CP-decomposition and Tucker decomposition, and give iterative decomposition algorithms for obtaining approximate solutions. In each iteration, we solve a particular Sylvester equation for CP-decomposition, or obtain eigendecomposition for Tucker decomposition. To best of our knowledge, our work is the first to incorporates auxiliary information into tensor decomposition.

Finally, we show experimental results on missing value imputation using both synthetic and real benchmark datasets. We test two kinds of assumptions on missing elements. The first one is element-wise missing where each element is missing independently. The second one is slice-wise missing (in other words, object-wise missing), where missing values occur in a more bursty manner, and all of the elements related to some objects are completely missing. The experimental results demonstrate that the use of auxiliary information improves imputation accuracy when observations are sparse, and the cross-mode regularization method especially works well in extremely sparse slice-wise missing cases.

The rest of the paper is organized as follows. Section 2 reviews the existing low-rank tensor decomposition methods, and introduces the tensor completion problem with auxiliary information that we focus on in this paper. In Section 3, we propose two regularization strategies, within-mode regularization and cross-mode regularization, for incorporating auxiliary information in tensor decomposition, and give their approximate solutions. Section 4 shows the experimental results using several datasets to demonstrate the proposed methods work well especially when observations are sparse. Section 5 reviews related work, and Section 6 concludes this paper with some ideas for future directions.

2 Tensor Completion Problem with Auxiliary Information

We first review the existing low-rank tensor decomposition methods, and then formulate the tensor completion problem with auxiliary information.

2.1 Tensor Analysis Using Low-Rank Decomposition

Let $\boldsymbol{\mathcal{X}}$ be third-order tensor (i.e. a three-way array) with $I \times J \times K$ real-valued elements[1]. The third-order tensor $\boldsymbol{\mathcal{X}}$ models relationships among objects from three sets S_1, S_2, and S_3. For example, in the context of on-line marketing, S_1, S_2, and S_3 represent sets of customers, items, and time stamps, respectively. The (i, j, k)-th element $[\boldsymbol{\mathcal{X}}]_{i,j,k}$ indicates the i-th user's rating of the j-th item at time k.

We often assume the tensor is of "low-rank" when we analyze tensor-represented data. In contrast to matrices (that are special cases of tensors), definitions of the "low-rank" tensor are not unique. CANDECOMP/PARAFAC (CP)-decomposition and Tucker decomposition are often used as definitions of low-rank tensors.

The CP-decomposition is a natural extension of the matrix rank, and it approximates a tensor by the sum of R rank-1 tensors. The CP-decomposition $\hat{\boldsymbol{\mathcal{X}}}$ of $\boldsymbol{\mathcal{X}}$ is defined as

$$\hat{\boldsymbol{\mathcal{X}}} \equiv \sum_{i=1}^{R} \mathbf{u}_i \circ \mathbf{v}_i \circ \mathbf{w}_i,$$

where $\circ$ indicates the outer product operation. Or, it can also be represented by using mode-i multiplications as

$$\hat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{J}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}, \tag{1}$$

where $\boldsymbol{\mathcal{J}} \in \mathbb{R}^{R \times R \times R}$ is a unit tensor with all of its super-diagonal elements being 1 and the other elements being 0, $\mathbf{U} \in \mathbb{R}^{I \times R}$, $\mathbf{V} \in \mathbb{R}^{J \times R}$, $\mathbf{W} \in \mathbb{R}^{K \times R}$ are factor matrices, and $\times_i$ is the mode-i multiplication [10]. When the left-hand side is equal to the right-hand side in the above relation, we say $\boldsymbol{\mathcal{X}}$ is of rank R.

The Tucker decomposition approximates a tensor with a small "core tensor" and factor matrices, which is defined as

$$\hat{\boldsymbol{\mathcal{X}}} \equiv \boldsymbol{\mathcal{G}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}, \tag{2}$$

where $\boldsymbol{\mathcal{G}}$ is a (P, Q, R)-tensor and $\mathbf{U} \in \mathbb{R}^{I \times P}$, $\mathbf{V} \in \mathbb{R}^{J \times Q}$, $\mathbf{W} \in \mathbb{R}^{K \times R}$ are factor matrices. In this case, we say $\boldsymbol{\mathcal{X}}$ is of rank (P, Q, R).

For most of realistic case, observations are perturbed by noise, and the strict low-rank decompositions do not hold even when the "true" $\boldsymbol{\mathcal{X}}$ is actually of low-rank. Therefore, we try to find a decomposition $\hat{\boldsymbol{\mathcal{X}}}$ that best approximates

[1] For simplicity, we focus on third-order tensors in this paper. However, the discussion can be directly applied to higher-order tensors.

the original tensor $\boldsymbol{\mathcal{X}}$ in terms of the squared loss by the following optimization problem,

$$\text{minimize}_{\hat{\boldsymbol{\mathcal{X}}}} \quad \|\boldsymbol{\mathcal{X}} - \hat{\boldsymbol{\mathcal{X}}}\|_F^2, \tag{3}$$

where $\|\cdot\|_F$ indicates the Frobenius norm, and $\hat{\boldsymbol{\mathcal{X}}}$ is defined by Eq. (1) for CP-decomposition, or by Eq. (2) for Tucker decomposition. It is generally hard to obtain the optimal solution, so we use approximation methods which optimize $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ alternately.

2.2 Tensor Completion with Auxiliary Information

Among various tensor-related problems, tensor completion problem is one of the important problems, where the task is to impute the missing values of a given tensor with missing values. The low-rank assumption is usually used as a heuristic for inferring the missing parts. Since not all of the elements are observed in the optimization problem (3) in this case, the EM algorithm is often applied to this purpose [18,20]. First, we fill the missing parts with some initial estimates (such as the average of the observed elements), and apply tensor decomposition to the filled tensor. We then obtain new estimates by assembling the decomposed tensor. We continue the decomposition step and the reconstruction step until convergence to obtain final estimates.

