Structurally Regularized Non-negative Tensor Factorization for Spatio-Temporal Pattern Discoveries

Koh Takeuchi[1,3(✉)], Yoshinobu Kawahara[2,4], and Tomoharu Iwata[1]

[1] NTT Communication Science Laboratories, Kyoto, Japan
{takeuchi.koh,iwata.tomoharu}@lab.ntt.co.jp
[2] The Institute of Scientific and Industrial Research (ISIR), Osaka University, Osaka, Japan
ykawahara@sanken.osaka-u.ac.jp
[3] Department of Intelligence Science and Technology, Kyoto University, Kyoto, Japan
[4] Center for Advanced Intelligence Project, RIKEN, Tokyo, Japan

Abstract. Understanding spatio-temporal activities in a city is a typical problem of spatio-temporal data analysis. For this analysis, tensor factorization methods have been widely applied for extracting a few essential patterns into latent factors. Non-negative Tensor Factorization (NTF) is popular because of its capability of learning interpretable factors from non-negative data, simple computation procedures, and dealing with missing observation. However, since existing NTF methods are not fully aware of spatial and temporal dependencies, they often fall short of learning latent factors where a large portion of missing observation exist in data. In this paper, we present a novel NTF method for extracting smooth and flat latent factors by leveraging various kinds of spatial and temporal structures. Our method incorporates a unified structured regularizer into NTF that can represent various kinds of auxiliary information, such as an order of timestamps, a daily and weekly periodicity, distances between sensor locations, and areas of locations. For the estimation of the factors for our model, we present a simple and efficient optimization procedure based on the alternating direction method of multipliers. In missing value interpolation experiments of traffic flow data and bike-sharing system data, we demonstrate that our proposed method improved interpolation performances from existing NTF, especially when a large portion of missing values exists.

1 Introduction

Spatio-temporal data covering a wide area of a city have become available due to the commoditization of sensor-monitoring systems and mobile-phone networks. These monitoring systems observe various types of data, such as vehicle transportation counts on a road network, bike-renting counts of a bike-sharing system, and the purchasing records of shops around a city, where missing values often

appear due to the failure of sensor nodes, data transmission errors, and trouble with data recording systems. We can find rich and bounteous information in such spatio-temporal data. However, it becomes difficult to grasp what spatio-temporal activities appeared in the data at a glance. Therefore, understanding of such activities via pattern extractions is a typical problem of spatio-temporal data analysis, in which the interpretability of the extracted patterns is regarded as one of the most important property for analysis methods.

Tensor factorization methods have been widely applied to discover spatial and temporal patterns from various kinds of spatio-temporal data [17]. These methods represent spatio-temporal data as a higher-order dimensional array, called a tensor that is a generalization of a matrix. For example, we can represent spatio-temporal data as a three-way tensor whose first, second, and third modes correspond to sensor locations, timestamps for 24 h, and the observed days. We illustrated an example of a tensor for spatio-temporal data analysis in Fig. 1. With this formulation, we can naturally incorporate an assumption that daily or weekly periodicity can be found in data and similar spatial patterns appear on different days. We can extract a few numbers of spatial, temporal, and daily patterns into latent factors by decomposing the tensor. However, since most existing tensor factorization methods do not consider the non-negativity of data where observations only contain non-negative values, they often result in messy and hard to interpret factors.
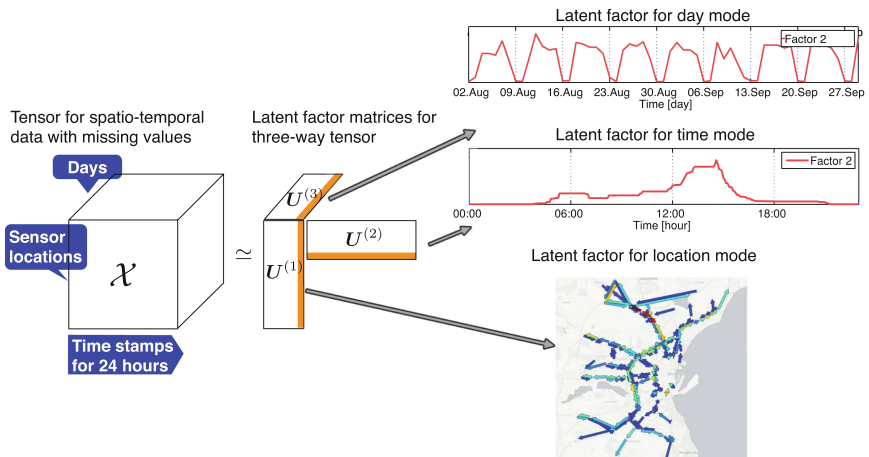

Fig. 1. Example for a non-negative tensor factorization method on analysis of a traffic-flow data set, where latent patterns for location, time, and day modes are extracted in the latent factor.

Unlike those tensor factorization methods, Non-negative Tensor Factorization (NTF) [8], which leverages non-negativity, is effective for extracting interpretable patterns from the non-negative data [13,18]. This method has successively yielded interpretable factors from various kinds of spatio-temporal

data, such as location-based social network services [14,25], mobile phone GPS logs [10], log messages of network equipment [16], and traffic records of road networks [32]. However, NTF was not applicable to the existence of missing values. To deal with missing values, NTF was recently extended to learn the latent factors from a subset of elements in a tensor, called the non-negative tensor completion [15,31]. With this NTF, we can interpolate missing values in data by learned latent factors. However, NTF methods for the missing value completion problem suffer from overfitting when just a few observations are available. Because they ignore spatial and temporal contextual information such as the order of time stamps, weekly periodicity, the distances between sensor locations and treats each feature of the tensor independently.

To incorporate such contextual information, most matrix/tensor factorization methods have employed a graph Laplacian based regularizer for encouraging the latent factors to be smooth with spatio-temporal dependencies [21]. The graph regularized non-negative matrix factorization [6] is a variant of such schemes and has been widely utilized in many applications, however, it does not consider scenarios where missing values exist and analyzing higher-order dimensional arrays.

Another choice for representing such auxiliary information is structured regularizers [2] that have become popular in the fields of machine learning, signal processing, and data mining [7,28]. For example, the fused lasso [27], which is also known as the total variation, approximates parameters by piecewise-constant values with the order of parameters. Since its estimated parameters have the same estimated value, this is beneficial for finding segments of parameters. In a pioneering work [29], the penalized matrix decomposition was proposed to utilize the fused lasso as a regularizer on latent factors and was applied to a gene data analysis problem. They presented latent factors easy to find gene segments rather than existing matrix factorization methods incorporated the lasso regularizer. However, this method and its subsequent works have only considered the fused lasso without incorporating more general structured regularizer such as spatial dependencies of sensors, and also ignored the non-negative properties and the existence of missing values.

