Tensor Completion for Weakly-Dependent Data on Graph for Metro Passenger Flow Prediction

Ziyue Li,[1]* **Nurettin Dorukhan Sergin,**[2]
Hao Yan,[3] **Chen Zhang,**[4] **Fugee Tsung**[5]
[1,5]The Hong Kong University of Science and Technology
[2,3]Arizona State University, [4]Tsinghua University

Abstract

Low-rank tensor decomposition and completion have attracted significant interest from academia given the ubiquity of tensor data. However, low-rank structure is a global property, which will not be fulfilled when the data presents complex and weak dependencies given specific graph structures. One particular application that motivates this study is the spatiotemporal data analysis. As shown in the preliminary study, weakly dependencies can worsen the low-rank tensor completion performance. In this paper, we propose a novel low-rank CANDECOMP / PARAFAC (CP) tensor decomposition and completion framework by introducing the L_1-norm penalty and Graph Laplacian penalty to model the weakly dependency on graph. We further propose an efficient optimization algorithm based on the Block Coordinate Descent for efficient estimation. A case study based on the metro passenger flow data in Hong Kong is conducted to demonstrate an improved performance over the regular tensor completion methods.

Introduction

Higher-order tensors have been actively used in research since they have an inclination to successfully preserve the complicated innate structural properties of data(Kolda and Bader 2009). A tensor can be defined mathematically as the multi-dimensional arrays. The order of a tensor is the number of dimensions, also known as modes. 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. Specifically, spatiotemporal data can normally be modeled as high-order tensors with spatial and temporal modes; tensor analysis for spatiotemporal data is widely used for prediction and feature extractions (Sun and Axhausen 2016; Dunlavy, Kolda, and Acar 2011). Tensor completion can be also used as a prediction method, with the prediction horizon regarded as missing entries, and historical data as observed entries (Tan et al, 2016, Luan, Zhang 2018).

*We show our great appreciation to the Metro Corporation for sharing this passenger flow data, and in the protection of privacy, all data has been desensitized.

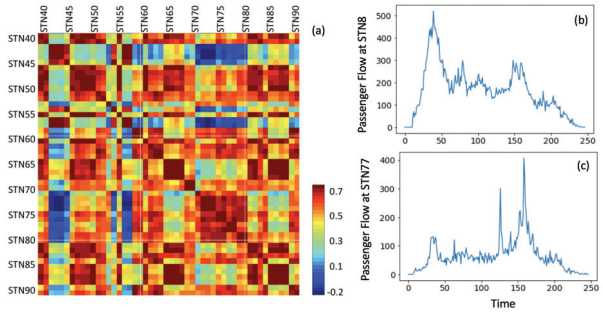

Figure 1: (a) The weakly-dependent property of each stations (b) Passenger Flow at Station 8 (c) Passenger Flow at Station 77

However, tensor completion of spatiotemporal data is a challenging task due to its complex spatial and temporal dependencies. Take the metro passenger flow tensor as an example, for temporal dependence, the prediction of a certain time is usually correlated with the historical observations. Passenger flow patterns also display a strong periodicity with period as 7 days.

For spatial dependency, graph structures are commonly used. For example, a geographical graph is often defined based on its spatially adjacent neighbours. Furthermore, a contextual graph is often defined based on the contextual similarity, which quantifies whether two stations share a similar function (e.g., business center, residential area or school etc.). In the relevant literature, it has been highlighted that both the geographical graph and contextual graphs significantly affect passenger flow patterns at station points (Zhong et al. 2017).

Taking the metro passenger flow data as example, denoted as three-mode tensor $\boldsymbol{\mathcal{X}}^{station \times point \times day}$, as a result of the geographical and contextual graphs along the station mode, if any two stations are picked, they are less likely to be highly-dependent to each other if they are not connected in either of the two graphs, such as Station 8 and Station 77 in Figure 1(b), (c) having quite different passenger flow profile. With the amount of stations increasing, the chance that all stations are strongly dependent is very small. This data

structure is defined as the weakly dependency on graphs, which has also been explored in other settings and applications (Zhang et al. 2018).

Here, we like to give a formal definition of the *Weakly-Dependent data on Graph (WDG)*. Intially, a graph or multiple graphs can be defined on the entities along a specific mode of a tensor. The tensor data is *WDG* only if it is linear dependent on the neighborhood of the graph. We can further define this mode as a *weakly-dependent mode*, and this data as *weakly-dependent data*. For example, in the previous metro passenger flow example, the station is the weakly dependent mode.

In the literature, the Low Rank Tensor Completion (LRTC) method is commonly used for a spatio-temporal data prediction, which fills in the incomplete tensor based on the low-rank tensor constraint. However, the low-rank property implies that data in all stations are linearly dependent, whereas the WDG is a local structure, which implies the data are only dependent on the neighborhood of the graph. Therefore, applying LRTC often yields an oversimplified model (Zhao, Zhang, and Cichocki 2015).