Since the EM algorithm uses unobserved elements for its computation, it is not efficient enough for large-scale data. Therefore, another approach modifies the objective function (3) to focus only on observed parts [1]. In this paper, we construct our methods based on the EM-based approach, however, the basic idea can also be applied to the other approaches.

The low-rank assumption of tensors makes it possible to impute missing values. However, when observations are sparse, in other words, the fraction of unobserved elements is high, predictive accuracy of tensor completion methods only with the low-rank assumption severely degrades. (See Figure 3 showing the predictive errors against the fraction of unobserved elements for a dataset.) Therefore, the low-rank assumption by itself is not sufficient, and we need other assumptions for obtaining satisfactory prediction accuracy.

In many realistic cases, we have not only relational information represented as tensors, but also information on the objects forming the relationships. For example, in the (customer, item, time)-relationships, each customer has his/her demographic information, and each item has its product information. We also know that time is continuous and can assume temporal smoothness. Therefore, we assume that we have similarity measures for S_1, S_2, and S_3, each of which corresponds to the sets of objects for each of the three modes. We define a non-negative symmetric matrix $\mathbf{A}_1$ for representing the similarity between arbitrary two objects in S_1. $\mathbf{A}_2$ and $\mathbf{A}_3$ are defined similarly.

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

Problem: (Third-order) tensor completion with auxiliary information

- **INPUT**
  - A third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$, some of whose elements are observed and the others are unobserved.
  - Three non-negative symmetric similarity matrices $\mathbf{A}_1 \in {\mathbb{R}^+}^{I \times I}$, $\mathbf{A}_2 \in {\mathbb{R}^+}^{J \times J}$, and $\mathbf{A}_3 \in {\mathbb{R}^+}^{K \times K}$, each of which corresponds to one of the three modes of $\boldsymbol{\mathcal{X}}$.
- **OUTPUT:** A decomposition $\hat{\boldsymbol{\mathcal{X}}}$ defined by either Eq. (1) for CP-decomposition or Eq. (2) for Tucker decomposition.

3 Proposed Methods: Within-Mode and Cross-Mode Regularization

In this section, we propose two regularization methods for incorporating auxiliary information into tensor factorization. Both CP-decomposition and Tucker decomposition are generalized with the regularization framework.

3.1 Regularization Using Auxiliary Similarity Matrices

Given three object similarity matrices $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$ besides $\boldsymbol{\mathcal{X}}$, how can we use them for improving tensor decompositions? Probably, one of natural assumptions we can make is that "two similar objects behave similarly". We implement this idea as regularization terms for the optimization problem (3). Namely, instead of the objective function in Eq. (3), we minimize the following regularized objective function with a regularization term $R(\hat{\boldsymbol{\mathcal{X}}}; \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$.

$$f(\hat{\boldsymbol{\mathcal{X}}}) \equiv \frac{1}{2}\|\boldsymbol{\mathcal{X}} - \hat{\boldsymbol{\mathcal{X}}}\|_F^2 + \frac{\alpha}{2} R(\hat{\boldsymbol{\mathcal{X}}}; \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3). \tag{4}$$

In Eq. (4), α is a positive regularization constant.

We propose two specific choices of the regularization term $R(\hat{\boldsymbol{\mathcal{X}}}; \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$. The first method we call "within-mode regularization" is a natural extension of the method proposed by Li *et al.* [12] for matrix factorization. It uses the graph Laplacians induced by the three similarity matrices to force two similar objects in each mode to behave similarly, in other words, to have similar factors. The second method we call "cross-mode regularization" exploits the similarity information more aggressively to address extremely sparse cases. It uses the graph Laplacian induced by the Kronecker product of the three similarity matrices to regularize factors for all of the modes at the same time by taking interactions across different modes into account.

3.2 Method 1: Within-Mode Regularization

The first regularization term we propose regularizes factor matrices for each mode using the similarity matrices. The "within-mode" regularization term is defined as

$$R(\hat{\boldsymbol{\mathcal{X}}}; \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3) \equiv \operatorname{tr}\left(\mathbf{U}^\top \mathbf{L}_1 \mathbf{U} + \mathbf{V}^\top \mathbf{L}_2 \mathbf{V} + \mathbf{W}^\top \mathbf{L}_3 \mathbf{W}\right), \tag{5}$$

where $\mathbf{L}_1$ is the Laplacian matrix induced from the similarity matrix $\mathbf{A}_1$ for the object set S_1. The Laplacian matrix is defined as

$$\mathbf{L}_1 \equiv \mathbf{D}_1 - \mathbf{A}_1,$$

where $\mathbf{D}_1$ is the diagonal matrix whose i-th diagonal element is the sum of all of the elements in the i-th row of $\mathbf{A}_1$. The Laplacian matrices for the other two sets, $\mathbf{L}_2 \equiv \mathbf{D}_2 - \mathbf{A}_2$ and $\mathbf{L}_3 \equiv \mathbf{D}_3 - \mathbf{A}_3$, are defined similarly.

To interpret the regularization term, we note $\mathrm{tr}(\mathbf{U}^\top \mathbf{L}_1 \mathbf{U})$ can be rewritten as

$$\mathrm{tr}(\mathbf{U}^\top \mathbf{L}_1 \mathbf{U}) = \sum_{i,j=1}^{I} [\mathbf{A}_1]_{i,j} \sum_{r=1}^{R} \left([\mathbf{U}]_{i,r} - [\mathbf{U}]_{j,r}\right)^2, \tag{6}$$

where $[\cdot]_{i,j}$ denotes the (i,j)-th element of a matrix. This term implies that, if two objects (say, $s_i, s_j \in S_1$) are similar to each other (that is, $[\mathbf{A}_1]_{i,j}$ is large), the corresponding factor vectors ($[\mathbf{U}]_{i*}$ and $[\mathbf{U}]_{j*}$) should be similar to each other.