In this paper, we attempt to solve a problem of extracting latent factors from spatio-temporal data where a lot of missing values exists. To tackle this problem, we propose a novel NTF that learns factors by employing spatial and temporal auxiliary information as regularizers. We utilize this information to represent phenomena often appear in spatio-temporal data, such as counts of vehicles passed roads smoothly grow or decrease or take the same value along with space and time. To exploit such information, we introduce a regularizer that consists of both a graph-based Laplacian regularizer and structured regularizers that incorporate not only the order of features but also more general graph and group based structures [3,24]. With our regularizer, we can utilize various kinds of auxiliary information into NTF including a daily and weekly periodicity, distances between sensor locations, and areas of locations. Our proposed method is highly robust to the presence of a large portion of missing values because it encourages latent factors to be smooth and flat with spatial

and temporal structures, where we regard segments of parameters that take the same value as flat. To estimate the latent factors for our proposed method, we present an efficient optimization procedure of the alternating direction method of multipliers [4] that utilizes simple proximity operators of the conjugate gradient method [21] and a parametric network flow algorithm [12].

We conducted missing value interpolation experiments with real-world traffic flow data and compared the performance of our proposed method with existing NTF methods. We demonstrate that our proposed method improved the interpolation performances from existing NTF methods. We also show that our extracted factors were interpretable to detect change points. Because our factors have segments, we can easily find a boundary of segments as a change point.

2   Non-negative Tensor Factorization

We denote a N-th way non-negative tensor as $\mathcal{X} \in \mathbb{R}_{\geq 0}^{I_1 \times \cdots \times I_N}$, where I_n is the number of features in the n-th mode. The n-th mode unfolding of a tensor $\mathcal{X}$ is denoted as $\mathcal{X}_n$. We use $i = (i_1, \ldots, i_N)$ and D to represent an element and the whole set of the elements in the tensor, respectively. A subset of the observed elements in the tensor is denoted by $\Omega = \{i \mid x_i \text{ is observed }, \forall i \in D\}$.

NTF decomposes the observed values of tensor $\mathcal{X}$ into K latent non-negative factors, where $K \ll \min(I_1, \ldots, I_N)$. The n-th mode factor matrix is denoted as $\boldsymbol{A}^{(n)} \in \mathbb{R}_{\geq 0}^{I_n \times K}$ whose k-th column is factor vector $\boldsymbol{a}_k^{(n)} \in \mathbb{R}_{\geq 0}^{I_n}$. We denote a whole set of factor vectors as $A = \{\boldsymbol{a}_k^{(n)} \mid \forall(n, k)\}$. An estimation for element x_i is given by a sum of latent factor vectors $\hat{x}_i = \sum_{k=1}^{K} a_{i_1,k}^{(1)} a_{i_2,k}^{(2)} \cdots a_{i_N,k}^{(n)} \in \hat{\mathcal{X}}$. We denote the transpose operator as $^\top$, the Khatri-Rao product as $\odot$, and its series as $\odot_{n=1}^{N} \boldsymbol{A}^{(n)} = \boldsymbol{A}^{(1)} \odot \cdots \odot \boldsymbol{A}^{(N)}$.

The empirical loss function for NTF can be defined as a sum of divergences that indicates a discrepancy between x_i and its estimation $\hat{x}_i$:

$$f(A) = D_\Omega(\mathcal{X}\|\hat{\mathcal{X}}) + \sum_{n=1}^{N} \sum_{k=1}^{K} g^{(n)}(\boldsymbol{a}_k^{(n)}), \tag{1}$$

where $D_\Omega(\mathcal{X}\|\hat{\mathcal{X}}) = \sum_{i \in \Omega} d(x_i\|\hat{x}_i)$. We $d(p\|q)$ to denote a divergence between scalars p and q, and $g^{(n)}$ to denote a penalty function for the n-th mode factor vector. Because loss function f is non-convex with respect to A, an NTF problem is to obtain a local minimizer A^* of the loss under a non-negative constraint:

$$A^* = \arg\min_A f(A) \text{ subject to } \boldsymbol{a}_k^{(n)} \geq 0 \ \ \forall(n, k). \tag{2}$$

The graph regularized non-negative matrix factorization method [6] employs a graph Laplacian regularizer [22] to represent the smoothness in latent factors. An adjacency matrix for the n-th mode features is denoted as $\boldsymbol{W}^{(n)} \in \mathbb{R}^{I_n \times I_n}$ that represents a graph whose nodes and capacities of edges correspond to the features of the n-th mode and the similarity measures between the two features,

respectively. The Laplacian matrix can be denoted as $\boldsymbol{L}^{(n)} = \boldsymbol{D}^{(n)} - \boldsymbol{W}^{(n)}$, where $\boldsymbol{D}^{(n)}$ is a diagonal matrix whose elements are the sums of each row of $\boldsymbol{W}^{(n)}$. Then a graph Laplacian regularizer can be defined:

$$g^{(n)}(\boldsymbol{a}_k^{(n)}) = \boldsymbol{a}_k^{(n)^\top} \boldsymbol{L}^{(n)} \boldsymbol{a}_k^{(n)}. \tag{3}$$

This regularizer penalty function encourages smoothness because its formulation equals putting a weighted quadratic term on the difference between the adjacency elements.

3   Proposed Model

We introduce a unified structured regularizer to employ both smooth and piecewise-constant properties with auxiliary structures:

$$g^{(n)}(\boldsymbol{a}_k^{(n)}) = \sum_{m=1}^{3} \lambda_m g_m^{(n)}(\boldsymbol{a}_k^{(n)}) + g_{\geq 0}^{(n)}(\boldsymbol{a}_k^{(n)}), \tag{4}$$

where λ_1, λ_2 and λ_3 are the hyperparameters for each regularizer. We employ a Generalized Fused Lasso (GFL) [5,30] and a Higher-Order Fused Lasso (HOFL) [24] as $g_1^{(n)}$ and $g_2^{(n)}$, respectively. $g_3^{(n)}$ corresponds to the Laplacian regularizer for extracting smooth patterns. We use an indicator function for the non-negative region:

$$g_{\geq 0}^{(n)}(\boldsymbol{a}_k^{(n)}) = \begin{cases} 0 & (\text{if } a_{i,k} \geq 0, \ \forall i) \\ +\infty & (\text{otherwise}) \end{cases}. \tag{5}$$

