Data Mining and Knowledge Discovery (2022) 36:1247–1278
https://doi.org/10.1007/s10618-022-00842-3

Individualized passenger travel pattern multi-clustering based on graph regularized tensor latent dirichlet allocation

Ziyue Li[1] · Hao Yan[2] · Chen Zhang[3] · Fugee Tsung[4]

Received: 30 September 2021 / Accepted: 20 May 2022 / Published online: 18 June 2022

Abstract

Individual passenger travel patterns have significant value in understanding passenger's behavior, such as learning the hidden clusters of locations, time, and passengers. The learned clusters further enable commercially beneficial actions such as customized services, promotions, data-driven urban-use planning, peak hour discovery, and so on. However, the individualized passenger modeling is very challenging for the following reasons: 1) The individual passenger travel data are multi-dimensional spatiotemporal big data, including at least the origin, destination, and time dimensions; 2) Moreover, individualized passenger travel patterns usually depend on the external environment, such as the distances and functions of locations, which are ignored in most current works. This work proposes a multi-clustering model to learn the latent clusters along

Responsible editor: Albrecht Zimmermann and Peggy Cellier.

Hao Yan, Chen Zhang and Fugee Tsung have contributed equally to this work.

✉ Ziyue Li
zlibn@wiso.uni-koeln.de

Hao Yan
haoyan@asu.edu

Chen Zhang
zhangchen01@tsinghua.edu.cn

Fugee Tsung
season@ust.hk

[1] Information Systems, Faculty of Management, Economics and Social Sciences, University of Cologne, Cologne, Germany

[2] School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, 699 S Mill Av, Tempe 85281, Arizona, USA

[3] Industrial Engineering, Tsinghua University, 602 Shunde Building, Beijing 100084, China

[4] Department of Industrial Engineering and Decision Analytics, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

the multiple dimensions of Origin, Destination, Time, and eventually, Passenger (ODT-P). We develop a graph-regularized tensor Latent Dirichlet Allocation (LDA) model by first extending the traditional LDA model into a tensor version and then applies to individual travel data. Then, the external information of stations is formulated as semantic graphs and incorporated as the Laplacian regularizations; Furthermore, to improve the model scalability when dealing with massive data, an online stochastic learning method based on tensorized variational Expectation-Maximization algorithm is developed. Finally, a case study based on passengers in the Hong Kong metro system is conducted and demonstrates that a better clustering performance is achieved compared to state-of-the-arts with the improvement in point-wise mutual information index and algorithm convergence speed by a factor of two.

Keywords Individualized analysis · Tensor · Topic model · Graph structure · Online algorithm · Spatiotemporal data

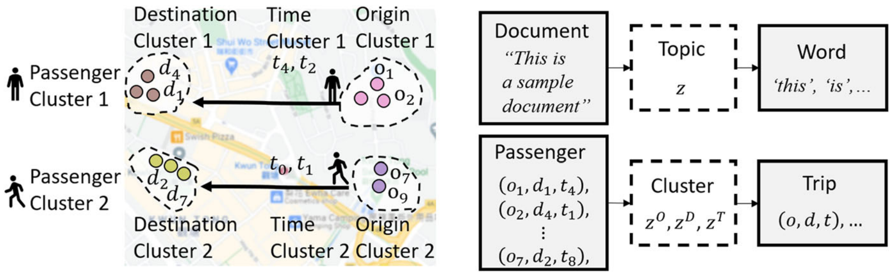
1 Introduction

The public transportation system is the backbone of a city's infrastructure, and the intelligent transportation system (ITS) has been an essential chapter for the smart city blueprint. Most studies for ITS focus on traffic flow prediction (Ren and Xie 2017; Geng et al. 2019; Guo et al. 2019; Shi et al. 2020; Wang et al. 2019; Li et al. 2020). Tensor-based methods, such as tensor decomposition (Ren and Xie 2017), tensor completion (Li et al. 2020), as well as deep learning methods, such as convolutional neural networks (Geng et al. 2019), graph convolutional networks (Yu et al. 2018), and spatiotemporal attentions (Guo et al. 2019), have been developed to predict city-wide traffic flow (Geng et al. 2019), metro station-level passenger flow (Li et al. 2020), origin-destination (OD) flow matrix (Ren and Xie 2017; Wang et al. 2019; Shi et al. 2020), etc.

However, the methods mentioned above target traffic flow prediction at a macro level and utilize the traffic data of passengers indiscriminately, consequently neglecting the personalized travel characteristics of individual passengers. For example, to calculate the passenger flow value, only the number of passengers is counted, which abandons individual information (Yi et al. 2019). Thus, those methods could not directly handle individual travel data.

To overcome this issue, we propose to fully utilize the "individual" travel data for an "individualized" travel pattern discovery. Individual travel data preserve the abundant trajectory information, i.e., that passenger u departs from origin o at time t and arrives at destination d at time t'. This encourages us to focus on the following individualized analysis tasks due to their high research values (Zhao et al. 2017). We aim at the following two goals:

- **Clustering of origin, destination, and time (ODT)**: The latent clusters for origin, destination, and time could be better learned from individual travel pattern data, given the abundant information is well preserved. The intuition is that those origins may belong to the same cluster if they all co-occur in the same type of passengers,



(a) Different Travel Patterns **(b)** Analogy from Document to Passenger

Fig. 1 **(a)** Different passengers travel from O cluster to D cluster at T cluster; **(b)** Analogy from document to passenger

as shown in Fig. 1(a). The learned clusters for ODT could guide better urban planning, more suitable station-surrounding facilities, and uncover the peak hour for crowd control.

- **Clustering of passengers**: Passengers will also be clustered into different groups based on their trajectories. For public transport providers, with a better understanding of individual passengers' travel patterns, customized promotions and more suitable operational policies can be designed. For example, the fare surcharge-reward scheme could be tailored for different passenger groups (Tang et al. 2020).

The two clustering tasks above for **O**rigin, **D**estination, **T**ime, and **P**assenger are called for short as "ODT-P Multi-clustering". However, these two tasks are rather challenging due to the multi-mode spatiotemporal big data and the influence from the external environment.

- **Challenge 1: Multi-mode spatiotemporal big data**. Take Hong Kong as an example, there are 2 million passengers daily: Each passenger has multiple trips, and each trip has multiple modes such as origin, destination, and time.
- **Challenge 2: external environment**. Moreover, passenger behaviors are also affected by the external environment, such as the locations and surroundings of stations. If two stations are geographically adjacent to each other or located in similar functional areas (such as business area, residential area, school), they will attract the same type of passengers.

To tackle the aforementioned challenges, we propose a novel Graph-Regularized Tensor Latent Dirichlet Allocation model (GR-TensorLDA) for ODT-P multi-clustering based on individual passenger travel patterns. First, to preserve the multi-mode structure of the high dimensional spatiotemporal data, we focus on the tensor-based methodology (Kolda and Bader 2009), which represents the original data with three-mode tensors, where different modes represent ODT respectively. Secondly, a tensor LDA model is proposed to achieve ODT-P multi-clustering.

The main novelty of our proposed method is that we extend the traditional LDA (Blei et al. 2003) to a tensor version and apply into individual traffic data. An important analogy is made as shown in Fig. 1(b):

(1) "Word"-level: A trip is viewed as a three-dimensional word $\boldsymbol{w} = (w^O, w^D, w^T)$; (2) "Document"-level: A passenger with several trips, i.e., "a bag of words", is

treated as a three-dimensional document $\mathbf{d}_{\boldsymbol{u}} \in \mathrm{R}^{O\times D\times T}$. Generative processes in the passenger-level and trip-level will be defined along with each mode of ODT; (3) "Topic"-level: Therefore, the latent topic will also be formulated as a tensor, with each element as $\boldsymbol{z} = (z^O, z^D, z^T)$.

The clusters of ODT-P will be eventually obtained in the following way: (1) ODT clustering: Along each dimension of ODT, the topic is a latent distribution of words, which can be viewed as a cluster containing different words; (2) P clustering: each passenger is represented by the latent distribution of the tensor topics, which will be utilized to cluster passengers.