CP-decomposition. The objective function for CP-decomposition with the within-mode regularization is written as

$$\begin{aligned} f(\boldsymbol{\mathcal{G}}, \mathbf{U}, \mathbf{V}, \mathbf{W}) \equiv & \frac{1}{2} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{J}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}\|_F^2 \\ & + \frac{\alpha}{2} \mathrm{tr}\left(\mathbf{U}^\top \mathbf{L}_1 \mathbf{U} + \mathbf{V}^\top \mathbf{L}_2 \mathbf{V} + \mathbf{W}^\top \mathbf{L}_3 \mathbf{W}\right). \end{aligned} \tag{7}$$

Eq. (7) is not a convex function for $(\mathbf{U}, \mathbf{V}, \mathbf{W})$, but is convex for each of $\mathbf{U}, \mathbf{V}$, and $\mathbf{W}$. Therefore, we optimize one of $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ with fixing the others to the current values, and alternately update them by changing the factor matrix to optimize.

Suppose we want to optimize $\mathbf{U}$ with fixing $\mathbf{V}$ and $\mathbf{W}$. Unfolding Eq. (7) by the first mode (i.e. making the mode-1 matricization), we obtain

$$\begin{aligned} f(\boldsymbol{\mathcal{G}}, \mathbf{U}, \mathbf{V}, \mathbf{W}) = & \frac{1}{2} \|\mathbf{X}_{(1)} - \mathbf{U} \left(\mathbf{W} \odot \mathbf{V}\right)^\top \|_F^2 \\ & + \frac{\alpha}{2} \mathrm{tr}\left(\mathbf{U}^\top \mathbf{L}_1 \mathbf{U} + \mathbf{V}^\top \mathbf{L}_2 \mathbf{V} + \mathbf{W}^\top \mathbf{L}_3 \mathbf{W}\right) \\ = & \frac{1}{2} \mathrm{tr}\left(\left(\mathbf{X}_{(1)} - \mathbf{U}(\mathbf{W} \odot \mathbf{V})^\top\right)^\top \left(\mathbf{X}_{(1)} - \mathbf{U}(\mathbf{W} \odot \mathbf{V})^\top\right)\right) \\ & + \frac{\alpha}{2} \mathrm{tr}\left(\mathbf{U}^\top \mathbf{L}_1 \mathbf{U} + \mathbf{V}^\top \mathbf{L}_2 \mathbf{V} + \mathbf{W}^\top \mathbf{L}_3 \mathbf{W}\right), \end{aligned}$$

where $\mathbf{X}_{(n)}$ denotes the mode-n matricization of $\boldsymbol{\mathcal{X}}$, and $\odot$ denotes the Khatri-Rao product [10]. Differentiating this with respect to $\mathbf{U}$, and setting it to be zero gives the Sylvester equation,

$$\mathbf{U}(\mathbf{W} \odot \mathbf{V})^\top (\mathbf{W} \odot \mathbf{V}) + \alpha \mathbf{L}_1 \mathbf{U} = \mathbf{U}\left(\mathbf{V}^\top \mathbf{V} * \mathbf{W}^\top \mathbf{W}\right) + \alpha \mathbf{L}_1 \mathbf{U} = \mathbf{X}_{(1)}(\mathbf{W} \odot \mathbf{V}),$$

where $*$ denotes the Hadamard product (i.e. element-wise product). The Sylvester equation can be solved by several numerical approaches such as the one implemented as the `dlyap` function in MATLAB®.

Tucker Decomposition. In the case of Tucker decomposition, the objective function becomes

$$f(\mathbf{U},\mathbf{V},\mathbf{W}) \equiv \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}\|_F^2 + \alpha \operatorname{tr}\left(\mathbf{U}^\top \mathbf{L}_1 \mathbf{U} + \mathbf{V}^\top \mathbf{L}_2 \mathbf{V} + \mathbf{W}^\top \mathbf{L}_3 \mathbf{W}\right). \quad (8)$$

We minimize the objective function (8) under the orthogonality constraints, $\mathbf{U}^\top\mathbf{U} = \mathbf{I}, \mathbf{V}^\top\mathbf{V} = \mathbf{I}$, and $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$. Noting the core tensor $\boldsymbol{\mathcal{G}}$ is obtained as the closed form solution,

$$\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{X}} \times_1 \mathbf{U}^\top \times_2 \mathbf{V}^\top \times_3 \mathbf{W}^\top,$$

the first term of Eq. (8) can be rewritten as

$$\begin{aligned}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}\|_F^2 &= \|\boldsymbol{\mathcal{X}}\|_F^2 - \|\boldsymbol{\mathcal{G}}\|_F^2 \\ &= \|\boldsymbol{\mathcal{X}}\|_F^2 - \|\boldsymbol{\mathcal{X}} \times_1 \mathbf{U}^\top \times_2 \mathbf{V}^\top \times_3 \mathbf{W}^\top\|_F^2.\end{aligned}$$

When we optimize Eq. (8) with respect to $\mathbf{U}$, by ignoring the terms unrelated to $\mathbf{U}$, we obtain an equivalent maximization problem of

$$\tilde{f}(\mathbf{U}) \equiv \|\boldsymbol{\mathcal{X}} \times_1 \mathbf{U}^\top \times_2 \mathbf{V}^\top \times_3 \mathbf{W}^\top\|_F^2 - \alpha \operatorname{tr}\left(\mathbf{U}^\top \mathbf{L}_1 \mathbf{U}\right). \quad (9)$$