In conclusion, we proposed a novel regularized CANDECOMP / PARAFAC (CP) tensor decomposition and completion framework considering the WDG property. It has been shown that the suggested framework is able to simultaneously achieve not only high accuracy considering both multi-networks but also a high degree of weak dependency. In summary, this paper makes following contributions:

- Formulate a Low Rank Tensor Completion with weakly-dependent pattern on graphs based on CP Decomposition.
- Propose an efficient optimization algorithm for the proposed framework
- Demonstrate the performance of the proposed framework in metro passenger flow data

Literature Review

Tensor Decomposition and Regularization: Tensor decomposition has been widely studied in the available literature. Specifically, CANDECOMP / PARAFAC (CP) decomposition represents a tensor as the weighted summation of a set of rank-one tensors. For more detailed introduction to tensor decomposition and CP decomposition, please refer to Kolda and Bader's work (2009).

$$\begin{aligned}\mathcal{X} &= \sum_{r=1}^{R} \lambda_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \cdots \circ \mathbf{u}_r^{(K)} \\ &= [[\lambda; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, ..., \mathbf{U}^{(K)}]]\end{aligned} \tag{1}$$

where $\mathbf{U}^{(k)}$ is the decomposed factor along k-th mode, which can be termed as the k-th mode factor.

To incorporate the domain knowledge into the tensor decomposition framework, constrained or regularized decomposition is commonly used. Some of the proposed frameworks include non-negative tensor decomposition (Mørup, Hansen, and Arnfred 2008), sparse tensor decomposition with $L_{2,1}$-norm penalty (Yu et al. 2019), and smooth tensor decomposition with total variation regularization (Yokota, Zhao, and Cichocki 2016), and a graph Laplacian penalty for tensor completion(Wang et al. 2015).

Tensor completion: In terms of missing data imputation in a tensor, the state-of-the-art tensor completion methodology can be summarized into two major types.

The first type is to define the low-rank penalty directly onto the original tensor. The benefit is that these frameworks are typically convex and can be optimized globally. One of the most popular methods involves the use of the tensor-version of the nuclear norm, which is defined as the sum of the nuclear norms of the all the unfolded matrices of $\mathcal{X}$ (Liu et al. 2012) For example, Simple Low-Rank Tensor Completion (SiLRTC) defined the tensor completion problem as follows:

$$\begin{aligned}\min_{\mathcal{X}, \mathbf{Y}_{(1)}, ..., \mathbf{Y}_{(K)}} & \sum_{k=1}^{K} \alpha_k \|\mathbf{Y}_{(k)}\|_* + \frac{\beta}{2} \|\mathbf{X}_{(k)} - \mathbf{Y}_{(k)}\|_F^2 \\ \textit{s.t. } & P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{T})\end{aligned} \tag{2}$$

where $\mathcal{X}$ is the completed output tensor, $\mathcal{T}$ is the incomplete input tensor, and $\{\mathbf{Y}_{(k)}\}_{k=1,...,K}$ are the low-rank matrices corresponding to k-th mode unfolded tensor $\mathbf{X}_{(k)}$.

However, nuclear norm minimization is often very computationally expensive and typically involves singular value decomposition of very large matrices (Kressner, Steinlechner, and Vandereycken 2014).

Furthermore, another drawback of first type is that by using tensor decomposition, the decomposed factors usually carry some quite meaningful physical information which reflects a number of features in reality (Sun and Axhausen 2016). Sun noted the peak hour pattern, dominant passenger type, highly-traveled area based on the decomposed factors along time, passenger type and location mode respectively. Inversely, if the auxiliary information from the physical world is available, then formulating a completion without a decomposition structure equated to being incapable of adding the prior information to the model. This shortage can be overcome by adopting a second approach, the completion based on a decomposition structure. With this, the prior information can be easily applied to the decomposed latent factors through proper regularization.

Simultaneous Tensor Decomposition and Completion: The second approach is Simultaneous Tensor Decomposition and Completion (STDC), which is defined by Chen, Hsu, and Liao. STDC estimates the decomposed latent factors using partially observed data (Chen, He, and Sun 2019). It is then possible to compute the missing entry from the estimated latent factors.

In the STDC framework, factor prior or regularization is often applied on decomposed components based on low-rank structure (Chen, Hsu, and Liao 2013). For example, in the available literature, a Bayesian hierarchical probabilistic structure for CP decomposition (Zhao, Zhang, and Cichocki 2015) and Tucker decomposition (Zhao, Zhang and Cichocki 2016) was proposed, which assumed that the decomposed factor matrices were generated from a same high-level Gaussian distribution. Through this the same sparsity

pattern can be obtained in latent matrices, yielding the minimum number of rank-one tensor . Low rank penalties such as L_1-norm penalty on the CP decomposition weight vector or nuclear-norm on the Tucker decomposition core tensor have also been proposed (Shi, Lu, and Cheung 2017).

However, as mentioned before, low-rank property can not be satisfied. Introducing the WDG into the tensor completion is new challenge in the literature. In our preliminary result, we have shown that the Bayesian Low-Rank Tensor Completion performs much worse in weakly-dependent data than in strongly-dependent data. The WDG data thus presents a more diverse and complicated spatiotemporal dependency structure.

Formulation

We will formulate the tensor completion problem given the WDG data structure.