The GFL penalty is defined:

$$g_1(\boldsymbol{a}_k^{(n)}) = \sum_{j=1}^{I_n} \sum_{j'=1}^{I_n} w_{j,j'}^{(n)} \left| a_{j,k}^{(n)} - a_{j',k}^{(n)} \right|. \tag{6}$$

The GFL prefers parameters with the same value if they are adjacent on the given graph, such as distances between sensor locations and temporal lags between time stamps. The HOFL encourages parameters in a given group to take identical values [24]. With this regularizer, we can utilize auxiliary information, such as sensors placed in a specific area that may output similar values and a group of time stamps when a specific train leaves from a station. We denote the r-th group of features in the n-th mode as $g_r^{(n)} \subseteq D_n$ and a set of groups by $\mathcal{G}^{(n)} = \{g_1^{(n)}, \cdots, g_{R_n}^{(n)}\}$, where D_n and R_n are a set of elements in the n-th mode and the number of groups, respectively. The weights of each element for the r-th group on the n-th mode are denoted by $c_{r,m}^{(n)} = \bar{c}_{r,m}^{(n)}$ if $m \in g_r^{(n)}$, and 0 otherwise, where $\bar{c}_{r,m}^{(n)} > 0$. Then a simplified HOFL penalty $g_2(\boldsymbol{a}_k^{(n)})$ is given:

$$\sum_{r=1}^{R} \sum_{m=1}^{I_n} c_{r,j_m}^{(n)} |a_{j_m,k}^{(n)} - \bar{a}_{r,j_m,k}^{(n)}| + \theta_r^{(n)}(a_{s_r,k}^{(n)} - a_{t_r,k}^{(n)}), \tag{7}$$

where $\theta_r^{(n)} > 0$ is a hyperparameter that controls the consistency of the parameters in a group. $\bar{a}_{r,k}^{(n)}$ is defined as $\bar{a}_{r,m,k}^{(n)} = a_{s_k,k}^{(n)}$ (if $m \geq s_k$), $a_{t_k,k}^{(n)}$ (if $m \leq t_k$) and $a_{j_m,k}^{(n)}$ (otherwise) for distinct indices $j_1, j_2, \ldots, j_{I_n} \in D_n$ that correspond to a permutation that arranges the entries of $\boldsymbol{a}_k^{(n)}$ in a non-increasing order. Thresholding indices s_r and t_r are given as $s_k = \min\left\{m' \mid \sum_{m=1}^{m'} c_{r,j_m}^{(n)} \geq \theta_r^{(n)}\right\}$ and $t_k = \min\left\{m' \mid \sum_{m=m'}^{I_n} c_{r,j_m}^{(n)} < \theta_r^{(n)}\right\}$.

For convenience, we denote $\bar{g}^{(n)}(\boldsymbol{a}_k^{(n)}) = \sum_{m=1}^{2} \lambda_m g_m^{(n)}(\boldsymbol{a}_k^{(n)}) + g_{\geq 0}^{(n)}(\boldsymbol{a}_k^{(n)})$. By adopting our structured regularizers to the loss of NTF, we define the following minimization problem for our purpose:

$$A^* = \arg\min_A D_\Omega(\mathcal{X}\|\hat{\mathcal{X}}) + \sum_{n=1}^{N}\sum_{k=1}^{K} \bar{g}^{(n)}(\boldsymbol{a}_k^{(n)}) + \lambda_3 g_3^{(n)}(\boldsymbol{a}_k^{(n)}). \qquad (8)$$

Note that when $\lambda_1 = \lambda_2 = \lambda_3 = 0$, our method is reduced to an original NTF. When $\lambda_2 = \lambda_3 = 0$, our method can be regarded as a tensor extension of the graph-regularized non-negative matrix factorization. Our method includes those methods as special cases.

4 Parameter Estimation

We present an efficient parameter estimation procedure for obtaining a local minimizer of our proposed method. We employ a scaled formulation of the Alternating Direction Method of Multipliers (ADMM) for NTF [15]. The minimization problem for our proposed method can be rewritten:

$$\min_{A,\mathcal{Z}} D_\Omega(\mathcal{X}\|\mathcal{Z}) + \sum_{n=1}^{N}\sum_{k=1}^{K} \bar{g}^{(n)}(\boldsymbol{b}_k^{(n)}) + g_3^{(n)}(\boldsymbol{a}_k^{(n)})$$
$$\text{subject to } \mathcal{Z} = \hat{\mathcal{X}}, \boldsymbol{a}_k^{(n)} = \boldsymbol{b}_k^{(n)} \ (\forall n, k), \qquad (9)$$

where $\mathcal{Z}$ and $\boldsymbol{b}_k^{(n)}$ are auxiliary variables, and P_Ω is a projection function that only retains the divergence of the observed elements. To solve our problem efficiently with keeping both constraints and separability, we define an augmented Lagrangian for our problem:

$$L_\rho(A, B, \mathcal{Z}) = D_\Omega(\mathcal{X}\|\mathcal{Z}) + \frac{\rho}{2}\|\mathcal{Z} - \hat{\mathcal{X}} + \mathcal{U}\|_\mathcal{F}^2$$
$$\sum_{n=1}^{N}\sum_{k=1}^{K} \bar{g}^{(n)}(\boldsymbol{b}_k^{(n)}) + g_3^{(n)}(\boldsymbol{a}_k^{(n)}) + \frac{\rho}{2}\left\|\boldsymbol{a}_k^{(n)} - \boldsymbol{b}_k^{(n)} + \boldsymbol{u}_k^{(n)}\right\|_2^2, \qquad (10)$$

where $\mathcal{U}$ and $\boldsymbol{u}_k^{(n)}$ are Lagrangian multipliers, and ρ is a step-size parameter, respectively. We summarize the minimization procedure for our proposed method in Algorithm 1. The minimization for ADMM can be efficiently calculated if a simple minimization operator for each of each $\boldsymbol{a}_k^{(n)}$ and $\boldsymbol{b}_k^{(n)}$ exists.