Unfolding Eq. (9) by the first mode, we have

$$\tilde{f}(\mathbf{U}) = \|\mathbf{U}^\top \mathbf{X}_{(1)} \left(\mathbf{W} \otimes \mathbf{V}\right)\|_F^2 - \alpha \operatorname{tr}\left(\mathbf{U}^\top \mathbf{L}_1 \mathbf{U}\right).$$

Setting $\mathbf{S} \equiv \mathbf{X}_{(1)} \left(\mathbf{W} \otimes \mathbf{V}\right) = (\boldsymbol{\mathcal{X}} \times_2 \mathbf{V} \times_3 \mathbf{W})_{(1)}$, $\tilde{f}(\mathbf{U})$ is further rewritten as

$$\tilde{f}(\mathbf{U}) = \|\mathbf{U}^\top \mathbf{S}\|_F^2 - \alpha \operatorname{tr}\left(\mathbf{U}^\top \mathbf{L}_1 \mathbf{U}\right) = \operatorname{tr}\left(\mathbf{U}^\top \left(\mathbf{S}\mathbf{S}^\top - \alpha \mathbf{L}_1\right) \mathbf{U}\right). \quad (10)$$

The maximizer of Eq. (10) satisfying the orthogonality constraint $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$ is obtained as the I leading eigenvectors of $\mathbf{S}\mathbf{S}^\top - \alpha \mathbf{L}_1$.

3.3 Proposed Method 2: Cross-Mode Regularization

The within-mode regularization scheme regularizes the elements only inside each factor matrix, because each element of $\mathbf{U}$ interacts only with at most $I - 1$ elements within $\mathbf{U}$. This fact sometimes limits the effect of the regularization when we have bursty missing values, For example, slice-level missing situations where no observations are given for some objects often occurs in the context of recommender systems as the "cold-start" problem. In such cases, the within-mode regularization can be sometimes too conservative.

The second regularization function we propose exploits the given auxiliary information more aggressively. It combines the given similarity matrices to co-regularize combinations of elements across different modes as

$$R(\hat{\boldsymbol{\mathcal{X}}}; \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3) \equiv \operatorname{tr}\left((\mathbf{W} \otimes \mathbf{V} \otimes \mathbf{U})^\top \mathbf{L} \left(\mathbf{W} \otimes \mathbf{V} \otimes \mathbf{U}\right)\right), \quad (11)$$

where the $IJK \times IJK$-Laplacian matrix $\mathbf{L}$ is defined as

$$\mathbf{L} \equiv \mathbf{D}_3 \otimes \mathbf{D}_2 \otimes \mathbf{D}_1 - \mathbf{A}_3 \otimes \mathbf{A}_2 \otimes \mathbf{A}_1.$$

The regularization term Eq. (11) is rewritten with the matrix elements as

$$R(\hat{\boldsymbol{\mathcal{X}}}; \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3) \\ = \sum_{i,j,k=1}^{I,J,K} \sum_{\ell,m,n=1}^{I,J,K} [\mathbf{A}_1]_{i,\ell}[\mathbf{A}_2]_{j,m}[\mathbf{A}_3]_{k,n} \sum_{p,q,r=1}^{P,Q,R} \left([\mathbf{U}]_{i,p}[\mathbf{V}]_{j,q}[\mathbf{W}]_{k,r} - [\mathbf{U}]_{\ell,p}[\mathbf{V}]_{m,q}[\mathbf{W}]_{n,r}\right)^2,$$

which regularizes the combinations of elements from three different factors in contrast with the within-mode regularization (6) considering each mode independently.

The cross-mode regularization (11) can be seen as a natural variant of the within-mode regularization (5). Because, if we use the Kronecker sum $\oplus$ instead of the Kronecker product $\otimes$ in Eq. (11), it is reduced to Eq. (5) under the orthogonality constraints.

CP-decomposition. The objective function for the cross-mode regularized CP-decomposition is defined as

$$f(\mathbf{U}, \mathbf{V}, \mathbf{W}) \equiv \frac{1}{2}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{J}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}\|_F^2 \\ + \frac{\alpha}{2}\mathrm{tr}\left((\mathbf{W} \otimes \mathbf{V} \otimes \mathbf{U})^\top \mathbf{L}\,(\mathbf{W} \otimes \mathbf{V} \otimes \mathbf{U})\right).$$

Noting that Eq. (11) is simplified as

$$R(\hat{\boldsymbol{\mathcal{X}}}; \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3) = \mathrm{tr}\left(\mathbf{W}^\top \mathbf{D}_3 \mathbf{W}\right) \mathrm{tr}\left(\mathbf{V}^\top \mathbf{D}_2 \mathbf{V}\right) \mathrm{tr}\left(\mathbf{U}^\top \mathbf{D}_1 \mathbf{U}\right) \\ - \mathrm{tr}\left(\mathbf{W}^\top \mathbf{A}_3 \mathbf{W}\right) \mathrm{tr}\left(\mathbf{V}^\top \mathbf{A}_2 \mathbf{V}\right) \mathrm{tr}\left(\mathbf{U}^\top \mathbf{A}_1 \mathbf{U}\right),$$

similar to the within-mode regularization, we obtain the Sylvester equation for $\mathbf{U}$ as

$$\mathbf{U}(\mathbf{W} \odot \mathbf{V})^\top(\mathbf{W} \odot \mathbf{V}) + (D_{VW}\mathbf{D}_1 - A_{VW}\mathbf{A}_1)\,\mathbf{U} = \mathbf{X}_{(1)}(\mathbf{W} \odot \mathbf{V}),$$

where D_{VW} and A_{VW} are defined as follows.