Notations and Operations

Through out this exposition, scalars are denoted in italics, e.g. n; vectors by lowercase letters in boldface, e.g. $\mathbf{u}$; and matrices by uppercase boldface letters, e.g. $\mathbf{U}$; High dimensional data, tensor by boldface script capital $\boldsymbol{\mathcal{X}}$.

For a given incomplete tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, we assume its first mode is WDG mode.

CP Decomposition Based Tensor Completion

(1) Low-Rank Tensor Completion

The low-rank tensor completion based on CP decomposition based on (1) can be formulated by L_1-norm on the weight vector:

$$\begin{aligned} \min_{R,\boldsymbol{\mathcal{Y}},\boldsymbol{\lambda},\{\mathbf{u}_r^{(k)}\}} & \frac{1}{2}\|\boldsymbol{\mathcal{Y}} - \sum_{r=1}^{R} \lambda_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \cdots \circ \mathbf{u}_r^{(K)}\|^2 \\ & + \beta\|\boldsymbol{\lambda}\|_1 \\ \textit{s.t. } & P_\Omega(\boldsymbol{\mathcal{Y}}) = P_\Omega(\boldsymbol{\mathcal{X}}) \\ & {\mathbf{u}_r^{(k)}}^T \mathbf{u}_r^{(k)} = 1, \quad \forall r \end{aligned} \tag{3}$$

where $\boldsymbol{\mathcal{X}}$ is the incomplete input tensor with the observation indicator tensor as Ω, and $\boldsymbol{\mathcal{Y}}$ is the complete output tensor. $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, ..., \lambda_R]^T$ is the weight vector of CP decomposition. L_1-norm penalty on $\boldsymbol{\lambda}$ helps control the rank of the tensor.

(2) Weak Dependency

Equation (3) assumes a global low rank property within the entire tensor. The low-rank structure holds in general when data are strongly-dependent. However, it does not hold for WDG data since this weak dependence or various features indicate that data may come from different sources or clusters. A relevant example here would be metro passenger flow pattern of different areas. Borrowing the idea of sparse coding (Lee et al. 2007), it is often more desirable to set $\mathbf{U}_1 \in \mathbb{R}^{I_1 \times R}$ to be sparse to force scores on unrelated entities to be zero (Zhang et al. 2018). Review the previous example of metro passenger flow tensor: based on their functionality (e.g., for commercial, entertainment, or residential use), all stations in the metro system can be regarded as coming from c clusters, which have strong within-cluster dependence but little or no cross-cluster dependence. Directly using the low-rank tensor completion for all the I_1 stations will achieve poor completion and decomposition since the low-rank tensor completion may use features of stations from unrelated clusters and can eventually contaminate the tensor completion result. By introducing sparsity on $\mathbf{U}_1$, we represent a station's profile using data from the few selected stations (e.g., within the same cluster).

$$\begin{aligned} \min_{R,\boldsymbol{\mathcal{Y}},\boldsymbol{\lambda},\{\mathbf{u}_r^{(k)}\}} & loss + \alpha\|\mathbf{U}_1\|_1 + \beta\|\boldsymbol{\lambda}\|_1 \\ \textit{s.t. } & P_\Omega(\boldsymbol{\mathcal{Y}}) = P_\Omega(\boldsymbol{\mathcal{X}}) \\ & {\mathbf{u}_r^{(k)}}^T \mathbf{u}_r^{(k)} = 1, \quad \forall r \end{aligned} \tag{4}$$

where $loss = \frac{1}{2}\|\boldsymbol{\mathcal{Y}} - \sum_{r=1}^{R} \lambda_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \cdots \circ \mathbf{u}_r^{(K)}\|^2$. Besides, if the WDG data has multiple weakly-dependent modes, this L_1 penalty can also be easily extended into $\alpha_1\|\mathbf{U}_1\|_1 + \alpha_2\|\mathbf{U}_2\|_1 + \ldots$

(3) Graph Structure

When auxiliary information is available, for example, the graph information of each entity in weakly-dependent mode is obtainable, then a more explicitly defined dependence across each entity can be formulated. For the graph $\mathbf{G}_i$, which contains I_1 entities or nodes inside, an corresponding Adjacency Matrix $\mathbf{A}_i \in \mathbb{R}^{I_1 \times I_1}$ can be obtained. According to the adjacency matrix, a penalty which force two entities close or similar to each other into sharing similar pattern is desired. With this in mind, a Laplacian penalty (Wang et al. 2015; Yu et al. 2019) is a sensible choice. In Wang's work of citywide traffic tensor, the traffic pattern of different road segments depends on road segment correlation, so the penalty $tr(\mathbf{U}^T\mathbf{L}\mathbf{U})$ is added with $\mathbf{L}$ as the Laplacian Matrix of correlation matrix (Wang et al. 2015). Therefore, we define the Laplacian Matrix of $\mathbf{A}_i$ as $\mathbf{L}_i$. $\mathbf{L}_i = \mathbf{D}_i - \mathbf{A}_i$ and $\mathbf{D}_i$ is a diagonal matrix with element $d_{i_1,i_2} = \sum_{i_1} a_{i_1,i_2}$. Consequently, the formulation with graph information is:

$$\begin{aligned} \min_{R,\boldsymbol{\mathcal{Y}},\boldsymbol{\lambda},\{\mathbf{u}_r^{(k)}\}} & loss + \alpha\|\mathbf{U}_1\|_1 + \beta\|\boldsymbol{\lambda}\|_1 + \frac{1}{2}\gamma \cdot \\ & + tr(\mathbf{U}_1^T\mathbf{L}_i\mathbf{U}_1) \\ \textit{s.t. } & P_\Omega(\boldsymbol{\mathcal{Y}}) = P_\Omega(\boldsymbol{\mathcal{X}}) \\ & {\mathbf{u}_r^{(k)}}^T \mathbf{u}_r^{(k)} = 1, \quad \forall r \end{aligned} \tag{5}$$

The penalty $tr(\mathbf{U}_1^T\mathbf{L}_i\mathbf{U}_1)$is obtained by considering two entities i_1 and i_2 with higher similarity (i.e. a_{i_1,i_2} is bigger) should have a closer distance between vector $\mathbf{u}_{i_1}$ and $\mathbf{u}_{i_2}$ in the matrix $\mathbf{U}_1$. When multiple graphs are involved, the formulation can also be easily extended by adding more Laplacian penalties. It is worth noting that the graph definition differs from case to case, and will be clearly explained in the following case study.

Efficient Optimization Algorithm

We propose to use the Block Coordinate Descent (BCD) method to solve our optimization problem. The latent factors

and complete tensor to be estimated are divided into $R+1$ blocks: $\lambda_1, \mathbf{u}_1^{(1)}, \mathbf{u}_1^{(2)}, ...\mathbf{u}_1^{(K)}, ..., \lambda_R, \mathbf{u}_R^{(1)}, \mathbf{u}_R^{(2)}, ...\mathbf{u}_R^{(K)}, \mathcal{Y}$. The updating policy is: to update a block of variables which fixing the other blocks, and to one variable which fixing the other variables in each block.

Algorithm 1 Weakly-dependent on graph LRTC-CP

Input: incomplete tensor $\mathcal{X}$, $\mathbf{\Omega}$, $\mathbf{L}$, initial rank R, maximum iteration max_iter, α, β, γ, stopping tolerance tol.

Initialization $\boldsymbol{\mathcal{Y}}_\Omega = \boldsymbol{\mathcal{X}}_\Omega, \boldsymbol{\mathcal{X}}_{\Omega^c} = 0$, randomly initialize $\boldsymbol{\lambda}, \{\mathbf{U}_k\}_{k=1,...,K}, \boldsymbol{\mathcal{Y}}_{\Omega^c} = \{\boldsymbol{\lambda} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} ... \times_K \mathbf{U}^{(K)}\}_{\Omega^c}$.
for $i = 1$ to max_iter **do**
  for $r = 1$ to R **do**
    calculate $\boldsymbol{\mathcal{Y}}_r$ by (7).
    if $\lambda_r \neq 0$ **then**
      Update $\mathbf{u}_r^{(1)}$ by eq. (10), normalize $\mathbf{u}_r^{(1)}$.
      Update $\mathbf{u}_r^{(k)}$ by eq. (11), normalize $\mathbf{u}_r^{(k)}$, $k \neq 1$.
      Update λ_r by (14).
      Update $\boldsymbol{\mathcal{Y}}_r$ by eq. (16).
    else
      Do nothing.
    end if
  end for
  Update $\boldsymbol{\mathcal{Y}}_\Omega$ by eq. (16).
  Update $R = nonzero(\boldsymbol{\lambda})$.
  if $loss_{i-1} - loss_i < tol$ **then**, break; Otherwise, continue.
  end if
end for

Output: $\boldsymbol{\mathcal{Y}}, R, \boldsymbol{\lambda}, \{\mathbf{U}_k\}_{k=1,...,K}$.

The Lagrangian function of (6) regarding to the r-th block $\lambda_r, \mathbf{u}_r^{(1)}, \mathbf{u}_r^{(2)}, ...\mathbf{u}_r^{(K)}$ is:

$$\begin{aligned} L_{\lambda_r, \mathbf{u}_r^{(k)}} = &\frac{1}{2}\|\boldsymbol{\mathcal{Y}}_r - \lambda_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \cdots \circ \mathbf{u}_r^{(K)}\|^2 \\ &+ \alpha|\mathbf{u}_r^{(1)}| + \beta|\lambda_r| + \frac{1}{2}\gamma \cdot tr({\mathbf{u}_r^{(1)}}^T \mathbf{L} \mathbf{u}_r^{(1)}) \end{aligned} \tag{6}$$

where

$$\boldsymbol{\mathcal{Y}}_r = \boldsymbol{\mathcal{Y}} - \sum_{q \neq r} \lambda_q \mathbf{u}_q^{(1)} \circ \mathbf{u}_q^{(2)} \circ \cdots \circ \mathbf{u}_q^{(K)} \tag{7}$$

is the residual of the approximation.