Our most significant technical contributions are twofold:

- **Semantic graph structure**: To tackle Challenge 2, we incorporate the external environment as graph structures into the model. Precisely, we first formulate the station-related information as two graphs: (1) A geographical graph measures the spatial distance between stations; (2) A contextual graph quantifies whether two stations are located in similar functional areas (Geng et al. 2019; Li et al. 2020). Then the graph structures are incorporated into the tensor LDA generative process for OD, such that if two stations are close on these two graphs, they are more likely to be in the same topic. We show that by adding such graph regularizations, the interpretability of the learned ODT-P clusters can be significantly improved.
- **Efficient online algorithm**: Since the graph regularization breaks the conjugacy, standard optimization techniques such as Gibbs sampling (Griffiths and Steyvers 2004) are no longer possible, we propose a tensorized variational Expectation-Maximization (EM) algorithm to estimate parameters. Moreover, to tackle Challenge 1, we need an efficient and scalable algorithm to deal with massive passenger data. Therefore, we further propose to conduct the algorithm in an online stochastic learning manner (Hoffman et al. 2010). We show that to reach the same level of performance, the online learning algorithm converges twice faster than the batch learning algorithm.

The remainder of the paper is structured as follows. Section 2 briefly reviews existing tensor methods, individual travel analysis, and topic models. Section 3 formulates the proposed model, and Section 4 proposes an efficient optimization algorithm. Section 5 provides a detailed experiment to demonstrate the improved meaningfulness of the learned clusters and model scalability; Section 6 gives the conclusion and discusses the future work and model generalization.

2 Related works

2.1 Tensor and tucker decomposition

We would like to first introduce tensor and tensor decomposition since high dimensional data are usually formulated as a tensor, and tensor decomposition is widely used for clustering (Kolda and Bader 2009; Sun and Axhausen 2016). Tensor is mathematically defined as a multi-dimensional array, which is believed to have sufficient capacity to preserve innate complex correlations across multiple dimensions

(Kolda and Bader 2009). One of the most popular techniques is tensor decomposition. Tucker decomposition is a high-order principal component analysis. It decomposes a tensor $\mathcal{X}^{O\times D\times T}$ into a core tensor $\mathcal{C}^{J\times K\times L}$ multiplied by a mode matrix along each dimension, $\mathbf{U}^O, \mathbf{U}^D, \mathbf{U}^T$: $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{C}} \times_1 \mathbf{U}^O \times_2 \mathbf{U}^D \times_3 \mathbf{U}^T$: Tensor decomposition has been applied into smart transportation for prediction (Li et al. 2020) and clustering (Sun and Axhausen 2016). However, tensor decomposition might be rather impractical to be applied to our problem. The main reason is: data formulated with each individual passenger are extremely sparse due to curse of dimensionality. According to our preliminary study (Li et al. 2021), the sparsity could reach 99.97%, which paralyzes traditional tensor decomposition methods (Tang et al. 2018). Therefore, a technique that specializes in individual analysis is needed.

2.2 Individual travel pattern analysis

The new generation of ITS aims to be more personalized. Recently individual travel data have been utilized for passenger clustering (Briand et al. 2017; Mohamed et al. 2016), station clustering (Mohamed et al. 2016), and personalized services such as travel time estimation (Tang et al. 2018), route recommendation (Liu et al. 2019), destination inference (Cheng et al. 2020), driving state recognition (Yi et al. 2019), and activity discovery (Zhao et al. 2020). There are mainly two kinds of approaches as follows.

2.2.1 Spatio-temporal feature engineering

The first kind of approach relies on intense feature engineering to extract useful features such as spatial, temporal, OD pair, transportation mode. It then combines the features with traditional statistical learning models for clustering and prediction, such as K-mean clustering (Zhao et al. 2017), boosting tree (Liu et al. 2019), random forest (Yi et al. 2019). However, the feature extraction is rather complicated and differs from one system to another, which does not offer a universal solution. Furthermore, it typically assumes that the feature has the same dimension for each passenger. However, the number of trips of each passenger can be dramatically different. In contrast, our model learns the latent dimension in a data-driven way and can be accommodated to different numbers of trips, which offers a general solution with explainable results.

2.2.2 Generative models

As the second option, generative models (Briand et al. 2017; Mohamed et al. 2016; Tang et al. 2018; Cheng et al. 2020; Zhao et al. 2020) have been adopted into individual traffic analysis. To cluster passengers' temporal behaviors, Briand et al. (2017) and Mohamed et al. (2016) proposed two-layer generative models with a mixture of Gaussian or unigrams model: The first layer partitions passengers into clusters, and the second layer formulates each cluster's temporal activity. However, the limitation is that passengers are only clustered based on their active or boarding time; Therefore, the passengers' abundant spatiotemporal information is not fully utilized. As a result, no insights about the latent nature of origins and destinations could be obtained.

To capture all dimensions of the spatiotemporal information for individual passengers, researchers have adopted topic models into individual traffic data (Cheng et al. 2020; Zhao et al. 2020), where an individual passenger's travel data is regarded as a document, where each trip is recorded as a word. Specifically, Cheng et al. (2020) and Zhao et al. (2020) proposed a high-dimensional LDA model with a generative process on each dimension, such as location, day, hour, and trip duration. However, the existing methods ignored the underlining spatial correlations in the passengers' travel data, which may lead to a clustering model that could not reflect reality. Compared with them, our most significant advantages are that we incorporate semantic graphs into the LDA generative process. This is inspired by the state-of-the-art topic models (Yao et al. 2017; Li et al. 2019b) that incorporates knowledge graph to improve the interpretability of model output, which we will review them in details in the following section. Such coupling of the "pure-trip" data with external contexts significantly improves topics' interpretability, and that we propose an efficient online stochastic learning algorithm based on a variational EM algorithm.

2.3 Graph-based topic models

Incorporating knowledge graphs into topic models could enhance the interpretability of the learned topics (Yao et al. 2017; Li et al. 2019a; Mei et al. 2008; Chen et al. 2016; Li et al. 2019b). In particular, two categories of incorporating methods are considered in state-of-the-art models. The first category embeds words into a continuous space with word relations defined by an external knowledge graph such as DBpedia, WordNet (Yao et al. 2017; Li et al. 2019a). However, in traffic data, such knowledge graph is only applied to a single word representation, which cannot be used for high-dimensional word representation. The second category introduces graph-based regularization (Mei et al. 2008; Chen et al. 2016; Li et al. 2019b), such as graph harmonic function, to encourage entities close on the graph to be more likely to have the same topic. These regularization-based techniques are compatible with our generative model. However, the existing graph-based topic models are formulated only for one-dimensional word, not for high-dimensional data like our passenger travel data. Moreover, the challenges lie in the parameter learning for our corresponding tensor topic model. To this end, we rigorously develop online stochastic learning based on tensorized variational EM algorithm to estimate parameters with higher efficiency and scalability.

2.4 Multi-view subspace clustering

Last but not the least, subspace clustering is also a popular method for high-dimensional clustering (Parsons et al. 2004), which learns data representation in certain low-dimensional subspaces and clusters the data points. Multi-view subspace clustering (Gao et al. 2015; Zhang et al. 2017, 2018) specifically deals with data represented by multiple distinct feature sets.

We would like to emphasize the difference between our model and subspace clustering from the perspectives of data and model: (1) The typical multi-view data are formulated as $\mathbf{X}_v \in \mathbb{R}^{d_v \times n}$, where d_v and n are the number of features and samples on the v-th view. Our data instead present a hierarchical structure: a passenger has a sequence of a few trip records, and each trip is an instance in a three-dimensional space of ODT. Moreover, our data suffer from high sparsity, which also hinders subspace clustering, e.g., factorization-based methods, from normal functioning. (2) A typical formulation of multi-view subspace clustering is based on the data's self-expression property (Gao et al. 2015), which is to use the data set to represent itself: $\mathbf{X}_v = \mathbf{X}_v\mathbf{Z}_v + \mathbf{E}_v$, where $\mathbf{Z}_v \in \mathbb{R}^{n\times n}$ is the subspace representation matrix of the v-th view, and the nonzero elements in $\mathbf{Z}_v$ correspond to the data points from the same subspace. Various methods are proposed to add different regularizations on $\mathbf{Z}_v$, such as sparsity (Elhamifar and Vidal 2013), low rank (Liu et al. 2012) and smoothness (Hu et al. 2014). However, self-expression property cannot be applied to our model since our data are already high-dimensional and extremely sparse, which may lead to a even higher-dimensional and more sparse $\mathbf{Z}$.

3 Proposed methodology

We will introduce the proposed methodology. Section 3.1 gives the data formulation and the notations; Section 3.2 introduces the tensor topic and tucker decomposition; Section 3.3 formulates the generative process along each dimension; Section 3.4 formulates the graph structure for dimension OD; Section 3.5 gives the final loss function. The concepts of "passenger" and "document", "trip" and "word", "topic" and "cluster" are interchangeably used here.