$$D_{VW} \equiv \mathrm{tr}\left(\mathbf{W}^\top \mathbf{D}_3 \mathbf{W}\right) \mathrm{tr}\left(\mathbf{V}^\top \mathbf{D}_2 \mathbf{V}\right)$$
$$A_{VW} \equiv \mathrm{tr}\left(\mathbf{W}^\top \mathbf{A}_3 \mathbf{W}\right) \mathrm{tr}\left(\mathbf{V}^\top \mathbf{A}_2 \mathbf{V}\right)$$

Tucker Decomposition. The objective function for the cross-mode regularized Tucker decomposition is defined as

$$f(\boldsymbol{\mathcal{G}}, \mathbf{U}, \mathbf{V}, \mathbf{W}) \equiv \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}\|_F^2 \\ + \alpha\,\mathrm{tr}\left((\mathbf{W} \otimes \mathbf{V} \otimes \mathbf{U})^\top \mathbf{L}\,(\mathbf{W} \otimes \mathbf{V} \otimes \mathbf{U})\right).$$

Again, we alternately minimize the objective function with respect to one of $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$. By a similar derivation to that for the within-mode regularization, the optimal $\mathbf{U}$ with $\mathbf{V}$ and $\mathbf{W}$ fixed is obtained as the I leading eigenvectors of

$$\mathbf{S}\mathbf{S}^\top \mathbf{A} - \alpha\,(D_{VW}\mathbf{D}_1 - A_{VW}\mathbf{A}_1).$$

4 Experiments

We show some results of numerical experiments of third-order tensor completion problems using synthetic and real benchmark datasets, and demonstrate that introducing auxiliary information improves predictive accuracy especially when observations are sparse.

4.1 Datasets

Synthetic Dataset. The first dataset is synthetic tensors with correlated objects. We generate CP-decomposed tensors with $\mathbf{U} \in \mathbb{R}^{I \times R}, \mathbf{V} \in \mathbb{R}^{J \times R}$ and $\mathbf{W} \in \mathbb{R}^{K \times R}$ with the rank $R \equiv 2$ and $I \equiv J \equiv K \equiv 30$ by using the linear formulae,

$$\begin{aligned}
[\mathbf{U}]_{ir} &\equiv i\epsilon_r + \epsilon'_r \ (1 \leq i \leq I, 1 \leq r \leq R) \\
[\mathbf{V}]_{jr} &\equiv j\zeta_r + \zeta'_r \ (1 \leq j \leq J, 1 \leq r \leq R) \\
[\mathbf{W}]_{kr} &\equiv k\eta_r + \eta'_r \ (1 \leq k \leq K, 1 \leq r \leq R),
\end{aligned}$$

where $\{\epsilon_r, \epsilon'_r, \zeta_r, \zeta'_r, \eta_r, \eta'_r\}_{r=1}^{R}$ are constants generated by using the standard Gaussian distribution. A synthetic tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$ is defined as

$$\boldsymbol{\mathcal{X}} \equiv \boldsymbol{\mathcal{J}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}.$$

Since the columns of each factor matrix are generated by linear functions, the consecutive rows are similar to each other. Therefore, the similarity matrix for the i-th mode is naturally defined as the following tri-diagonal matrix.

$$\mathbf{A}_i \equiv \begin{bmatrix} 0 & 1 & 0 & \cdots \\ 1 & 0 & 1 & \cdots \\ 0 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Benchmark Dataset 1: Flow Injection. As a real benchmark dataset with auxiliary information, we used the 'Rank-deficient spectral FIA dataset'[2], which consists of results of flow injection analysis on 12 different chemical substances. They are represented as a tensor of size 12 (substances)×100 (wavelengths)×89 (reaction times).

We constructed three similarity matrices for the three modes as follows. Since 12 chemical substances differ in contents of three structural isomers of a certain chemical compound, each substance can be represented as a three-dimensional feature vector. We defined the similarity between two substances as the inverse of Euclidean distance between their feature vectors. Also, since wavelength and reaction time have continuous real values, we simply set the similarity of two consecutive wavelength values (or reaction time values) to one.

[2] The datasets are available from `http://www.models.life.ku.dk/datasets`

Benchmark Dataset 2: Licorice. Another benchmark dataset we use is the 'Three-way electronic nose dataset'[2], which consists of measurements of an odor sensing system applied to licorices for checking their quality, and is represented as a third-order tensor of size 18 (samples)×241 (reaction times)×12 (sensors).

Since each of 18 samples is labeled with one of the three quality labels, {'BAD','FBAD','GOOD'}, we set the similarity between two samples sharing an identical label to one. The similarity for reaction time is defined in the same way as for the flow injection dataset. Eventually, we obtained two similarity matrices for this dataset.

4.2 Experimental Settings

Comparison Methods. We compared the following 3 (regularization methods) ×2 (decomposition models) = 6 methods.

1. Ordinary {CP, Tucker}-decomposition,
2. Within-mode regularized {CP, Tucker}-decomposition,
3. Cross-mode regularized {CP, Tucker}-decomposition.

We used EM-based algorithms to impute missing values, especially, we used the one which updates missing value estimates each time a factor is updated [20], since it converged faster. We set the initial estimates for the unobserved elements of $\boldsymbol{\mathcal{X}}$ to the average of the observed elements, and those for $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ to the leading eigenvectors of $\mathbf{X}_{(1)}$, $\mathbf{X}_{(2)}$, and $\mathbf{X}_{(3)}$, respectively.