Proposition 1: To update $\mathbf{u}_r^{(1)}$:

$$\begin{aligned} L_{\mathbf{u}_r^{(1)}} = &\frac{1}{2}\|\boldsymbol{\mathcal{Y}}_r - \lambda_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \cdots \circ \mathbf{u}_r^{(K)}\|^2 \\ &+ \alpha|\mathbf{u}_r^{(1)}| + \frac{1}{2}\gamma \cdot tr({\mathbf{u}_r^{(1)}}^T \mathbf{L} \mathbf{u}_r^{(1)}) \\ &- v \cdot ({\mathbf{u}_r^{(1)}}^T \mathbf{u}_r^{(1)} - 1) \end{aligned} \tag{8}$$

where v is the Lagrangian multiplier. Then partial derivative on $\mathbf{u}_r^{(1)}$:

$$\begin{aligned} \frac{\partial L_{\mathbf{u}_r^{(1)}}}{\partial \mathbf{u}_r^{(1)}} = &(\boldsymbol{\mathcal{Y}}_r - \lambda_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ ... \circ \mathbf{u}_r^{(K)}) \\ &\cdot (-\lambda_r \mathbf{u}_r^{(2)} \circ ... \circ \mathbf{u}_r^{(K)}) + \alpha \frac{|\mathbf{u}_r^{(1)}|}{\partial \mathbf{u}_r^{(1)}} \\ &+ \gamma \cdot L \cdot \mathbf{u}_r^{(1)} - 2v\mathbf{u}_r^{(1)} \end{aligned} \tag{9}$$

By setting the partial derivative as zero, $\mathbf{u}_r^{(1)}$ can be solved in the closed form solution as:

$$\begin{aligned} \mathbf{u}_r^{(1)} = &shrink_\alpha((\lambda_r^2 \mathbf{I} + \gamma \mathbf{L})^{-1} \\ &\cdot (\lambda_r \boldsymbol{\mathcal{Y}}_r \times_2 \mathbf{u}_r^{(2)} ... \times_K \mathbf{u}_r^{(K)})) \end{aligned} \tag{10}$$

where $shrink_a(\cdot)$ is the soft-thresholding operator where α is the regularization parameter. Then we consider the normalization: $\mathbf{u}_r^{(1)} \leftarrow \frac{\mathbf{u}_r^{(1)}}{\|\mathbf{u}_r^{(1)}\|_2}$.

Proposition 2: To update $\mathbf{u}_k^{(r)}, k = 2, 3, ..., K$:

$$\begin{aligned} L_{\mathbf{u}_r^{(k)}} = &\frac{1}{2}\|\boldsymbol{\mathcal{Y}}_r - \lambda_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \cdots \mathbf{u}_r^{(K)}\|^2 \\ &- \gamma({\mathbf{u}_r^{(k)}}^T \mathbf{u}_r^{(k)} - 1) \end{aligned} \tag{11}$$

Similarly we ignore the normalization constraint first by moving normalization step after solving its derivative equation. By setting the partial derivative as zero, we can have:

$$\mathbf{u}_r^{(k)} = \frac{\boldsymbol{\mathcal{Y}}_r \times_j \{\mathbf{u}_r^{(j)}\}_{j \neq k}}{\lambda_r} \tag{12}$$

Proposition 3: To update λ_r:

$$L_{\lambda_r} = \frac{1}{2}\|\boldsymbol{\mathcal{Y}}_r - \lambda_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \cdots \circ \mathbf{u}_r^{(K)}\|^2 + \beta|\lambda_r| \tag{13}$$

By setting the derivative on λ_r as zero, we can have:

$$\lambda_r = shrink_\beta(\boldsymbol{\mathcal{Y}}_r \times_1 \mathbf{u}_r^{(1)} \times_2 \mathbf{u}_r^{(2)} \times ... \times_K \mathbf{u}_r^{(K)}) \tag{14}$$

Proposition 4: To update $\mathcal{Y}$:

$$\begin{aligned} &\min_{\boldsymbol{\mathcal{Y}}} \frac{1}{2}\|\boldsymbol{\mathcal{Y}} - \sum_{r=1}^{R} \lambda_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ ... \circ \mathbf{u}_r^{(K)}\|^2 \\ &s.t.\ P_\Omega(\boldsymbol{\mathcal{Y}}) = P_\Omega(\boldsymbol{\mathcal{X}}) \end{aligned} \tag{15}$$

thus we have:

$$\boldsymbol{\mathcal{Y}} = P_\Omega(\boldsymbol{\mathcal{Y}}) + P_{\Omega^C}(\sum_{r=1}^{R} \lambda_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ ... \circ \mathbf{u}_r^{(K)}) \tag{16}$$

Experiments

In this session, we will implement the proposed model in metro passenger flow data and compare the proposed method with other state-of-art tensor completion methods.