3.1 Data representation and notation

Firstly, the notations throughout this paper are as follows: We denote scalars in italics, e.g., n, vectors by lowercase letters in boldface, e.g., $\boldsymbol{\beta}$, matrices by uppercase boldface letters, e.g., $\mathbf{B}$, and tensors by boldface script capital $\boldsymbol{\mathcal{W}}$.

Then, we would like to define the data representation. A trip is defined as a three-dimensional tuple (i.e., word) $\boldsymbol{w} = (w^O, w^D, w^T)$, indicating a trip that starts from origin w^O at time w^T and heads to destination w^D. V^O, V^D, V^T are the vocabulary sizes for ODT respectively. A passenger who has traveled several trips is regarded as "a bag of words" (i.e., document): $\mathbf{d}_u = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{\underline{i}}, \ldots, \boldsymbol{w}_{N_u}\}$, with $\underline{i}$ as the i-trip of the passenger u, N_u as the number of trips from passenger $u \in \mathbb{R}^M$, and M as the total number of passengers. All the notations in the paper are summarized in Table 1.

3.2 Tensor topic definition and decomposition

The topic for the i-th word $\boldsymbol{w}_{\underline{i}}$ in passenger u is also formulated as a three-element tuple $\mathbf{z}_{j,k,l} = (z^O_{\underline{i}j}, z^D_{\underline{i}k}, z^T_{\underline{i}l})(\underline{i} \in \mathbb{R}^{N_u}, j \in \mathbb{R}^J, k \in \mathbb{R}^K, l \in \mathbb{R}^L)$, where J, K, L are the number of latent topics of ODT respectively, and $(z^O_{\underline{i}j}, z^D_{\underline{i}k}, z^T_{\underline{i}l})$ indicates the i-th word belongs to the j-th 'O' topic, k-th 'D' topic and l-th 'T' topic, respectively (Cheng et al. 2020).

Table 1 Notation

Symbols	Description
$()^{O,D,T}$	Superscripts O, D, T mean three dimensions: Origin, Destination, Time respectively
$\mathcal{W}_u$	The tensor ODT data for each passenger u
M	The total number of passengers (iterator u)
$\mathbf{d}_u$	The trip sets of u-th passenger
N_u	The number of trips (words) in $\mathbf{d}_u$ (iterator $\underline{i}$)
J, K, L	Amount of topics for ODT (iterator j, k, l) respectively
$\mathcal{A}(\alpha_{j,k,l})$	$\mathcal{A} \in \mathbb{R}^{J\times K\times L}$ 3d-Dirichlet distribution parameter
$\mathcal{C}_u(c_{u,j,k,l})$	$\mathcal{C}_u \in \mathbb{R}^{J\times K\times L}$ topic distribution for passenger u
$z^O_{\underline{i}}, z^O_j, z^O_{\underline{i}j}$	$z^O_{\underline{i}}$: the origin topic for i-th word; z^O_j: the origin topic is j; $z^O_{\underline{i}j}$: the origin topic for i-th word is j
$w^O_{\underline{i}}, w^O_o$	$w^O_{\underline{i}}$: the origin element of the i-trip; w^O_o: the word in Origin dimension is o, $w^O = o$
V^O, V^D, V^T	The vocabulary of all unique ODT
$\mathbf{B}^O(\beta^O_{jo})$	$\mathbf{B}^O \in \mathbb{R}^{J\times V^O}$; its element β^O_{jo}: the multinomial parameter for word $o \in V^O$ drawn from topic j.
$\boldsymbol{\Phi}^O(\phi^O_{\underline{i}j})$	Variational variable for z^O, the probability of the word at i-th position generated from j-th topic.
$\mathcal{E}_u(\epsilon_{s,j,k,l})$	$\mathcal{E}_u \in \mathbb{R}^{J\times K\times L}$ variational variable for $\mathcal{C}_u$

According to Bayes' theorem, the probability of the i-trip $\boldsymbol{w}_{\underline{i}} = (w^O_{\underline{i}}, w^D_{\underline{i}}, w^T_{\underline{i}})$ from passenger $\mathbf{d}_u$ can be written as:

$$\begin{aligned} P(\boldsymbol{w}_{\underline{i}} = (o, d, t) \mid \mathbf{d}_u) = \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{L} P(z^O_{\underline{i}j}, z^D_{\underline{i}k}, z^T_{\underline{i}l} \mid \mathbf{d}_u)\times \\ P(w^O_{\underline{i}} = o \mid z^O_{\underline{i}j})P(w^D_{\underline{i}} = d \mid z^D_{\underline{i}k})P(w^T_{\underline{i}} = t \mid z^T_{\underline{i}l}). \end{aligned} \tag{1}$$

We denote $P(z^O_{\underline{i}j}, z^D_{\underline{i}k}, z^T_{\underline{i}l} \mid \mathbf{d}_u) = c_{u,j,k,l}$ as the probability that topic $\mathbf{z}$ for i-trip in passenger $\mathbf{d}_u$ is (j, k, l); We further denote $P(w^O_{\underline{i}} = o \mid z^O_{\underline{i}j}) = \beta^O_{jo}$ as the probability of w^O in i-trip is o given the j-th origin topic; Similarly for dimension D and T we have $P(w^D_{\underline{i}} = d \mid z^D_{\underline{i}k}) = \beta^D_{kd}$, $P(w^T_{\underline{i}} = t \mid z^T_{\underline{i}l}) = \beta^T_{lt}$.

Tucker Decomposition: Eq. (1) can be presented in Tucker decomposition as follows:

$$\mathcal{W}_u = \mathcal{C}_u \times_1 \mathbf{B}^O \times_2 \mathbf{B}^D \times_3 \mathbf{B}^T. \tag{2}$$

The essence of the model is revealed as probabilistic tucker decomposition (Kolda and Bader 2009), where the tensor ODT data for each passenger u is $\mathcal{W}_u \in \mathbb{R}^{V^O\times V^D\times V^T}$, which is decomposed into a core tensor $\mathcal{C}_u \in \mathbb{R}^{J\times K\times L}$, and along

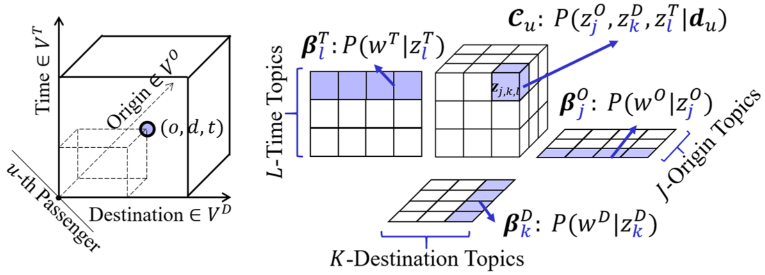

Time ∈ V^T
Origin ∈ V^O
(o, d, t)
u-th Passenger
Destination ∈ V^D
$\boldsymbol{\beta}_l^T : P(w^T|z_l^T)$
L-Time Topics
$\mathcal{C}_u$: $P(z_j^O, z_k^D, z_l^T | \boldsymbol{d}_u)$
$\mathbf{z}_{j,k,l}$
$\boldsymbol{\beta}_j^O$: $P(w^O|z_j^O)$
J-Origin Topics
$\boldsymbol{\beta}_k^D$: $P(w^D|z_k^D)$
K-Destination Topics

Fig. 2 Tensor topic and tucker decomposition

each dimension there is a mode matrix $\mathbf{B}^O \in \mathbb{R}^{J\times V^O}$, $\mathbf{B}^D \in \mathbb{R}^{K\times V^D}$, $\mathbf{B}^T \in \mathbb{R}^{L\times V^T}$ as shown in Fig. 2.

It is worth mentioning that although the essence of the model is a decomposition, yet since $\boldsymbol{\mathcal{W}}_u$ is intractable, $\boldsymbol{\mathcal{C}}_u$, $\mathbf{B}^O$, $\mathbf{B}^D$, $\mathbf{B}^T$ could not be learned by decomposing $\boldsymbol{\mathcal{W}}_u$. Instead, we learn the latent parameters first and then $\boldsymbol{\mathcal{W}}_u$ could be calculated.

3.3 Generative process

The generative process for a trip $\boldsymbol{w}$ will be defined along ODT.