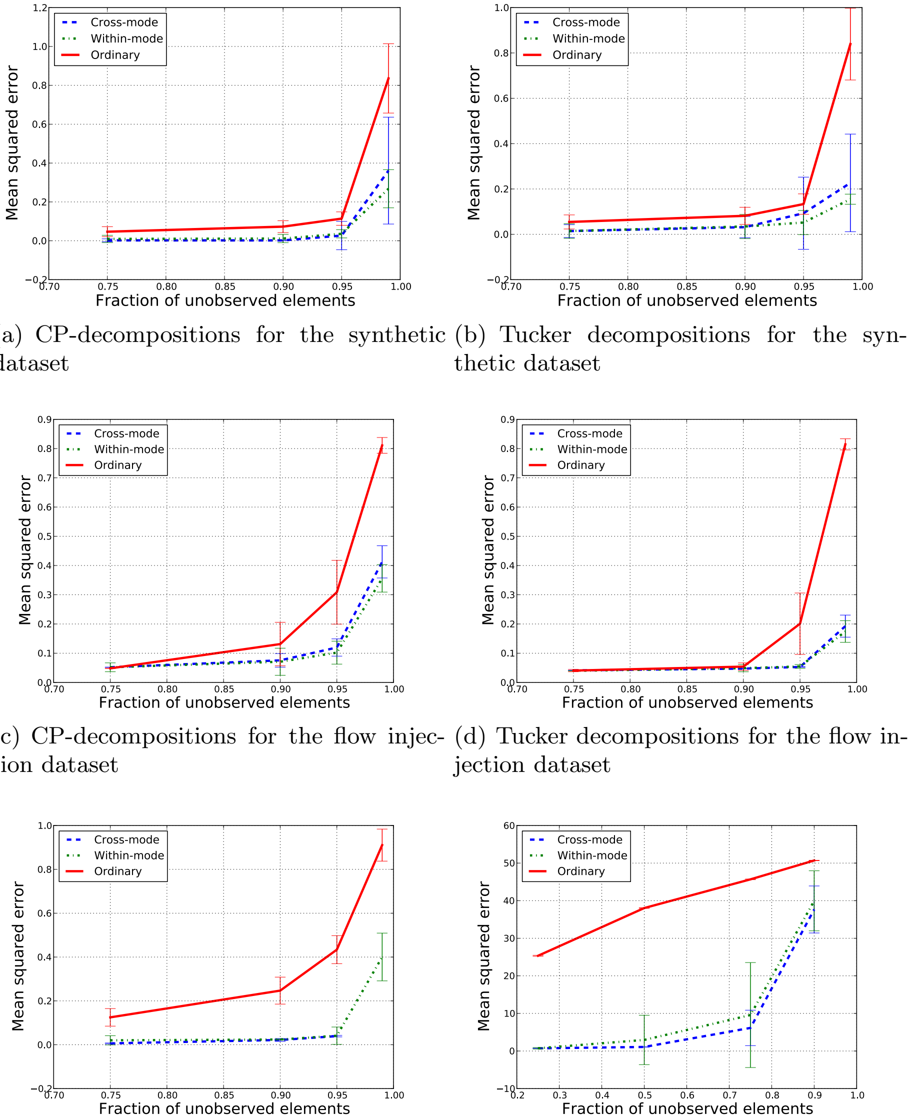
We set the model ranks as $P \equiv Q \equiv R \equiv 2$ for the synthetic dataset, $P \equiv Q \equiv R \equiv 4$ for the flow injection dataset, and $P \equiv Q \equiv R \equiv 3$ for the licorice dataset, based on the results of preliminary experiments. The hyper-parameter α was selected from $\left\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\right\}$ by using cross-validation.

Element-Wise Missing vs. Slice-Wise Missing. We test two kinds of assumptions on missing elements, that are, element-wise missing and slice-wise missing. In the element-wise missing setting, each element $[\boldsymbol{\mathcal{X}}]_{i,j,k}$ is missing independently. On the other hand, in the slice-wise missing setting, missing values occur at object level, and therefore all of the elements related to some objects are totally missing. In other words, slices such as $\{[\boldsymbol{\mathcal{X}}]_{i,j,k}\}_{j,k}$ for some i, $\{[\boldsymbol{\mathcal{X}}]_{i,j,k}\}_{i,k}$ for some j, and $\{[\boldsymbol{\mathcal{X}}_{i,j,k}]\}_{i,j}$ for some k are missing, which means that missing values occurring in a more bursty manner.

We varied the fraction of unobserved elements among $\{0.75, 0.9, 0.95, 0.99\}$ for the element-wise missing setting, and among $\{0.5, 0.75, 0.9, 0.95\}$ for the slice-wise missing setting. We randomly selected elements or objects to be used as the unobserved parts, and evaluated the mean squared errors between true values and predicted values. We continued the evaluation ten times, and recorded the averaged errors and their standard errors.

4.3 Results

Figure 4 shows the accuracy of tensor completion by the six methods ({'Ordinary', 'Within-mode', 'Cross-mode'}×{CP-decomposition, Tucker-decomposition}) for