Dataset: The dataset we use is passenger entry and exit data in Hong Kong metro stations. Each business day is defined as 5:00 AM to 1:00 AM of the next day, and we downsample it by grouping every 5 minutes, with 247 data points

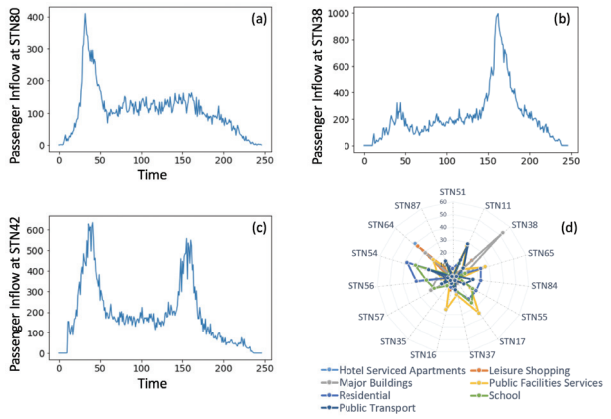

Figure 2: (a),(b),(c) are the passenger flow profile for station 80, 38, 42 at 06-Jan-2017 (Fri) respectively; (d) the POI information of selected stations

each day. For data split, we use data from 5:00AM, 01-Jan-2017 to 01:00AM, 19-Feb-2017 as observed data (training set), data from 05:00AM, 19-Feb-2017 to 11:10AM, 20-Feb-2017 as validation set and data from 11:10AM, 20-Feb-2017 to 1:00AM, 21-Feb-2017 as missing data (testing set). Besides, for data simplification (Sun and Axhausen 2016), we randomly choose 15 stations with diverse features. The 15 stations present quite different passenger flow pattern. So eventually the data we use is: $\boldsymbol{\mathcal{X}}^{L\times P\times T}$ with $L = 15$, $P = 247$ and $T = 51$.

The 15 stations presents quite different inflow passenger flow profile as shown in Figure 2: station 80, located in school area, has strong morning peak (probably for the commuting to school); station 38, locating in business area, has obvious afternoon peak (maybe because of work-off); station 42 in residential area have dual peak for to-and-from home.

Graph: As mentioned before, the contextual similarity graph and metro network graph play an remarkable role in the passenger flow pattern of each station. If two stations are quite similar in contextual graph or close in metro network graph, then they are more likely to share some passenger flow pattern. For contextual similarity, the Point Of Information (POI) is quite well applied (Zhong et al. 2017), and we denote it as Graph $\mathbf{G}_{POI}$. For each station, the information of its surrounding facilities are collected as vector (Hotel Serviced Apartments, Leisure Shopping, Major Buildings, Public Facilities Services, Residential, School, Public Transport), with each dimension value as the total amount of the corresponding facilities. In Figure 2.(d), we can easily observe that, some stations have dominant POI pattern: station 38 is dominant by major office building, thus its passenger flow has afternoon peak pattern.

After obtaining the POI vector of each station, then the POI similarity of station i and j can be calculated as cosine similarity:

$$\{\mathbf{G}_{POI}\}_{i,j} = \frac{\mathbf{POI}_i \cdot \mathbf{POI}_j}{\|\mathbf{POI}_i\| \cdot \|\mathbf{POI}_j\|} \tag{17}$$

Table 1: Best Parameter Searching

No.	β (1)	α (2)	γ (3)	δ (4)	MSE($\times 10^6$)
1	1000	50	70	80	3.34
2	1100	60	100	100	6.07
3	1200	70	20	20	7.98
4	1300	30	0	80	7.95
5	1400	80	50	80	**2.68**
6	1500	80	70	60	6.53
7	1600	30	10	0	4.21
8	1700	60	70	80	5.77
9	1800	80	80	60	6.94
10	1900	30	100	20	5.98
11	2000	70	30	40	5.95
12	1400	0	0	0	8.32
13	1400	80	0	0	4.94
14	1400	80	50	0	3.25
15	1400	80	50	80	2.68

(1) Low-Rank Penalty; (2) Weak Dependency Penalty; (3) $\mathbf{G}_{POI}$ Penalty; (4) $\mathbf{G}_{NET}$ Penalty

The metro network graph $\mathbf{G}_{NET}$ can be also defined in several ways: some define it as K-hop, which means how many hops are needed for travelling from station i to station j, which is $\{\mathbf{G}_{NET}\}_{i,j} = hop\_amount_{i,j}$; Or some define it as binary graph: when $hop\_amount_{i,j} \leqslant K$ is called 'connected' with label '1', otherwise with label '0', which is shown as following:

$$\{\mathbf{G}_{NET}\}_{i,j} = \begin{cases} 1 & hop\_amount_{i,j} <= K \\ 0 & hop\_amount_{i,j} > K \end{cases} \tag{18}$$

Method for Evaluation: We compare the proposed model with the following methods for our case.