Prior: Dirichlet distribution is known as a good conjugate prior for multinomial distribution (Zhou 2018). The tensor topic distribution $\boldsymbol{\mathcal{C}}_u \in \mathbb{R}^{J\times K\times L}$ for the u-th passenger is generated from the 3D-Dirichlet distribution with parameter $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{J\times K\times L}$ and each element $c_{u,j,k,l}(\sum_j \sum_k \sum_l c_{u,j,k,l} = 1)$ defines the possibility for the passenger to have trips from topic $\mathbf{z}_{j,k,l}$:

$$\boldsymbol{\mathcal{C}}_u \sim \text{Dir}_{J\times K\times L}(\boldsymbol{\mathcal{A}}). \tag{3}$$

Passenger to Tensor Topic: The topic for the i-trip in the u-th passenger is drawn from the multinomial distribution:

$$z_{j,k,l} \mid \boldsymbol{w}_{\underline{i}} \sim \text{Multi}(c_{u,j,k,l}). \tag{4}$$

Topic to Trip: We define $\mathbf{B}^O \in \mathbb{R}^{J\times V^O}$ as the topic-trip matrix, in which the element β_{jo}^O is the multinomial probability that $w^O = o$ is drawn from the j-th origin topic, $P(w_{\underline{i}}^O = o \mid z_{\underline{i}j}^O)$. $\mathbf{B}^D$, $\mathbf{B}^T$ are defined the same way. Therefore the word (w^O, w^D, w^T) is drawn from each multinomial distribution with parameter $\mathbf{B}^O$, $\mathbf{B}^D$ and $\mathbf{B}^T$ respectively:

$$w^O \sim \text{Multi}(\boldsymbol{\beta}_j^O),\, w^D \sim \text{Multi}(\boldsymbol{\beta}_k^D),\, w^T \sim \text{Multi}(\boldsymbol{\beta}_l^T). \tag{5}$$

3.4 Graph structure on origin and destination

If two stations are geographically close to each other or located in the similar functional area, intuitively these two stations are more likely to be in the same topic. This external information will be formulated as a graph and then introduced into the model as the Laplacian regularization.

Precisely, two graphs are defined accordingly to capture the inter-relationships of different stations: (1) Geographical graph $\mathbf{G}_{net}$: describes how two stations are geographically close to each other on the network; (2) Functional similarity graph $\mathbf{G}_{poi}$: quantifies how similar the functions of two stations' locations are. These two graphs have effects on both OD dimensions, with definition details in Section 5.2.

The graph regularization term is defined as follows. Take one graph in origin dimension as an example,

$$R(\mathbf{G}^O) = \frac{1}{2} \sum_{o_1, o_2 \in \mathbf{G}^O} \kappa(o_1, o_2) \sum_{j=1}^{J} (\beta^O_{jo_1} - \beta^O_{jo_2})^2 = \frac{1}{2} \sum_{j=1}^{J} (\boldsymbol{\beta}^O_j)^T \mathbf{L} \boldsymbol{\beta}^O_j, \tag{6}$$

where $\mathbf{G}^O \in \mathbb{R}^{V^O \times V^O}$ is the graph on origin stations, $\kappa(o_1, o_2) = \{\mathbf{G}^O\}_{o_1, o_2}$ defines the weight between entity o_1 and o_2, and $\mathbf{L}$ is the Laplacian matrix for graph $\mathbf{G}^O$ (Li et al. 2020; Wang et al. 2015; Yu et al. 2019). The intuition is that two stations that are closer on the graph will be more likely to have the same topic (Mei et al. 2008).

With the corresponding Laplacian matrices as $\mathbf{L}_{net}$ and $\mathbf{L}_{poi}$, the final graph Laplacian penalty on OD could be formulated as:

$$\begin{aligned} R(\mathbf{G}^{O,D}) = &\frac{1}{2} \sum_{j} (\boldsymbol{\beta}^O_j)^T (\mu \mathbf{L}_{net} + (1 - \mu) \mathbf{L}_{poi}) \boldsymbol{\beta}^O_j \\ &+ \frac{1}{2} \sum_{k} (\boldsymbol{\beta}^D_k)^T (\nu \mathbf{L}_{net} + (1 - \nu) \mathbf{L}_{poi}) \boldsymbol{\beta}^D_k. \end{aligned} \tag{7}$$

where the tuning parameters μ and ν adjust the relative effect of two graphs on OD respectively. The whole generative process is shown in Fig. 3.

3.5 Loss function

In order to learn the model parameters $\boldsymbol{\mathcal{A}}, \mathbf{B}^O, \mathbf{B}^D$ and $\mathbf{B}^T$ for the GR-TensorLDA model, the log-likelihood function could be formulated as follows:

$$L(\boldsymbol{\mathcal{A}}, \mathbf{B}^O, \mathbf{B}^D, \mathbf{B}^T) = \sum_{u=1}^{M} \log P(\mathbf{d}_u \mid \boldsymbol{\mathcal{A}}, \mathbf{B}^O, \mathbf{B}^D, \mathbf{B}^T) \tag{8}$$

where $P(\mathbf{d}_u \mid \boldsymbol{\mathcal{A}}, \mathbf{B}^O, \mathbf{B}^D, \mathbf{B}^T)$ is the marginal distribution of a passenger which can be defined as: $P(\mathbf{d}_u \mid \boldsymbol{\mathcal{A}}, \mathbf{B}^O, \mathbf{B}^D, \mathbf{B}^T) = \int P(\boldsymbol{\mathcal{C}}_u \mid \boldsymbol{\mathcal{A}}) \Big(\prod_{\underline{i}=1}^{N_u} \sum_{\mathbf{z}_{\underline{i}}} P(\mathbf{z}_{\underline{i}} \mid \boldsymbol{\mathcal{C}}_u) P(w^O_{\underline{i}} \mid$

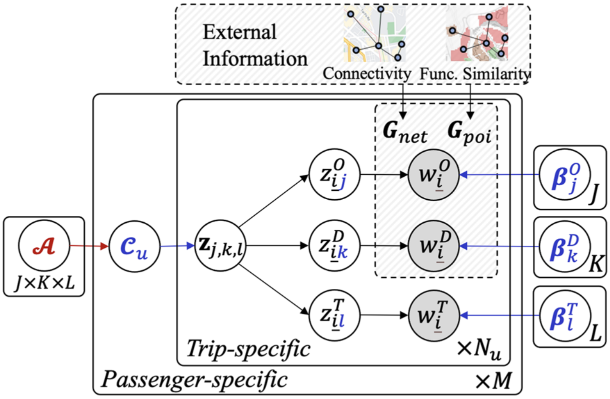


Fig. 3 Generative Process for trips from each passenger via latent topics **z**

$z_{\underline{i}}^{O}, \mathbf{B}^{O})P(w_{\underline{i}}^{D} \mid z_{\underline{i}}^{D}, \mathbf{B}^{D})P(w_{\underline{i}}^{T} \mid z_{\underline{i}}^{T}, \mathbf{B}^{T})\Big) d\mathcal{C}_u$. However, the quantity $P(\mathbf{d}_u \mid \boldsymbol{\mathcal{A}}, \mathbf{B}^{O}, \mathbf{B}^{D}, \mathbf{B}^{T})$ cannot be computed tractably due to the coupling between $\boldsymbol{\mathcal{A}}$ and $\mathbf{B}^{O}, \mathbf{B}^{D}, \mathbf{B}^{T}$ in the summation over latent topics Blei et al. (2003). Luckily, variational inference provides us with a tractable lower bound on the log likelihood, which will be elaborated in detail in Section 4.