(a) CP-decompositions for the synthetic dataset

(b) Tucker decompositions for the synthetic dataset

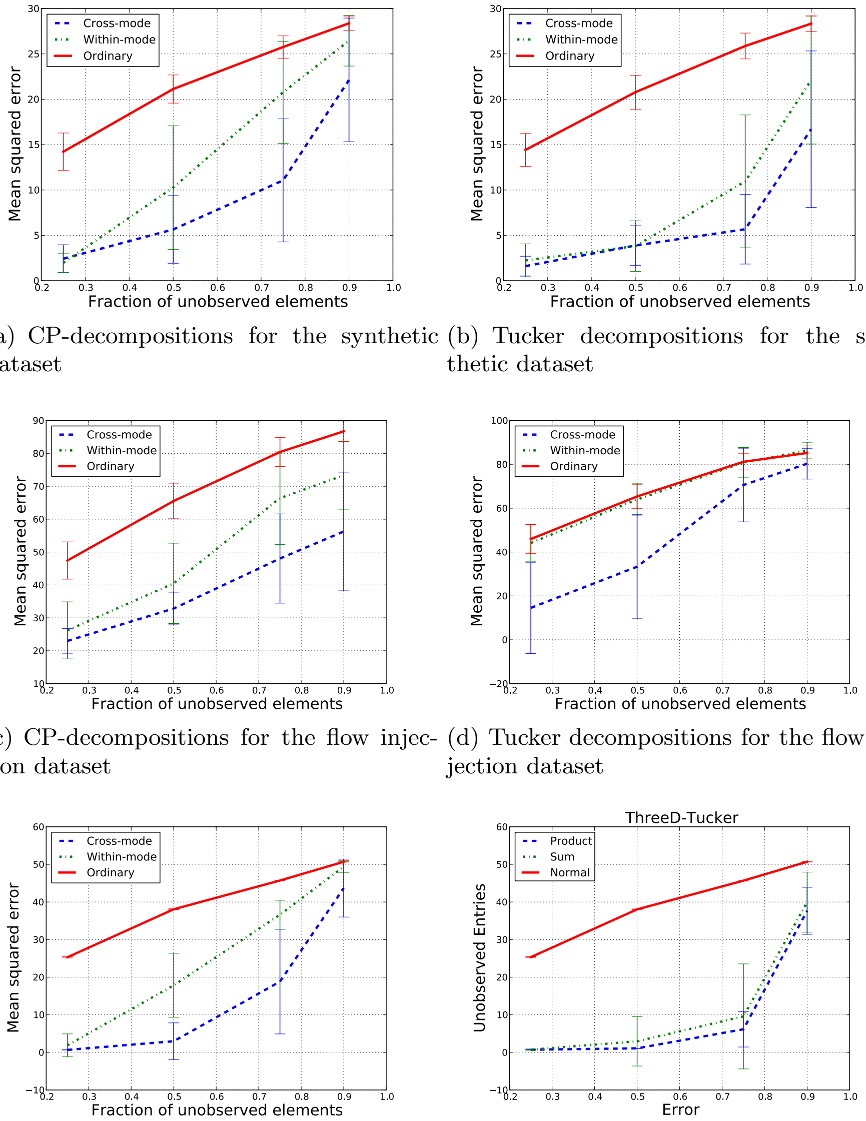
(c) CP-decompositions for the flow injection dataset

(d) Tucker decompositions for the flow injection dataset

(e) CP-decompositions for the licorice dataset

(f) Tucker decompositions for the licorice dataset

Fig. 4. Accuracy of tensor completion for three datasets in the *element-wise* missing setting. The proposed methods perform well when observations are sparse.



(a) CP-decompositions for the synthetic dataset

(b) Tucker decompositions for the synthetic dataset

(c) CP-decompositions for the flow injection dataset

(d) Tucker decompositions for the flow injection dataset

(e) CP-decompositions for the licorice dataset

(f) Tucker decompositions for the licorice dataset

Fig. 5. Accuracy of tensor completion for three datasets in the *slice-wise* missing setting. The cross-mode regularization method performs especially well when observations are sparse.

three datasets (the synthetic dataset, the flow injection dataset, and the licorice dataset) in the element-wise missing setting. Overall, we can see that incorporating auxiliary information improves the predictive accuracy, although the ordinary tensor decomposition methods still work fairly well when the fraction of unobserved elements is less than 95%. The proposed methods perform well especially when observations are sparse.

On the other hand, Figure 5 shows the results for the slice-wise missing setting. In this case, since missing values occur at object level, reasonable inference is not possible only with the low-rank assumption, and the performance severely gets worse in sparse cases. When with auxiliary information, especially, the cross-mode regularization keeps its performance compared with the other methods even in the bursty sparse cases. This is because the cross-mode regularization uses the auxiliary information more aggressively than the within-mode regularization, that is, each element of $\mathbf{U}$ interacts with at most $I - 1$ other elements in $\mathbf{U}$ in the within-mode regularization, while it interacts with all of the other elements in all of $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ in the cross-mode regularization.

Finally, we briefly mention the computation time. Although introducing auxiliary information slightly increases the time and space complexity of the decomposition algorithms, the actual computation time was almost as same as that for ordinary decomposition methods (without auxiliary information). This is partially because we used relatively small datasets in the experiments, and further investigation with larger datasets should be made in future work.

5 Related Work

Tensor factorization methods have recently been studied extensively, and widely applied in the data mining communities. CANDECOMP/PARAFAC(CP)-decomposition [7] is a tensor factorization method that can be seen as a special case of Tucker decomposition [19] by making its core tensor super-diagonal. The CP-decomposition is applied to various problems including chemo-informatics [10]. Its variant called pair-wise interaction tensor factorization [15] accelerates its computation by using the stochastic gradient descent, and is applied to a large-scale tag recommendation problem.

There also exist probabilistic extensions of tensor factorization methods. Shashua and Hazan [17] studied the PARAFAC model under the non-negativity constraint with latent variables. Chu and Ghahramani [4] proposed a probabilistic extension of the Tucker method, known as pTucker.

Although we focus on the squared loss function (3) in this paper, changing the loss function corresponds to non-Gaussian probabilistic models of tensor factorization. In several matrix and tensor factorization studies, non-Gaussian observations have been dealt with. Collins *et al.* [5] generalized the likelihood of the probabilistic PCA to the exponential family, which was further extended to tensors by Hayashi *et al.* [8].