- **Low-Rank Tensor Completion by Riemannian Optimization** (geomCG): which adds nuclear norm on the unfolded tensor along each mode, and performs Riemannian optimization techniques (Kressner, Steinlechner, and Vandereycken 2014)
- **High Accuracy Low Rank Tensor Completion** (HaLRTC): which similarly adds nuclear norm on the unfolded tensor along each mode, but solves it by Alternating Direction Method of Multipliers (ADMM) algorithm (Liu et al. 2013).
- **Fully Bayesian CP Factorization** (FBCP): which is based on CP decomposition and bayesian graphical model, and assumes that all the decomposed mode matrices are generated from a same Gaussian distribution, whose inverse covariance matrix is generated by another gamma distribution (Zhao, Zhang, and Cichocki 2015).
- **Tensor Rank Estimation based on L_1-regularized orthogonal CP decomposition** (TREL1_CP); which achieve the Low-Rank on CP decomposition by introducing L_1 penalty on weight vector (Shi, Lu, and Cheung 2017).
- **Low-Rank Tensor Decomposition with feature Variance Maximization via CP** (TDVM_CP):

Table 2: Comparison of Benchmark Methods

Method	MSE($\times 10^6$)	MAPE (%)
geomCG	74.2	2881.5
HaLRTC	8.41	84.12
FBCP	3.98	172.81
TREL1_CP	11.19	290.85
TDVM_CP	8.44	180.95
MTIOP#LRS	6.15	116.41
WDGTC (our method)	**2.68**	**63**

which is based on TREL1-CP and maximizes the feature variance (Shi et al. 2018).

- **Multi-Task for Inflow and Outflow Prediction combining the Lasso regularization term, Ridge regularization term, and Laplacian regularization term** (MTIOP#LRS): which also introduces POI, traffic capacity, connectivity and weather information as factors into the model, whereas the traffic flow as response (Zhong et al. 2017).

Performance Comparison

For all methods, we tune the model parameters by grid search according to the performance on the validation dataset and report the performance on the testing dataset.Ther performance metric we consider are Mean Square Error (MSE), Relative Residual (RES) and Mean Absolute Percentage Error (MAPE). Table 2 illustrates the performance comparison of all the mentioned methods. Take the method we proposed in this case study for example, since we introduced two graphs, four parameter are required to be tuned, which is shown as: $\min_{R,\boldsymbol{\lambda},\{\mathbf{u}_r^{(k)}\}} loss + \alpha\|\mathbf{U}_1\|_1 + \beta\|\boldsymbol{\lambda}\|_1 + \frac{1}{2}\gamma \cdot tr(\mathbf{U}_1^T\mathbf{L}_{POI}\mathbf{U}_1) + \frac{1}{2}\delta \cdot tr(\mathbf{U}_1^T\mathbf{L}_{NET}\mathbf{U}_1)$, where $\alpha, \beta, \gamma, \delta$ are searched within the grid [0, 100], [1000, 2000], [0, 100] and [0, 100] respectively, with searching step as 10, 100, 10 and 10 respectively. These grids are selected by trial-and-error until we find the satisfactory grid. And the steps are chosen out of the consideration of lowering computational cost and the insensibility of results to the tuning parameters.

According to Table 1, we can make the decision that the parameter combination No. 5 can yield the best completion result; And besides, for each low-rank penalty β, the best weak dependency penalty α, POI graph penalty γ, and NET graph penalty δ are almost always nonzero, which means introducing those penalty can truly improve the completion performance. Compared with No.5 with No. 12 to 15, we found that the main contribution to the improvement is introducing the weakly-correlated penalty, achieving almost 60% MSE decrease, the POI graph contributing around 30% MSE decrease, whereas NET graph contributing 10%.

According to Table 2, We can observe that : (1) geomCG cannot almost yield prediction at all, and the possible reason is that: in our experiment setting all the missing data aggregate in a quite small portion of the tensor, that is: only the data from 11:10AM at last day is missing with missing rate as 1.31% only, so geomCG is prone to converge on the observation set immediately before starting to estimate missing part; (2) According to Table 1, for the overall MSE, our method can achieve almost 30% error reduction comparing with the second best baseline method FBCP; (3) Although method MTIOP#LRS introduces auxiliary information as well, passenger flow does not directly have a statistically functional relationship with all those factors.

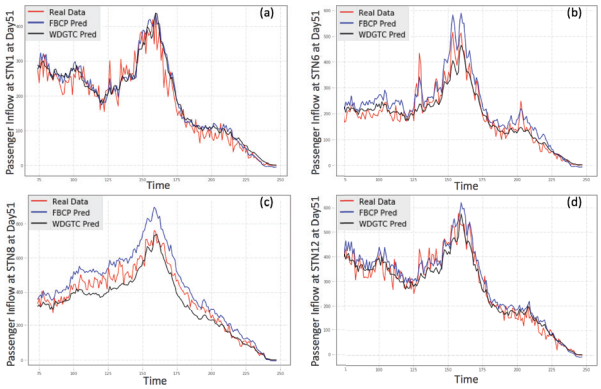

Figure 3: (a), (b), (c) and (d) Completion Result for station 1,6,8 and 12 respectively

According to Table 3, we can find that: (1) our method can yield the most accurate completion in most of the selected stations (with the best performance in boldface); (2) in some stations, for example, station 5, station 8 and station 11, since its passenger flow pattern is quite unique, so most of other method perform badly (with the bad performance underlined). And the reason is: those method don't complete the missing data obeying the weakly-dependent pattern and graph structure. The completed passenger flow profile by our method, and the second best method (FBCP) are plotted in Figure 3.