After combining the external knowledge, the model parameters are learned by maximizing the regularized likelihood function, with λ tuning the penalty strength:

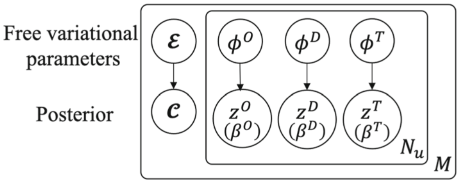
$$\max_{\boldsymbol{\mathcal{A}}, \mathbf{B}^{O}, \mathbf{B}^{D}, \mathbf{B}^{T}} \lambda L(\boldsymbol{\mathcal{A}}, \mathbf{B}^{O}, \mathbf{B}^{D}, \mathbf{B}^{T}) - (1-\lambda) R(\mathbf{G}^{O,D}). \tag{9}$$

4 Parameter estimation

In this section, we describe a tensorized variational EM-algorithm to optimize the model parameters $\boldsymbol{\mathcal{A}}, \mathbf{B}^{O}, \mathbf{B}^{D}, \mathbf{B}^{T}$ in Eq. (9), efficiently. Based on the existing variational EM-algorithm (Blei et al. 2003), we mainly emphasize the following significant contributions in our algorithm: (1) For E-step in Section 4.1, the variational E-step is extended from one-dimension words to high-dimension words with tucker decomposition; (2) For M-step in Section 4.2, the gradient ascend method is adopted to address the graph regularizations; (3) Most importantly, in Section 4.3, an online learning algorithm is proposed to handle the big data problem in smart transportation systems. The algorithm is summarized as follows:

- Tensorized variational E-step: to approximate posterior, four variational distributions $q(\cdot)$s are introduced for $\mathcal{C}, z^{O}, z^{D}, z^{T}$ with free variational parameters $\boldsymbol{\mathcal{E}}, \boldsymbol{\Phi}^{O}, \boldsymbol{\Phi}^{D}, \boldsymbol{\Phi}^{T}$ respectively, as shown in Fig. 4; then the lower bound (LB) for the original log-likelihood is calculated by Jensen's inequality, with optimal variational parameters learned to maximize the LB;
- M-step: $\boldsymbol{\mathcal{A}}, \mathbf{B}^{O}, \mathbf{B}^{D}, \mathbf{B}^{T}$ are estimated to maximize the tightest LB learned from E-step.

Fig. 4 Variational distribution to approximate posterior



4.1 Tensorized variational E-Step

The variational distribution is formulated as Eq. (10) to approximate the posterior distribution of each passenger:

$$\begin{aligned} Q &= q\Big(\mathcal{C}, (z^O, z^D, z^T) \mid \mathcal{E}, (\boldsymbol{\Phi}^O, \boldsymbol{\Phi}^D, \boldsymbol{\Phi}^T)\Big) \\ &= q(\mathcal{C} \mid \mathcal{E}) \prod_{i=1}^{N_u} q(z_{\underline{i}}^O \mid \boldsymbol{\phi}_{\underline{i}}^O) q(z_{\underline{i}}^D \mid \boldsymbol{\phi}_{\underline{i}}^D) q(z_{\underline{i}}^T \mid \boldsymbol{\phi}_{\underline{i}}^T), \end{aligned} \tag{10}$$

where $\boldsymbol{\phi}_{\underline{i}}^O \in \mathbb{R}^J$, with $\phi_{\underline{i}j}^O$ interpreting the probability that word at i-th position in current document is generated from origin topic j.

A tight lower-bound is found by minimizing Kullback-Leibler (KL) divergence between the inference distribution Q and posterior P:

$$\min_{\mathcal{E}, \boldsymbol{\Phi}^O, \boldsymbol{\Phi}^D, \boldsymbol{\Phi}^T} KL\Big[Q \parallel P\Big(\mathcal{C}, z^O, z^D, z^T \mid \mathbf{d}_u, \boldsymbol{\mathcal{A}}, \mathbf{B}^O, \mathbf{B}^D, \mathbf{B}^T\Big)\Big]. \tag{11}$$

As shown in Appendix A.2, the optimal variational parameters $\mathcal{E}^*$, $(\boldsymbol{\Phi}^{O*}, \boldsymbol{\Phi}^{D*}, \boldsymbol{\Phi}^{T*})$ are learned by computing the derivatives of the KL divergence and setting them to zero, with results shown in Eqs. (12) and (13).

To estimate $\phi_{\underline{i}j}^O$ for the u-th passenger by Appendix A.2.1:

$$\phi_{u,\underline{i}j}^O \propto \beta_{jo}^O \exp\Big[\sum_{k=1}^K \sum_{l=1}^L \phi_{u,\underline{i}k}^D \phi_{u,\underline{i}l}^T \Big(\Psi(\epsilon_{u,j,k,l}) - \Psi(\sum_{jkl} \epsilon_{u,j,k,l})\Big)\Big]. \tag{12}$$

The parameter of one dimension, for example, $\phi_{\underline{i}j}^O$, is not only related to its own dimension but also other dimensions $\phi_{\underline{i}k}^D$ and $\phi_{\underline{i}l}^T$.

Therefore, we could perform the block coordinate descent algorithm, which iteratively update the parameters for ODT dimensions until convergence.

To estimate $\epsilon_{j,k,l}$ for the u-th passenger via Appendix A.2.2:

$$\epsilon_{u,j,k,l} = \alpha_{j,k,l} + \sum_{i=1}^{N_u} \phi_{u,\underline{i}j}^O \phi_{u,\underline{i}k}^D \phi_{u,\underline{i}l}^T. \tag{13}$$

Algorithm 1 GR-TensorLDA in batch learning: Tensorized Variational EM Algorithm

```
Input: passengers, $J, K, L, \lambda, \mu, \nu$, stopping tolerance $tol$, $max\_iterEM$, $max\_iterInfer$, $maxIter_{\beta}$.
for $iter = 1$ to $max\_iterEM$ do
    E-Step:
    Initialization: $\phi^{O}_{\underline{i}j} = \frac{1}{J}, \phi^{D}_{\underline{i}k} = \frac{1}{K}, \phi^{T}_{\underline{i}l} = \frac{1}{L}$ and $\epsilon_{u,j,k,l} = \alpha_{u,j,k,l} + \frac{N_u}{J \times K \times L}$ for all $u, \underline{i}, j, k, l$.
    for $iter = 1$ to $max\_iterInfer$ do
        for $u = 1$ to $M$ do
            for $\underline{i} = 1$ to $N_u$ do
                Update $\phi^{O}_{u,\underline{i}j}, \phi^{D}_{u,\underline{i}k}, \phi^{T}_{u,\underline{i}l}$ by Eq. (12) $\forall j, k, l$; Normalize
                Update $\epsilon_{u,j,k,l}$ by Eq. (13) $\forall (j, k, l)$.
            end for
        end for
        if $parameter_{i+1} - parameter_{i} < tol$, break
    end for
    M-Step:
    Update $\boldsymbol{\beta}^{O}_{j}, \boldsymbol{\beta}^{D}_{k}, \boldsymbol{\beta}^{T}_{l}$ by Eqs. (14), (15) until convergence
    Update $\alpha_{j,k,l}$ by Eq. (16) until convergence
end for
Output: $\boldsymbol{\mathcal{A}}, \mathbf{B}^{O}, \mathbf{B}^{D}, \mathbf{B}^{T}$.
```

4.2 M-Step

In the M-step, we aim to maximize the lower bound learned from E-step with respect to $\mathbf{B}^{O}$, $\mathbf{B}^{D}$, $\mathbf{B}^{T}$ and $\boldsymbol{\mathcal{A}}$.

As shown in Appendix A.3.1, $\mathbf{B}^{O}$ and $\mathbf{B}^{D}$ cannot be solved in the closed-form solution due to the graph regularization. Therefore, we propose to use the gradient ascend method to update $\mathbf{B}^{O}$ and $\mathbf{B}^{D}$:

$$\boldsymbol{\beta}^{O,\tau+1}_{j} = \boldsymbol{\beta}^{O,\tau}_{j} + r\nabla L(\boldsymbol{\beta}^{O}_{j}). \tag{14}$$

the gradient with respect to β^{O}_{jo} is $\nabla L(\boldsymbol{\beta}^{O}_{j}) = \lambda \frac{1}{\beta^{O}_{jo_1}} \sum_{u=1}^{M} \sum_{\underline{i}=1}^{N_u} \phi^{O}_{u,\underline{i}j} \mathbf{1}(w^{O}_{u\underline{i}} = o_1) - (1-\lambda) \sum_{o_2} (\mu \kappa^{G_{net}}_{o_1 o_2} + (1-\mu) \kappa^{G_{poi}}_{o_1 o_2})(\beta^{O}_{jo_1} - \beta^{O}_{jo_2}) + a^{O}_{j}$.

$\mathbf{B}^{T}$ has a closed form solution as shown in Appendix A.3.2:

$$\beta^{T}_{l,t} = \sum_{u=1}^{M} \sum_{i=1}^{N_u} \phi^{T}_{u,\underline{i},l} \mathbf{1}(w^{T}_{u\underline{i}} = t). \tag{15}$$

Finally, similar to the original LDA model (Blei et al. 2003), $\boldsymbol{\mathcal{A}}$ can be estimated using the Newton-Raphson method:

$$\alpha^{s+1}_{j,k,l} = \alpha^{s}_{j,k,l} - \left[H^{-1}(\boldsymbol{\mathcal{A}}) g(\boldsymbol{\mathcal{A}}) \right]_{j,k,l}. \tag{16}$$

Its detailed derivation is given in Appendix A.3.3.