Algorithm 1. Alternative direction method of multiplier for our proposed non-negative tensor factorization

---

    Input   : $\mathcal{X}, \Omega, \lambda_1, \lambda_2, \lambda_3, K, \boldsymbol{W}^{(n)}$
    Output: set of factor matrices A
1 Initialize parameters
2 Sample A, B, and $\mathcal{Z}$ from random distributions
3 repeat
4   Alternatively update parameters;
5 $\mathcal{Z} \leftarrow \arg\min_{\mathcal{Z}} D_{\Omega}(\mathcal{X}\|\mathcal{Z}) + (\rho/2)\|\mathcal{Z} - \hat{\mathcal{X}} + \mathcal{U}\|_F^2$
6   **for** $n = 1$ **to** N **do**
7   Update $\boldsymbol{A}^{(n)}$ by solving Eq. (11)
8   **for** $k = 1$ **to** K **do**
9   Update $\boldsymbol{b}_k^{(n)}$ by solving Eq. (12)
10   **end**
11 $\boldsymbol{U}^{(n)} \leftarrow \boldsymbol{U}^{(n)} + (\boldsymbol{A}^{(n)} - \boldsymbol{B}^{(n)})$
12   **end**
13 $\mathcal{U} \leftarrow \mathcal{U} + (\mathcal{X} - \hat{\mathcal{X}})$
14 until *convergence*;

The loss function with respect to $\boldsymbol{A}^{(n)}$ and $\boldsymbol{b}_k^{(n)}$ contains the graph Laplacian regularizer and the non-separable graph-based and group-based penalties, respectively. Thus the main difficulty with our proposed method lies in the minimization of $\boldsymbol{A}^{(n)}$ and $\boldsymbol{b}_k^{(n)}$, whose minimization problems can be rewritten:

$$\boldsymbol{A}^{(n)} = \underset{\boldsymbol{A}^{(n)}}{\arg\min} \frac{\rho}{2}\|\bar{\mathcal{Z}}_n - \boldsymbol{A}^{(n)}\boldsymbol{V}_n^{\top}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{A}^{(n)} - \bar{\boldsymbol{V}}_n\|_2^2 + \lambda_3 \sum_{k=1}^{K} g_3^{(n)}(\boldsymbol{a}_k^{(n)}) \quad (11)$$

$$\boldsymbol{b}_k^{(n)} = \underset{\boldsymbol{b}_k^{(n)}}{\arg\min} \bar{g}^{(n)}(\boldsymbol{b}_k^{(n)}) + \frac{\rho}{2}\|\bar{\boldsymbol{v}}_k^{(n)} - \boldsymbol{b}_k^{(n)}\|_2^2, \quad (12)$$

where $\bar{\mathcal{Z}} = \mathcal{Z} + \mathcal{U}$, $\boldsymbol{V}_n = \odot_{n=n'}^{N} \boldsymbol{A}^{(n)}$, $\bar{\boldsymbol{V}}_n = \boldsymbol{B}^{(n)} - \boldsymbol{U}^{(n)}$, and $\bar{\boldsymbol{v}}_k^{(n)} = \boldsymbol{a}_k^{(n)} + \boldsymbol{u}_k^{(n)}$. We efficiently solve the minimization of Eq. (11) by using the fact that it corresponds to the loss function of the graph regularized alternating least squares [21], which approximately runs in $\mathcal{O}(\text{nnz}(\boldsymbol{L}^{(n)})K)$ ($\text{nnz}(\cdot)$ is the number of non-zero elements).

The minimization problem in Eq. (12) corresponds to the calculation of the proximity operator, which is defined as: $\text{prox}_{\gamma h}(\theta) = \arg\min_{\theta} h(\theta) + \frac{1}{2\gamma}\|\hat{\theta} - \theta\|_2^2$. We present a minimization procedure for Eq. (12) by leveraging the properties of the proximity operator, and obtaining a minimizer for the sum of the non-negative indication function and other convex functions by the following property [26]: $\text{prox}_{\bar{g}^{(n)}} = \text{prox}_{g_{\geq 0}^{(n)}} \circ \text{prox}_{\lambda_1 g_1^{(n)} + \lambda_2 g_2^{(n)}}$. Thus, if we have a minimizer for $\lambda_1 g_1^{(n)} + \lambda_2 g_2^{(n)}$, we can attain the exact minimizer for $\bar{g}^{(n)}$ by setting negative parameters to zeros. A minimizer for $\lambda_1 g_1^{(n)} + \lambda_2 g_2^{(n)}$ can be simply

calculated by employing a submodular function minimization procedure. Because the penalty functions of GFL and HOFL are the Lovász extensions [19] of the graph-representable submodular functions [11], we can attain a minimizer for the sum of functions $\lambda_1 g_1^{(n)} + \lambda_2 g_2^{(n)}$ by an efficient parametric network flow algorithm [7,24,30]. We show the details of our minimization procedure for this function in the appendix.

5   Related Works

There has been a lot of articles in which NTF was applied to analyze spatio-temporal data. Kimura et al. proposed a special NTF that decomposes a three-way tensor into two-factor matrices and a three-mode tensor for extracting log messages related to network failures [16]. Yang et al. proposed a combination of NTF without regularizers and post-processing for modeling user activities [32]. Takeuchi et al. proposed an NTF that simultaneously decomposes multiple tensors to extract patterns appeared among different tensors [25]. NTF was used to extract spatio-temporal patterns from human-flow data [10]. However, all of those methods did not employ regularizers into NTF and their methods were not applicable to missing values. One exception is a paper of Sun and Axhausen [23], in which they proposed a probabilistic non-negative Tucker decomposition for discovering interactions among factors. However, they did not incorporate the spatial and temporal structures into regularizers. Han and Moutarde proposed an extension of NTF for predicting future observations [14]. However, they did not consider spatial structures. Our method can be applied to their framework to utilizing spatial and temporal regularizers. The estimation procedures of them were based on the multiplicative update rule and EM algorithm. Our proposed method can utilize graphs and groups of spatial and temporal features to regularize parameters and also employ ADMM as an estimation procedure.

6   Experiments

We conducted missing value completion problems with a traffic flow data set provided by City Pulse [1] and two bike-sharing system data sets recorded in Washington D.C.[1] and New York[2] [1].

The traffic flow data consist of the numbers of cars that passed at 419 locations every thirty minutes in Arhus City, Denmark. We picked 30 days from August 2nd to 31st 2014, and constructed three-way tensor $\mathcal{X} \in \mathbb{R}^{48 \times 30 \times 441}$ whose modes corresponded to 48 daily time points, 30 days, and 441 observation locations, respectively. From the bike-sharing system data in Washington D.C. and New York, we employed 15 days from April 1st to the 15th with 351 and 344 bike stations. We constructed three-way tensors $\mathcal{X} \in \mathbb{R}^{24 \times 15 \times 351}$ and $\mathcal{X} \in \mathbb{R}^{24 \times 15 \times 44}$ whose values were the numbers of bikes returned to the station

---

[1] https://www.capitalbikeshare.com.
[2] http://www.citibikenyc.com/.

in an hour. For the time mode, we utilized the adjacency of the time points as a graph. For the day mode, we employed the adjacency of days and the days of the week as a graph and groups. For the location mode, we used the inverse of the Euclid distance of GPS locations and clusters attained by k-nearest neighbors ($k = 5, 10$) for a graph and groups.