There are several studies to incorporate auxiliary information into matrix factorization. Li *et al.* [12] introduced a regularizer for one of factor matrices by a graph Laplacian based on geometry of data distribution. A similar approach is proposed by Cai *et al.* [3]. Lu *et al.* [13] proposed incorporated both spatial and temporal information by using graph Laplacian and Kalman filter. Adams *et al.* [2] extended the probabilistic matrix factorization [16] to incorporate side information. In their work, Gaussian process priors are introduced to enforce smoothness to factors. Some work use auxiliary information not in regularization terms but as bias variables added to model parameters [21,14]. To best of our knowledge, our work is the first attempt to incorporate auxiliary information into tensor factorization.

6 Conclusion and Future Work

In this paper, we proposed to use relationships among data as auxiliary information in addition to the low-rank assumption to improve accuracy of tensor factorization. We introduced two regularization approaches using graph Laplacians induced from the relationships, and designed approximate solutions for the optimization problems. Numerical experiments using synthetic and real datasets showed that the use of auxiliary information improved completion accuracy over the existing methods based only on the low-rank assumption, especially when observations were sparse.

Although the focus of this paper is to show the usefulness of auxiliary information for tensor factorization, we will address its computational aspects extensively in future. Indeed, scalability is an important issue. In real data such as EEG data, tensors can be huge and of high dimensionality, and we need fast and memory efficient algorithms. For example, Acar *et al.* [1] eliminate some of the observed elements of large data tensors. They report entire information of the data tensor is not necessary when the tensor is of low-rank, and only a small fraction of elements ($\sim 0.5\%$ for a $1,000 \times 1,000 \times 1,000$ tensor) are sufficient for reconstruction. The idea might also work well in our approach, and the regularization using auxiliary information might further reduce the sufficient number of elements. Memory efficiency is also an important factor, since the number of parameters are linearly dependent on the dimensionality of each mode. Kolda and Sun [11] use the sparsity of tensor, and dramatically reduce the memory space as 1,000 times smaller than the original algorithm for Tucker decomposition. The stochastic gradient optimization approach [15] would be also promising.

References

1. Acar, E., Dunlavy, D.M., Kolda, T.G., Mørup, M.: Scalable tensor factorizations with missing data. In: Proceedings of the 2010 SIAM International Conference on Data Mining, pp. 701–712 (2010)
2. Adams, R.P., Dahl, G.E., Murray, I.: Incorporating side information into probabilistic matrix factorization using Gaussian processes. In: Grünwald, P., Spirtes, P. (eds.) Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, pp. 1–9 (2010)
3. Cai, D., He, X., Han, J., Huang, T.S.: Graph regularized non-negative matrix factorization for data representation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2010)

4. Chu, W., Ghahramani, Z.: Probabilistic models for incomplete multi-dimensional arrays. In: Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (2009)
5. Collins, M., Dasgupta, S., Schapire, R.E.: A generalization of principal components analysis to the exponential family. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) Advances in Neural Information Processing Systems, vol. 14, MIT Press, Cambridge (2002)
6. Dunlavy, D.M., Kolda, T.G., Acar, E.: Temporal link prediction using matrix and tensor factorizations. ACM Transactions on Knowledge Discovery from Data 5, 10:1–10:27 (2011)
7. Harshman, R.A.: Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multi-modal factor analysis. UCLA Working Papers in Phonetics 16(1), 84 (1970)
8. Hayashi, K., Takenouchi, T., Shibata, T., Kamiya, Y., Kato, D., Kunieda, K., Yamada, K., Ikeda, K.: Exponential family tensor factorization for missing-values prediction and anomaly detection. In: Proceedings of the 10th IEEE International Conference on Data Mining, pp. 216–225 (2010)
9. Kashima, H., Kato, T., Yamanishi, Y., Sugiyama, M., Tsuda, K.: Link propagation: A fast semi-supervised algorithm for link prediction. In: Proceedings of the 2009 SIAM International Conference on Data Mining (2009)
10. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Review 51(3), 455–500 (2009)
11. Kolda, T.G., Sun, J.: Scalable tensor decompositions for multi-aspect data mining. In: Proceedings of the 8th IEEE International Conference on Data Mining, pp. 363–372 (2008)
12. Li, W.-J., Yeung, D.-Y.: Relation regularized matrix factorization. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, pp. 1126–1131 (2009)
13. Lu, Z., Agarwal, D., Dhillon, I.S.: A spatio-temporal approach to collaborative filtering. In: Proceedings of the 3rd ACM Conference on Recommender Systems, pp. 13–20 (2009)
14. Porteous, I., Asuncion, A., Welling, M.: Bayesian matrix factorization with side information and Dirichlet process mixtures. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence, pp. 563–568 (2010)
15. Rendle, S., Thieme, L.S.: Pairwise interaction tensor factorization for personalized tag recommendation. In: Proceedings of the 3rd ACM International Conference on Web Search and Data Mining, pp. 81–90 (2010)
16. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: Platt, J.C., Koller, D., Singer, Y., Roweis, S. (eds.) Advances in Neural Information Processing Systems, vol. 20. MIT Press, Cambridge (2008)
17. Shashua, A., Hazan, T.: Non-negative tensor factorization with applications to statistics and computer vision. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 792–799 (2005)
18. Srebro, N.: Learning with matrix factorizations. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (2004)
19. Tucker, L.: Some mathematical notes on three-mode factor analysis. Psychometrika 31(3), 279–311 (1966)
20. Walczak, B.: Dealing with missing data Part I. Chemometrics and Intelligent Laboratory Systems 58(1), 15–27 (2001)
21. Yu, K., Lafferty, J., Zhu, S., Gong, Y.: Large-scale collaborative prediction using a nonparametric random effects model. In: Proceedings of the 26th International Conference on Machine Learning, pp. 1185–1192 (2009)