Conclusion and Future Work

In this paper, we studied a tensor completion problem for a weakly-dependent tensor on graphs. Based on the metro passenger flow tensor, we defined the weakly-dependent pattern and the graph structure behind this. Based on the problem definition, we proposed a novel tensor completion method regularized tensor decomposition by introducing weakly dependent penalty and graph penalty. In addition, we offered a solution for this optimization through Block Coordinate Descent. In the case study based on metro passenger flow data, we defined two graphs which are decisive for each station's pattern. Here, we prove the overall accuracy of the proposed method. For future research, our aim is to formulate the Tucker Decomposition based completion version for the weakly dependent on graph structure.

Acknowledgments

This research is supported by Hong Kong MTR Co. with grant numbers RGC GRF 16203917 and 16201718. We also

Table 3: Performance of All Methods on Each Station

(RES)	geomCG	HaLRTC	FBCP	TREL1_CP	TDVM_CP	MTIOP#LRS	WDGTC
STN0	21.57	0.09	0.11	0.13	0.13	0.11	**0.06**
STN1	7.61	0.14	0.19	0.09	0.09	0.09	**0.08**
STN2	9.12	2.71	0.19	3.65	3.24	2.88	1.13
STN3	16.93	0.31	0.33	0.34	0.32	0.31	**0.19**
STN4	22.10	0.13	0.26	0.12	0.1	0.1	**0.09**
STN5	6.50	1.52	0.76	1.25	0.68	1.12	**0.47**
STN6	8.59	0.31	0.33	0.39	0.28	0.26	**0.19**
STN7	7.60	0.11	0.12	0.07	**0.06**	0.07	0.1
STN8	6.78	0.87	0.57	1.17	0.56	0.54	**0.27**
STN9	20.55	0.60	0.23	0.67	0.64	0.47	**0.22**
STN10	10.52	1.68	0.58	2.46	2.22	1.22	1.03
STN11	10.29	2.04	0.22	3.14	2.83	2.07	**0.18**
STN12	4.31	0.26	0.19	0.13	0.11	0.17	**0.1**
STN13	7.07	3.27	**0.72**	5.32	3.45	2.31	1.79
STN14	4.07	0.14	0.38	0.14	0.18	0.14	**0.11**

give special thanks to Dr. Qiquan Shi, and Dr. Qibin Zhao for sharing the scripts of their methods.

References

Chen, X.; He, Z.; and Sun, L. 2019. A bayesian tensor decomposition approach for spatiotemporal traffic data imputation. *Transportation research part C: emerging technologies* 98:73–84.

Chen, Y.-L.; Hsu, C.-T.; and Liao, H.-Y. M. 2013. Simultaneous tensor decomposition and completion using factor priors. *IEEE transactions on pattern analysis and machine intelligence* 36(3):577–591.

Dunlavy, D. M.; Kolda, T. G.; and Acar, E. 2011. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 5(2):10.

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

Kressner, D.; Steinlechner, M.; and Vandereycken, B. 2014. Low-rank tensor completion by riemannian optimization. *BIT Numerical Mathematics* 54(2):447–468.

Lee, H.; Battle, A.; Raina, R.; and Ng, A. Y. 2007. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, 801–808.

Liu, J.; Musialski, P.; Wonka, P.; and Ye, J. 2012. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence* 35(1):208–220.

Mørup, M.; Hansen, L. K.; and Arnfred, S. M. 2008. Algorithms for sparse nonnegative tucker decompositions. *Neural computation* 20(8):2112–2131.

Shi, Q.; Cheung, Y.-M.; Zhao, Q.; and Lu, H. 2018. Feature extraction for incomplete data via low-rank tensor decomposition with feature regularization. *IEEE transactions on neural networks and learning systems* 30(6):1803–1817.

Shi, Q.; Lu, H.; and Cheung, Y.-m. 2017. Tensor rank estimation and completion via cp-based nuclear norm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 949–958. ACM.

Sun, L., and Axhausen, K. W. 2016. Understanding urban mobility patterns with a probabilistic tensor factorization framework. *Transportation Research Part B: Methodological* 91:511–524.

Wang, S.; He, L.; Stenneth, L.; Yu, P. S.; and Li, Z. 2015. Citywide traffic congestion estimation with social media. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 34. ACM.

Yokota, T.; Zhao, Q.; and Cichocki, A. 2016. Smooth parafac decomposition for tensor completion. *IEEE Transactions on Signal Processing* 64(20):5423–5436.

Yu, K.; He, L.; Philip, S. Y.; Zhang, W.; and Liu, Y. 2019. Coupled tensor decomposition for user clustering in mobile internet traffic interaction pattern. *IEEE Access* 7:18113–18124.

Zhang, C.; Yan, H.; Lee, S.; and Shi, J. 2018. Weakly correlated profile monitoring based on sparse multi-channel functional principal component analysis. *IISE Transactions* 50(10):878–891.

Zhao, Q.; Zhang, L.; and Cichocki, A. 2015. Bayesian cp factorization of incomplete tensors with automatic rank determination. *IEEE transactions on pattern analysis and machine intelligence* 37(9):1751–1763.

Zhong, R.; Lv, W.; Du, B.; Lei, S.; and Huang, R. 2017. Spatiotemporal multi-task learning for citywide passenger flow prediction. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 1–8. IEEE.