We exploited the Euclid distance as the divergence in experiments. We compared our proposed method (Proposed 1) and our proposed method with only the graph Laplacian regularizer (Proposed 2, $\lambda_1 = \lambda_2 = 0$) with NTF estimated by ADMM [15] (ADMM), NTF with the graph Laplacian regularizer [6] estimated by a multiplicative update rule considering missing values (Multi+Lap) [9], and NTF estimated by the multiplicative update rule (Multi). We set the proportion of observations to $p = \{0.1, 0.01, 0.005, 0.001\}$. By five-fold cross validation, we selected K and other hyperparameters from $K = \{3, 5, 10\}$ and $\{0.1, 1, 10\}$. We utilized the normalized RMSE (NRMSE) and the normalized deviation (ND) as error measurements:

$$\text{NRMSE} = \sqrt{(1/|\Omega|) \sum_{(p,t) \in \Omega} (x_{p,t} - \hat{x}_{p,t})^2}/Q, \tag{13}$$

$$\text{ND} = (1/|\Omega|) \sum_{(p,t) \in \Omega} |x_{p,t} - \hat{x}_{p,t}|/Q, \tag{14}$$

here $Q = (1/|\Omega|) \sum_{(p,t) \in \Omega} |x_{p,t}|$. We ran our experiments five times with randomly selected different missing values.

The results are shown in Tables 1, 2, 3, 4, 5, and 6, where the left and right values in a cell correspond to the average and the standard deviation of those values. We confirmed that our proposed methods showed the best performance in every setting. Our proposed method was robust to the appearance of a large portion of missing values for every data set $p = \{0.01, 0.005, 0.001\}$. Our proposed method with both the graph-based Laplacian and structured regularizer (Proposed 1) showed better or competitive performance with our proposed method with the graph-based Laplacian regularizer (Proposed 2). Furthermore, our proposed method with the graph-based Laplacian regularizer (Proposed 2)

Table 1. NRAME for the traffic flow data of our proposed method (Proposed 1), our proposed method with the graph Laplacian regularizer (Proposed 2), NTF estimated by ADMM (ADMM), NTF with the graph Laplacian regularizer estimated by the multiplicative update rule (Multi+Lap), and NTF (Multi)

	$p = 0.1$	$p = 0.01$	$p = 0.005$	$p = 0.001$
Proposed 1	**0.50 (0.00)**	**0.99 (0.03)**	**1.49 (0.03)**	**1.87 (0.01)**
Proposed 2	0.51 (0.00)	1.12 (0.05)	**1.49 (0.03)**	1.89 (0.01)
ADMM	0.51 (0.00)	1.15 (0.03)	1.50 (0.02)	1.91 (0.00)
Multi+Lap	0.52 (0.00)	2.98 (1.86)	2.92 (1.00)	11.9 (11.7)
Multi	0.52 (0.00)	2.89 (2.22)	3.27 (1.80)	5.98 (6.24)

Table 2. NRAME for the bike-sharing record data of Washington D.C.

Method	$p = 0.1$	$p = 0.01$	$p = 0.005$	$p = 0.001$
Proposed 1	**1.67 (0.02)**	**2.14 (0.02)**	**2.21 (0.04)**	**2.43 (0.02)**
Proposed 2	1.68 (0.01)	**2.14 (0.05)**	2.22 (0.05)	2.62 (0.08)
ADMM	1.68 (0.02)	2.21 (0.05)	2.32 (0.03)	2.47 (0.01)
Multi+Lap	1.69 (0.01)	2.72 (0.22)	2.76 (0.24)	11.2 (4.62)
Multi	1.70 (0.01)	299.1 (405.7)	8.25 (4.87)	16.3 (3.13)

Table 3. NRAME for the bike-sharing record data of New York

Method	$p = 0.1$	$p = 0.01$	$p = 0.005$	$p = 0.001$
Proposed 1	**0.98 (0.00)**	**1.28 (0.02)**	**1.42 (0.01)**	**1.62 (0.01)**
Proposed 2	**0.98 (0.00)**	1.30 (0.03)	1.44 (0.01)	1.63 (0.01)
ADMM	**0.98 (0.00)**	1.34 (0.02)	1.49 (0.01)	1.65 (0.01)
Multi+Lap	1.00 (0.02)	27.2 (41.3)	5.68 (3.22)	1.86 (0.17)
Multi	1.00 (0.02)	53.7 (22.0)	26.6 (29.4)	3.04 (1.03)

Table 4. ND for the traffic flow data of our proposed method (Proposed 1), our proposed method with the graph Laplacian regularizer (Proposed 2), NTF estimated by ADMM (ADMM), NTF with the graph Laplacian regularizer estimated by the multiplicative update rule (Multi+Lap), and NTF (Multi)

Method	$p = 0.1$	$p = 0.01$	$p = 0.005$	$p = 0.001$
Proposed 1	**0.27 (0.00)**	**0.46 (0.01)**	**0.70 (0.01)**	**0.92 (0.01)**
Proposed 2	0.28 (0.00)	0.51 (0.02)	0.71 (0.02)	0.94 (0.00)
ADMM	0.28 (0.00)	0.53 (0.01)	0.73 (0.01)	0.94 (0.00)
Multi+Lap	0.28 (0.00)	0.61 (0.08)	0.78 (0.02)	1.19 (0.16)
Multi	0.28 (0.00)	0.60 (0.08)	0.81 (0.04)	1.18 (0.24)

Table 5. ND for the bike-sharing record data of Washington D.C.