It is worth mentioning that Eqs. (14), (15), and (16) do not show a direct relation between J, K, L and model parameters: this is because J, K, L affect the variational

Algorithm 2 Online GR-TensorLDA: Tensorized Variational EM Algorithm

```
Input: passengers, $J, K, L, \lambda, \mu, \nu$, stopping tolerance $tol$, $max\_iterInfer$, $maxIter_{\beta}, r, \kappa, \tau_0$.
1:  for $s = 0$ to $\infty$ do
2:      E-Step:
3:      Initialization: $\phi^O_{\underline{i}j} = \frac{1}{J}, \phi^D_{\underline{i}k} = \frac{1}{K}, \phi^T_{\underline{i}l} = \frac{1}{L}$ and $\epsilon_{s,j,k,l} = \alpha_{s,j,k,l} + \frac{N_s}{J \times K \times L}$ for all $s, \underline{i}, j, k, l$.
4:      for $iter = 1$ to $max\_iterInfer$ do
5:          for $\underline{i} = 1$ to $N_s$ do
6:              Update $\phi^O_{s,\underline{i}j}, \phi^D_{s,\underline{i}k}, \phi^T_{s,\underline{i}l}$ by Eq.(12) $\forall j, k, l$; Normalize
7:              Update $\epsilon_{s,j,k,l}$ by Eq.(13) $\forall (j, k, l)$.
8:          end for
9:          if $parameter_{i+1} - parameter_i < tol$, break
10:     end for
11:     M-Step:
12:     Update $\tilde{\mathbf{B}}^O, \tilde{\mathbf{B}}^D$ by $\tilde{\boldsymbol{\beta}}^O_j \leftarrow \tilde{\boldsymbol{\beta}}^O_j + r \nabla L_s(\tilde{\boldsymbol{\beta}}^O_j)$ until convergence
13:     Update $\tilde{\mathbf{B}}^T$ by $\tilde{\beta}^T_{l,t} = M \sum_{i=1}^{N_s} \phi^T_{s,\underline{i}l} \mathbf{1}(w^T_{s\underline{i}} = t)$
14:     Set $\mathbf{B}^{O,D,T} = (1 - \rho_s)\mathbf{B}^{O,D,T} + \rho_s \tilde{\mathbf{B}}^{O,D,T}$
15:     Update $\alpha_{j,k,l}$ by Eq.(16) until convergence
16: end for
Output: $\boldsymbol{\mathcal{A}}, \mathbf{B}^O, \mathbf{B}^D, \mathbf{B}^T$.
```

variables and then the variational variables affect model parameters. Besides, in the parameter initialization, since we do not have a prior knowledge about the distributions, equalized and uniform parameters are initialized.

The whole algorithm is shown as Algorithm 1: it is in a batch learning manner which needs to read through the whole document set for each iteration.

4.3 Online learning algorithm

In practice, the proposed Algorithm 1 is very computationally intense since it updates parameters in a batch learning manner, which iterates between analyzing each observation and updating dataset-wide variational parameters. Therefore, as shown in Line 5 in Algorithm 1, E-step needs a full pass through the entire corpus each iteration, which is impractical when dealing with a large dataset containing tens of thousands of passengers. For example, Hong Kong, as an international transport hub, the monthly visitor arrivals were recorded to 6 million in Dec-2018, and the batch learning algorithm is not suitable for the situation where new visitors are continually arriving (Hoffman et al. 2010).

To this end, to efficiently implement the proposed model in real traffic systems, we further develop an online stochastic algorithm, which outputs good estimates outstandingly faster than the batch algorithm. To avoid repetitious details, only the critical differences in the algorithm will be explained.

In E-step, the updating equations from Eq. (12) to Eq. (13) remain the same except that variational variables are updated each time a passenger s is read, as shown in Line 6–7 in Algorithm 2.

In M-step, to update model parameters stochastically, once observing the current passenger s, we first assume the optimal model parameters are learned if the entire corpus contained this passenger s repeated M times: under this setting (denoted with the accent symbol $\tilde{\beta}$), $\tilde{\mathbf{B}}^O$ and $\tilde{\mathbf{B}}^D$ are updated by stochastic gradient ascend, with the gradient calculated based on the single observation $\mathbf{d}_s$ times M:

$$\tilde{\boldsymbol{\beta}}_j^O = \tilde{\boldsymbol{\beta}}_j^O + r\nabla L_s(\tilde{\boldsymbol{\beta}}_j^O). \tag{17}$$

where gradient $\nabla L_s(\tilde{\boldsymbol{\beta}}_j^O)$ with respect to $\tilde{\beta}_{jo}^O$ is $\nabla L_s(\tilde{\boldsymbol{\beta}}_j^O) = \lambda \frac{M}{\tilde{\beta}_{jo_1}^O} \sum_{\underline{i}=1}^{N_s} \phi_{s,\underline{i}j}^O \mathbf{1}(w_{s\underline{i}}^O = o_1) - (1-\lambda) \sum_{o_2} (\mu\kappa_{o_1o_2}^{G_{net}} + (1-\mu)\kappa_{o_1o_2}^{G_{poi}})(\tilde{\beta}_{jo_1}^O - \tilde{\beta}_{jo_2}^O) + a_j^O$, with details in Appendix A.3.1.

Similar with Eq. (15), $\tilde{\mathbf{B}}^T$ has closed-form solution with passenger s repeated M times:

$$\tilde{\beta}_{l,t}^T = M \sum_{i=1}^{N_s} \phi_{s,\underline{i}l}^T \mathbf{1}(w_{s\underline{i}}^T = t). \tag{18}$$

Then the final model parameters $\mathbf{B}^{O,D,T}$ are updated by using a weighted average of its previous value and $\tilde{\mathbf{B}}^{O,D,T}$:

$$\mathbf{B}^{O,D,T} = (1-\rho_s)\mathbf{B}^{O,D,T} + \rho_s \tilde{\mathbf{B}}^{O,D,T}. \tag{19}$$

where $\rho_s = (\tau_0 + s)^{-\kappa}$. $\kappa \in (0.5, 1]$ controls the rate at which old values of $\tilde{\mathbf{B}}$ are forgotten and guarantees convergence; $\tau_0 \leq 0$ slows down the early iterations.

The proposed online learning algorithm of the GR-TensorLDA method is summarized in Algorithm 2.

The computational complexity of each iteration in the batch learning algorithm is $\mathcal{O}(MN(J+K+L))$ ($M \ggg N, J, K, L$), whereas the complexity of each iteration in the online learning algorithm reduces to $\mathcal{O}(N(J+K+L))$ (Teh et al. 2007).

Mini-Batches: To reduce noise, parameters could be updated with a mini-batch containing multiple observations, with mini-batch size as S. $\tilde{\mathbf{B}}^T$ is updated as $\tilde{\beta}_{l,t}^T = \frac{M}{S} \sum_{s=1}^{S} \sum_{i=1}^{N_s} \phi_{s,\underline{i}l}^T \mathbf{1}(w_{s\underline{i}}^T = t)$. $\tilde{\mathbf{B}}^{O,D}$ are updated by stochastic gradient ascend with mini-batches.

5 Experiments

5.1 Dataset

The individual travel data from 1st-Jan-2017 to 31st-Mar-2017 are chosen for analysis. Each trip has recorded the anonymized passenger ID, entry station, exit station, entry time, and exit time. In this implementation, entry station, exit station, and hour stamp of entry time have been collected for each trip and aggregated over the whole three months to ensure each passenger has enough trips for analysis, with average amount of trips around 134. The Hong Kong MTR system has 98 stations in total and operates

Table 2 Data information

Data	Dimension and Description
Passenger (Training)	$M = 50,000$
Passenger (Validation, Test)	1000
Average Length	$\bar{N} \approx 134$ (trips)
Origin, Destination	$V^O, V^D = 98$ (stations)
Time (Entry)	$V^T = 24$ (hours)

in 24 hours. Thus the vocabulary size for origin, destination, and time is 98, 98, 24. We randomly pick 50,000 passengers as the training set, 1000 passengers as the validation set for tuning parameter selection, and another 1000 passengers as the test set. The data information is summarized in Table 2.