Method	$p = 0.1$	$p = 0.01$	$p = 0.005$	$p = 0.001$
Proposed 1	**0.81 (0.00)**	**0.91 (0.01)**	**1.04 (0.02)**	**1.04 (0.08)**
Proposed 2	**0.81 (0.01)**	**0.91 (0.00)**	**1.04 (0.02)**	1.05 (0.09)
ADMM	**0.81 (0.01)**	**0.91 (0.01)**	1.10 (0.01)	1.20 (0.01)
Multi+Lap	**0.81 (0.00)**	1.19 (0.05)	1.51 (0.15)	1.99 (0.54)
Multi	**0.81 (0.00)**	9.70 (6.91)	1.44 (0.10)	2.56 (0.31)

Table 6. ND for the bike-sharing record data of New York

Method	$p = 0.1$	$p = 0.01$	$p = 0.005$	$p = 0.001$
Proposed 1	**0.60 (0.00)**	**0.72 (0.01)**	**0.81 (0.01)**	**0.93 (0.01)**
Proposed 2	**0.60 (0.00)**	0.73 (0.01)	0.82 (0.01)	**0.93 (0.01)**
ADMM	**0.60 (0.00)**	0.74 (0.01)	0.84 (0.01)	0.94 (0.01)
Multi+Lap	**0.60 (0.00)**	2.13 (1.48)	1.53 (0.21)	1.05 (0.05)
Multi	**0.60 (0.00)**	4.48 (0.79)	2.51 (0.95)	1.21 (0.07)
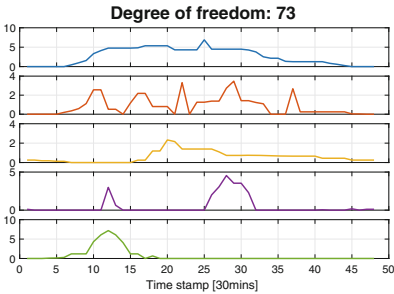
Fig. 2. Time factors of Proposed 1 on the traffic flow data (Color figure online)
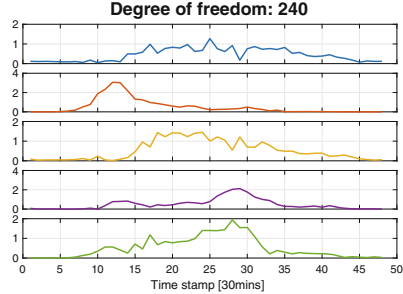
Fig. 3. Time factors of Multi+Lap on the traffic flow data (Color figure online)

always outperformed the same model estimated by the multiplicative update rule (Multi+Lap). This result was caused by the benefits of simultaneously combining graph-based and structured regularizers with graph and group structures. Thus our proposed model and parameter estimation procedure both contributed to the improvements on missing value interpolations. The existing methods resulted in poor performances with settings where a large portions of tensor elements were missing.

To check the qualitative performances of the interpretability, we showed the extracted factors of proposed method (Proposed 1) and existing NTF with the Laplacian regularizer (Multi+Lap) from traffic flow data in Figs. 2, 3, 4, 5, 6, and 7, where $p = 0.1$. The degree of freedom (DoF) in Figures corresponded the number of segments in a factor matrix. Thanks to the Laplacian and structured regularizers, proposed method extracted the interpretable latent factors in which both smooth and flat properties appeared, whose DoF of parameters in factor matrices were extremely less than that of NTF with the Laplacian regularizer. Our factors with low DoF were easy to find change points. For example, the blue factor had a change at 3 am and gradually grew until 6 am. Then it took the constant values until 3 pm in Fig. 2. This factor also has the same value from day 2 to day 6 and from day 8 to day 13. Thus, we can easily understand that the blue factor in Figs. 2 and 4 corresponded to activity that occurred in weekday during daylight with a spatial pattern in Fig. 6. However, NMF with the Laplacian
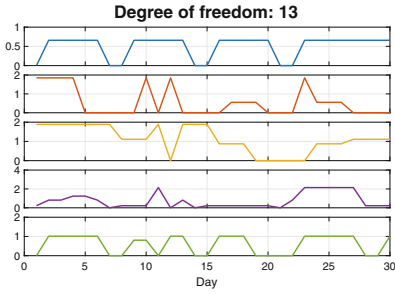
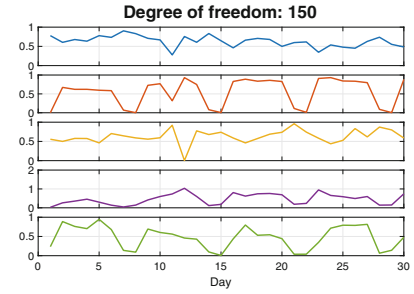**Fig. 4.** Day factors of Proposed 1 on the traffic flow data (Color figure online)

Fig. 5. Day factors of Multi+Lap on the traffic flow data (Color figure online)

Fig. 6. A spatial pattern of the blue factor of Proposed 1 on the traffic flow data (Color figure online)

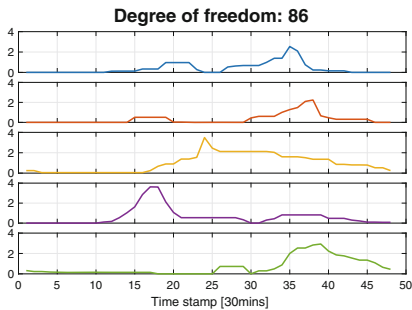**Fig. 7.** A spatial pattern of the blue factor of Multi+Lap on the traffic flow data (Color figure online)

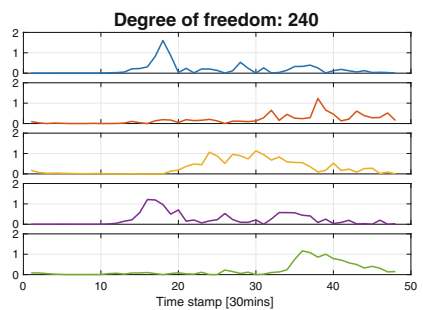**Fig. 8.** Time factors of Proposed 1 on the bike-sharing data of Washington D.C. (Color figure online)

Fig. 9. Time factors of Multi+Lap on the bike-sharing data of Washington D.C. (Color figure online)
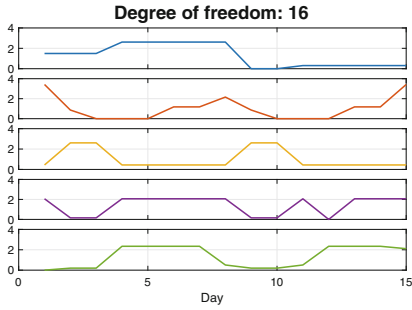
Fig. 10. Day factors of Proposed 1 on the bike-sharing data of Washington D.C. (Color figure online)
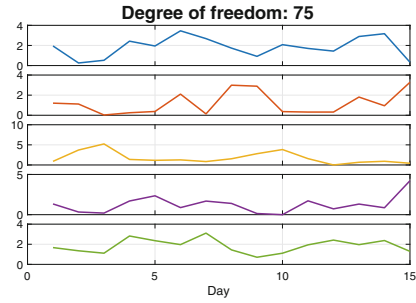

Fig. 11. Day factors of Multi+Lap on the bike-sharing data of Washington D.C. (Color figure online)
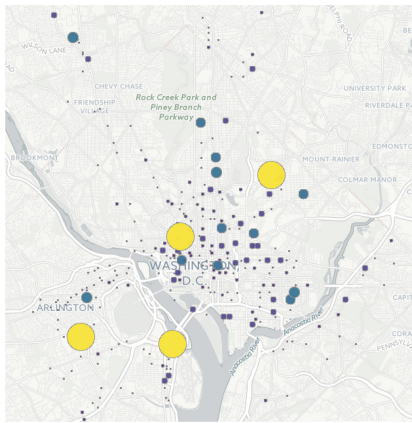

Fig. 12. A spatial pattern of the yellow factor of Proposed 1 on the bike-sharing data of Washington D.C. (Color figure online)
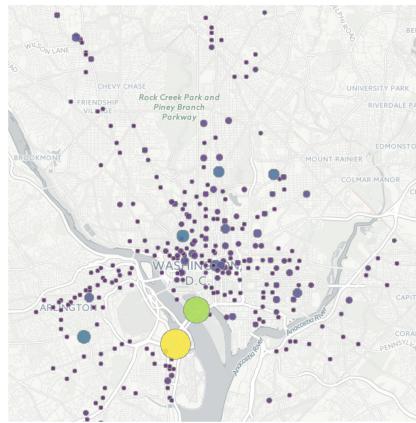

Fig. 13. A spatial pattern of the yellow factor of Multi+Lap on the bike-sharing data of Washington D.C. (Color figure online)

regularizer resulted in messy factors. We also showed that of bike-sharing data in Washington D.C. in Figs. 8, 9, 10, 11, 12, and 13. Our proposed method also extracted more interpretable patterns than existing NTF. For example, the yellow factor of ours in Fig. 8 had a change point at 8 am. After it had taken a peak at 12 am, it kept the same value from 1 pm to 5 pm. Then its value gradually decreased to zero. The yellow factor in Fig. 10 had the same high value on day 2, 3, 9, and 10. Thus, we confirmed that this factor indicated a weekend afternoon activity with a spatial pattern in Fig. 12. Similar interpretations can be obtained from other factors of ours.

7   Conclusion

In this paper, we proposed a structurally regularized non-negative tensor factorization that incorporated both the graph Laplacian and the structured regularizers on latent factors. For the structured regularizer, we employed the generalized fused lasso and the higher-order fused lasso to represent both graph-based and group-based information in time and space. We introduced a flexible and efficient parameter estimation method based on the alternating direction method of multipliers and showed a proximity operator for our unified structured regularizer. With experiments on a missing value imputation problem of three data sets, we confirmed that our proposed method showed the best quantitative performance and successfully extracted more interpretable latent factors than the existing non-negative tensor factorization methods.

A   Appendix

Although the issue in Eq. (12) is a general problem containing the previous problems [5,24,30] as special cases, we can solve it in a similar manner as these works. We first briefly introduce the parametric optimization method for a non-decreasing set function. Let $\alpha \in \mathbb{R}_{\geq 0}$, and define set function $l_\alpha(S) = l(S) - \alpha \mathbf{1}(S)$ ($\forall S \subset V$), where $\mathbf{1}(S) = \sum_{i \in S} 1$. Then if l is a non-decreasing submodular function, then there exists a set of $r + 1$ ($\leq |V|$) subsets: $S^* = \{S_0 \subset S_1 \subset \cdots \subset S_r\}$, where $S_j \subset V$, $S_0 = \emptyset$, and $S_r = V$, and $r + 1$ subintervals Q_r of α: $Q_0 = [0, \alpha_0), Q_1 = [\alpha_1, \alpha_2), \cdots, Q_r = [\alpha_r, \infty)$, such that, for each $j \in \{0, 1, \cdots, r\}$, S_j is the unique maximal minimizer of $h_\alpha(S), \forall \alpha \in Q_j$. A minimizer of Eq. (12) $\boldsymbol{t}^* = (t_1^*, t_2^*, \cdots, t_{|V|}^*)$ is then determined: $t_i^* = \frac{f(S_{j+1}) - f(S_j)}{\mathbf{1}(S_{j+1} \setminus S_j)}, \ \forall i \in (S_{j+1} \setminus S_j), \ j = (1, \cdots, r)$. We introduce two lemmas [20] to see that l is a non-decreasing submodular function.

Lemma 1 (Lemma). *For any $\eta \in \mathbb{R}$ and submodular function h, $\boldsymbol{t}^*$ is an optimal solution to $\min_{\boldsymbol{t} \in \mathcal{B}(l)} \|\boldsymbol{t}\|_2^2$ if and only if $\boldsymbol{t}^* - \eta \mathbf{1}$ is an optimal solution to $\min_{\boldsymbol{t} \in \mathcal{B}(l) + \eta \mathbf{1}} \|\boldsymbol{t}\|_2^2$.*

Lemma 2 (Lemma). *Set $\eta = \max_{i=1,\cdots,|V|} \{0, l(V \setminus \{i\}) - l(V)\}$, and then $l + \eta \mathbf{1}$ is a non-decreasing submodular function.*

With Lemma 2, we solve

$$\min_{S \subset V} f(S) - \hat{\boldsymbol{z}}(S) + (\eta - \alpha)\mathbf{1}(S), \tag{15}$$

and apply Lemma 1 to obtain a solution to the original problem. With fixed α, we can efficiently attain the optimal of Eq. (15) because this is a minimum cut problem.
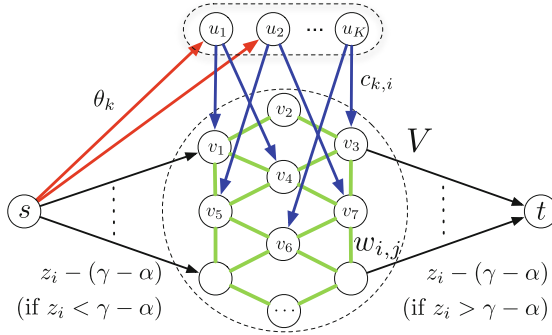
Fig. 14. Minimum s/t-cut problem of Problem (15). Given graph $G = (V', E')$ for our proposed method, capacities of edges $c(v'_i, v'_j)$ are defined as: $c(s, u_k) = \theta_k$, $c(v_i, v_j) = w_{i,j}$, $c(u_k, v_i) = c_{k,i}$, $c(s, v_i) = z_i - (\gamma - \alpha)$ if $z_i > \gamma - \alpha$, and $c(v_i, t) = (\gamma - \alpha) - z_i$ if $z_i < \gamma - \alpha$. Nodes u_k $k = (1, \cdots, K)$ are hyper nodes that correspond to the groups g_k. And s, t, and v_i are source, sink, and parameters nodes, respectively.