5.2 Graph definition

As discussed in Section 3.4, the geographical graph and the functional similarity graph affect passengers' travel patterns. Here we would like to define the two graphs in detail.

Geographical graph: Two spatially close stations are more likely to be in the same topic. The distance from station i to j in the graph $\{G_{net}\}_{i,j}$ is usually simplified as an "H-hop" binary graph: if from station i to j less than H-hops are needed, two stations are connected, and the edge between them is '1'; Otherwise, the edge is '0'. We set $H = 3$ since a survey stated that passengers are willing to travel freely if two stations are only three hops away (Geng et al. 2019; Li et al. 2021).

$$\{\mathbf{G}_{net}\}_{i,j} = \begin{cases} 1 & \text{hop distance}_{i,j} <= H \\ 0 & \text{hop distance}_{i,j} > H \end{cases}$$

Functional similarity graph: Two stations located in highly similar functional areas are also prone to be in the same topic. A functional similarity graph is commonly formulated with the point of information (POI) (Li et al. 2020; Zhong et al. 2017). We collect each station's surrounding POIs[1] with the following seven services: hotel, leisure shopping, major building, public facilities, residential, school, and public transport. Each element of the POI vector indicates the amount of the corresponding service nearby the station. The element $\{G_{poi}\}_{i,j}$ can be defined as cosine similarity between POI vectors of station i and station j, and a higher value in $\mathbf{G}_{poi}$ means a higher functional similarity between two stations:

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

[1] This information is available at "Location Map, MTR" showing leading hotels, shopping centres and major buildings of each stations.

5.3 Benchmark methods

We apply the following benchmark methods to passenger travel data and compare the results with the proposed model. However, a relatively limited amount of research targets ODT-P multi-clustering based on individual passenger travel data.

- **One-dimensional LDA** (1d-LDA): It defines the generative process from document to topic, topic to word in a single dimension (Blei et al. 2003). To apply it, three-dimensional data $(w^O, w^D, w^T) = (o, d, t)$ are flattened into one-dimensional $w = odt$: each different element creates a new word; thus, the total vocabulary size for the new data format is expanded to $98 \times 98 \times 24 \sim 10^5$, and the computational complexity of each iteration is $\mathcal{O}(MNK)$.
- **Tucker Decomposition** (Tucker): It decomposes an ODT flow tensor into a core tensor and mode matrices along each dimension. Each rank vector from a mode matrix can be regarded as a cluster (Sun and Axhausen 2016). However, this method is not applicable to individual travel data due to extreme sparsity. Merely to examine its ODT clustering performance, we feed it with macro-level passenger flow data with the dimension of origin, destination, and time (Sun and Axhausen 2016).
- **CP Decomposition with Graphs** (CP-G): Similar to tucker decomposition, the input is passenger flow data to check its ODT clustering performance. It decomposes an ODT flow tensor into a weight vector and mode matrices along each dimension, with graphs on OD (Li et al. 2020).
- **Three-dimensional LDA with Gibbs sampling** (3d-LDA(*Gibbs*)): It also defines a generative process for each dimension, however, without any semantic graph structure. Parameters are estimated by Gibbs sampling (Cheng et al. 2020), with the computational complexity of each iteration as $\mathcal{O}(MN(J + K + L))$ (Porteous et al. 2008; Xiao and Stibor 2010).

All the methods are compared by whether it is an individualized analysis (*Indiv.* for short), tensor-based (*Tensor*), and graph-structured (*Graph*) model with efficient online algorithm (*Eff.*) and low computational complexity (*Complexity*) as shown in Table 3. Only our model ticks all the boxes.

Table 3 Comparison of benchmark methods

Methods	*Indiv.*	*Tensor*	*Graph*	*Eff.*	*Complexity* [2]
1d-LDA	✓	×	×	×	$\mathcal{O}(MNK)$
Tucker	×[1]	✓	×	-	-
CP-G	×[1]	✓	✓	-	-
3d-LDA(*Gibbs*)	✓	✓	×	×	$\mathcal{O}(MN(J+K+L))$
GR-TensorLDA (online)	✓	✓	✓	✓	$\mathcal{O}(N(J+K+L))$

[1] Not individualized analysis, input with passenger flow data;
[2] Computational complexity of each iteration, and $M \ggg N, J, K, L$

5.4 Evaluation metrics

Traditional topic models only measure the quality of the model via perplexity, but ignore how “interpretable” the learned topics are. For example, a topic containing all words related to covid (*e.g.*, *omicron*, *vaccine*, *quarantine*, *etc*: these words are all connected in a knowledge graph) is more “interpretable” than a topic containing words from various themes (*e.g.*, *covid*, *solar energy*, *iPhone*: these words are far away to each other in a knowledge graph). Such “interpretability” is usually measured by point-wise mutual information (PMI), known as topic coherence. Besides, given our graph structure on origin and destination dimensions, we also innovatively design distance of graph to measure the “interpretability”: a topic with words that are close to each other on a graph is more interpretable than the one that does not.

Topic Coherence PMI: PMI is to evaluate how meaningful the learned topics along each dimension are (Yao et al. 2017; Newman et al. 2010). For example, topic j in dimension of the origin stations is calculated as $PMI(\boldsymbol{\beta}_j^O) = \sum_{o_1,o_2 \in N_j^o, o_1 \neq o_2} \frac{P(w_{o_1}^O, w_{o_2}^O)}{P(w_{o_1}^O)P(w_{o_2}^O)}$, where N_j^o is the top N words in origin topic j, and we choose the top 10 words. $P(w_o^O)$ is the probability that word $w^O = o$ is observed in a passenger, and $P(w_{o_1}^O, w_{o_2}^O)$ captures the probability that $w^O = o_1$ and $w^O = o_2$ co-occur in the same passenger. A higher PMI value means a more coherent topic.

Distance on Graph (d_G): Based on the Laplacian matrix of a graph, d_G measures the distance of the word components from a topic. A smaller value means a more concentrated topic. For example, the distance on graph for origin topic j is defined as $d_{G_{net}} = (\boldsymbol{\beta}_j^O)^T \mathbf{L}_{net} \boldsymbol{\beta}_j^O$, $d_{G_{poi}} = (\boldsymbol{\beta}_j^O)^T \mathbf{L}_{poi} \boldsymbol{\beta}_j^O$.

Perplexity: Perplexity (Blei et al. 2003) examines the likelihood of the proposed model in the test set. A lower perplexity means a higher likelihood.

5.5 Parameter tuning and station topics

5.5.1 Parameter tuning

The number of topics in each ODT dimension is set as $J = 10, K = 10, L = 4$ in our dataset. This is chosen by the expert knowledge: $J, K = 10$ since the POI of each station has seven elements; $L = 4$ since there are usually at least three time-components capturing morning peak, evening peak, and midday trend. Generally, if there is no prior information about the parameters, J, K, L could be determined by a grid search to minimize perplexity (Blei et al. 2003) or maximize topic coherence (Yao et al. 2017). Theoretically, with bigger J, K, L, the dimensions of model parameters $\boldsymbol{\mathcal{A}}$, $\mathbf{B}^O$, $\mathbf{B}^D$, $\mathbf{B}^T$ also increase, which means more latent clusters are introduced to describe data pattern: this will naturally increase model’s likelihood and decrease perplexity. However, too large J, K, L may cause overfitting. Tuning parameters for graph regularization λ, μ, ν are searched from grids $\lambda, \mu, \nu \in \{0.1, 0.2, \ldots, 0.9\}$, and configuration parameters for online algorithm $\kappa \in \{0.5, 0.6, \ldots, 1.0\}$, $\tau_0 \in \{1, 4, 16, 64, 256, 1024\}$ and $S \in \{1, 4, 16, 64, 256, 1024\}$. The optimal values are