Proposition 1. *The problem in Eq. (15) is equivalent to a minimum s/t-cut problem.*

Proof. Each component in f is graph-representable. The graph is obtained due to the additive property of the graph-representative submodular functions, where the groups of parameters are represented with hyper nodes u_1^k, u_0^k that corresponds to each group, and the capacities of the edges between hyper and ordinal nodes $v_i \in V$.

The attained graph includes both of the GFL and HOFL graphs as spacial cases. As a consequence, we can attain a sequence of solutions for all α of the parametric s/t minimun-cut problem (15) using an efficient parametric-flow algorithm, such as [12], that runs in $O(|V'||E'| \log(|V'|^2/|E'|))$ as the worst case and $|V'|$ and $|E'|$ are the number of nodes and edges of the graph (Fig. 14).

References

1. Ali, M.I., Gao, F., Mileo, A.: CityBench: a configurable benchmark to evaluate RSP engines using smart city datasets. In: Arenas, M., Corcho, O., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P., Dumontier, M., Heflin, J., Thirunarayan, K., Staab, S. (eds.) ISWC 2015. LNCS, vol. 9367, pp. 374–389. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25010-6_25
2. Bach, F.R.: Structured sparsity-inducing norms through submodular functions. In: Proceedings of NIPS, pp. 118–126 (2010)
3. Barbero, A., Sra, S.: Fast Newton-type methods for total variation regularization. In: Proceedings of ICML, pp. 313–320 (2011)
4. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**(1), 1–122 (2011)

5. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. Pattern Anal. Mach. Intell. **26**(9), 1124–1137 (2004)
6. Cai, D., He, X., Han, J., Huang, T.S.: Graph regularized nonnegative matrix factorization for data representation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(8), 1548–1560 (2011)
7. Chambolle, A., Darbon, J.: On total variation minimization and surface evolution using parametric maximum flows. Int. J. Comput. Vis. **84**(3), 288 (2009)
8. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.-I.: Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. Wiley, Hoboken (2009)
9. Dhillon, I.S., Sra, S.: Generalized nonnegative matrix approximations with Bregman divergences. In: Proceedings of NIPS, vol. 18 (2005)
10. Fan, Z., Song, X., Shibasaki, R.: CitySpectrum: a non-negative tensor factorization approach. In: Proceedings of UbiComp, pp. 213–223 (2014)
11. Fujishige, S.: Submodular Functions and Optimization, vol. 58. Elsevier, Amsterdam (2005)
12. Gallo, G., Grigoriadis, M.D., Tarjan, R.E.: A fast parametric maximum flow algorithm and applications. SIAM J. Comput. **18**(1), 30–55 (1989)
13. Gillis, N.: The why and how of nonnegative matrix factorization. In: Regularization, Optimization, Kernels, and Support Vector Machines, pp. 257–291. Chapman and Hall/CRC (2014)
14. Han, Y., Moutarde, F.: Analysis of large-scale traffic dynamics in an urban transportation network using non-negative tensor factorization. Int. J. Intell. Transp. Syst. Res. **14**(1), 36–49 (2016)
15. Huang, K., Sidiropoulos, N.D., Liavas, A.P.: A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. IEEE Trans. Sig. Process. **64**(19), 5052–5065 (2016)
16. Kimura, T., Ishibashi, K., Mori, T., Sawada, H., Toyono, T., Nishimatsu, K., Watanabe, A., Shimoda, A., Shiomoto, K.: Spatio-temporal factorization of log data for understanding network events. In: Proceedings of INFOCOM, pp. 610–618 (2014)
17. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
18. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788–791 (1999)
19. Lovász, L.: Submodular functions and convexity. In: Bachem, A., Korte, B., Grötschel, M. (eds.) Mathematical Programming The State of the Art, pp. 235–257. Springer, Heidelberg (1983). https://doi.org/10.1007/978-3-642-68874-4_10
20. Nagano, K., Kawahara, Y., Aihara, K.: Size-constrained submodular minimization through minimum norm base. In: Proceedings of ICML, pp. 977–984 (2011)
21. Rao, N., Yu, H.-F., Ravikumar, P.K., Dhillon, I.S.: Collaborative filtering with graph information: consistency and scalable methods. In: Proceedings of NIPS, pp. 2107–2115 (2015)
22. Smola, A.J., Kondor, R.: Kernels and regularization on graphs. In: Schölkopf, B., Warmuth, M.K. (eds.) COLT-Kernel 2003. LNCS (LNAI), vol. 2777, pp. 144–158. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45167-9_12
23. Sun, L., Axhausen, K.W.: Understanding urban mobility patterns with a probabilistic tensor factorization framework. Transp. Res. Part B: Methodol. **91**, 511–524 (2016)

24. Takeuchi, K., Kawahara, Y., Iwata, T.: Higher order fused regularization for supervised learning with grouped parameters. In: Appice, A., Rodrigues, P.P., Santos Costa, V., Soares, C., Gama, J., Jorge, A. (eds.) ECML PKDD 2015. LNCS (LNAI), vol. 9284, pp. 577–593. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23528-8_36
25. Takeuchi, K., Tomioka, R., Ishiguro, K., Kimura, A., Sawada, H.: Non-negative multiple tensor factorization. In: Proceedings of ICDM, pp. 1199–1204 (2013)
26. Tandon, R., Sra, S.: Sparse nonnegative matrix approximation: new formulations and algorithms. Rapp. Tech. **193**, 38–42 (2010)
27. Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., Knight, K.: Sparsity and smoothness via the fused lasso. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **67**(1), 91–108 (2005)
28. Wang, Y.-X., Sharpnack, J., Smola, A., Tibshirani, R.J.: Trend filtering on graphs. J. Mach. Learn. Res. **17**(105), 1–41 (2016)
29. Witten, D.M., Tibshirani, R., Hastie, T.: A penalized matrix decomposition with applications to sparse principal components and canonical correlation analysis. Biostatistics **10**, 515–534 (2009)
30. Xin, B., Kawahara, Y., Wang, Y., Gao, W.: Efficient generalized fused lasso with its application to the diagnosis of Alzheimer's disease. In: Proceedings of AAAI, pp. 2163–2169 (2014)
31. Yangyang, X., Yin, W.: A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM J. Imaging Sci. **6**(3), 1758–1789 (2013)
32. Yang, D., Zhang, D., Zheng, V.W., Yu, Z.: Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. IEEE Trans. Syst. Man Cybern.: Syst. **45**(1), 129–142 (2015)