Table 4 Origin topic

Metric	Method	z_0^O	z_1^O	z_2^O	z_3^O	z_4^O	z_5^O	z_6^O	z_7^O	z_8^O	z_9^O
PMI	Tucker	−33.29	−43.53	−177.27	−49.02	−102.80	−106.24	−39.06	−26.31	−39.44	−24.69
	CP-G	−34.59	−43.05	−73.91	−36.57	−38.17	−64.46	−35.81	−25.25	−47.15	−40.66
	3d-LDA(*Gibbs*)	3.21	2.86	9.31	26.77	13.40	12.09	24.02	−18.94	3.95	33.80
	GR-TensorLDA	3.29	3.0	24.31	19.21	28.95	26.41	57.86	−19.42	14.12	34.38
$d_{G_{poi}}$	Tucker	2.27	13.14	8.02	3.72	3.28	4.28	2.85	19.54	1.66	2.53
	CP-G	2.20	2.88	2.48	2.85	2.21	3.03	2.56	2.59	2.26	2.20
	3d-LDA(*Gibbs*)	2.71	2.67	3.50	2.27	4.32	1.79	2.94	2.50	3.82	4.54
	GR-TensorLDA	2.92	1.32	1.42	1.42	1.46	0.91	1.42	0.89	2.25	2.43
$d_{G_{net}}$	Tucker	0.46	1.82	1.91	0.99	1.01	0.99	0.72	5.4	0.50	0.67
	CP-G	0.38	0.49	0.40	0.48	0.36	0.47	0.43	0.44	0.42	0.37
	3d-LDA(*Gibbs*)	0.63	0.57	0.73	0.55	0.99	0.48	0.64	0.55	0.75	1.12
	GR-TensorLDA	0.65	0.35	0.34	0.43	0.29	0.25	0.39	0.20	0.44	0.53

chosen to maximize likelihood in the validation set, with $\lambda = 0.5$, $\mu = 0.4$, $\nu = 0.2$ and $S = 128$, $\kappa = 0.5$, $\tau_0 = 256$.

5.5.2 Topic matching

It is worth mentioning that before the comparison, the topics learned from different methods need to be matched first. We match two topics from two methods if they have the highest cosine similarity. For example, topic i from method m_1 refers to topic $\hat{j}$ from method m_2 when $\hat{j} = \arg max_j\{\text{cos-similarity}(\boldsymbol{\beta}_i^{m_1}, \boldsymbol{\beta}_j^{m_2})\}$.

5.5.3 PMI and distance on graph

The PMI and d_G for the learned origin topics are shown in Table 4, with the best performance highlighted in boldface and the second-best performance highlighted in underline.

From Table 4, we find that: (1) Tucker and CP decomposition have the worst PMI since they are methods targeting macro-level traffic analysis. Thus it ignores the individual passenger information; However, tensor decomposition with graphs considered (i.e., CP-G) still has higher PMI and lower d_G than that without graphs; (2) Our proposed method achieves twice higher PMI than 3d-LDA(*Gibbs*) for most topics, which means the proposed model can discover more meaningful topics. This is due to the external information introduced as graphs, with more than 50% lower d_G observed on both graphs.

5.5.4 Perplexity vs interpretability

In Table 5, our model has a 10% higher perplexity in the test set, which means a lower likelihood score for these passengers. When we introduce the regularization

Table 5 Trade-off between perplexity and intepretability

	Perplexity				Interpretability[1]		
Methods	O	D	T	Overall	$\bar{\text{PMI}}$	$\bar{d}_{G_{poi}}$	$\bar{d}_{G_{net}}$
1d-LDA	–	–	–	575.11	–	–	–
3d-LDA(*Gibbs*)	44.69	48.24	17.48	110.41	11.05	3.11	0.71
GR-TensorLDA	50.61	53.33	17.31	121.25	19.21	1.64	0.38

[1] Under the interpretability, $\bar{\text{PMI}}$, $\bar{d}_{G_{poi}}$, $\bar{d}_{G_{net}}$ are the average of PMI, $d_{G_{poi}}$, $d_{G_{net}}$ over the 10 origin topics in Table 4

term into the loss function, it naturally decreases the likelihood score because regularization terms generally reduce the model fitting accuracy (*i.e.*, likelihood score) in the exchange of a better generalizability on the testing samples.

However, in the literature, it has been shown that perplexity is not a good measure compared to the topic coherence PMI score. This is because the perplexity itself does not reflect the meaningfulness of topics, and topics with lower perplexity might even conflict with the real-world knowledge (Yao et al. 2017; Chang et al. 2009). Our experiments show that, by adding the graph regularization, although the model likelihood score (i.e., 10% higher perplexity measure) is lower, the model interpretability and generalization power (i.e., as seen in the twice higher PMI and 50% lower d_G measures) are significantly improved. Therefore, we observed such a trade-off between perplexity and interpretability.

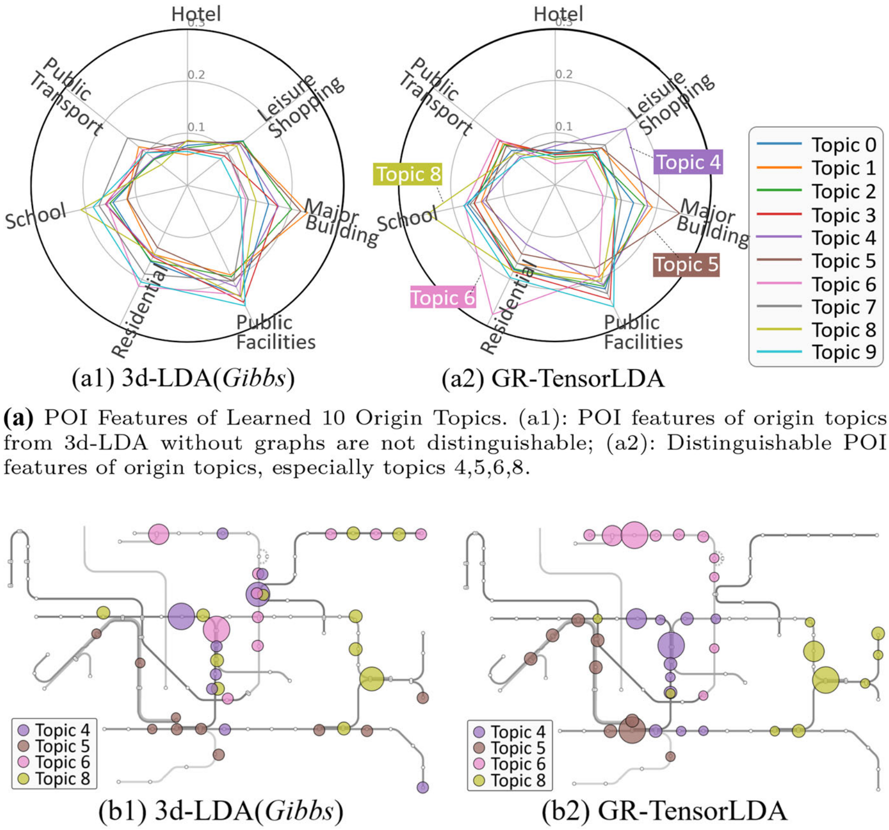
1d-LDA has the highest perplexity since it is not a tensor-based model, thus cannot preserve the innate spatiotemporal correlations.

5.6 Improved interpretability in station topics

To better demonstrate the enhanced interpretability of the proposed model's topics and check how they reflect the real world, we also visualize: (1) the topic POI features, (2) topic locations, and (3) topic station components based on the real metro map, in comparison to the second-best model 3d-LDA (*Gibbs*) in origin dimension only. The discovered topics can be used as station clusters.

(1) Topic POI Feature to check the POI feature of each topic: Usually, more distinguishable topics are better (Zhu et al. 2012). The POI features of topics from 3d-LDA(*Gibbs*) and our model are calculated as $\mathbf{B}^{O}_{(poi)} = \mathbf{B}^{O}\boldsymbol{G}_{poi}$, where $\mathbf{B}^{O}_{(poi)} \in \mathbb{R}^{J\times N_{poi}}$ and the j-th row $\boldsymbol{\beta}^{O}_{j(poi)} \in \mathbb{R}^{N_{poi}}$ indicates the POI feature of this topic. As shown in Fig. 5. (a2), topics from the proposed model capture more distinct POI groups: origin topic 4, 5, 6, and 8 capture the POI leisure shopping, major building, residential and school respectively; Topics from 3d-LDA(*Gibbs*) instead, as shown in Fig. 5. (a1), capture the topics with similar and non-distinguishable POI distribution. The distinct POI patterns are observed in our destination topics too.

(2) Topic Location on Map to check each topic's location on map: The top 10 stations with the highest weights from origin topics 4, 5, 6, and 8 are located on the metro map. In Fig. 5. (b), the stations from our topics are concentrated in the same



(a) POI Features of Learned 10 Origin Topics. (a1): POI features of origin topics from 3d-LDA without graphs are not distinguishable; (a2): Distinguishable POI features of origin topics, especially topics 4,5,6,8.

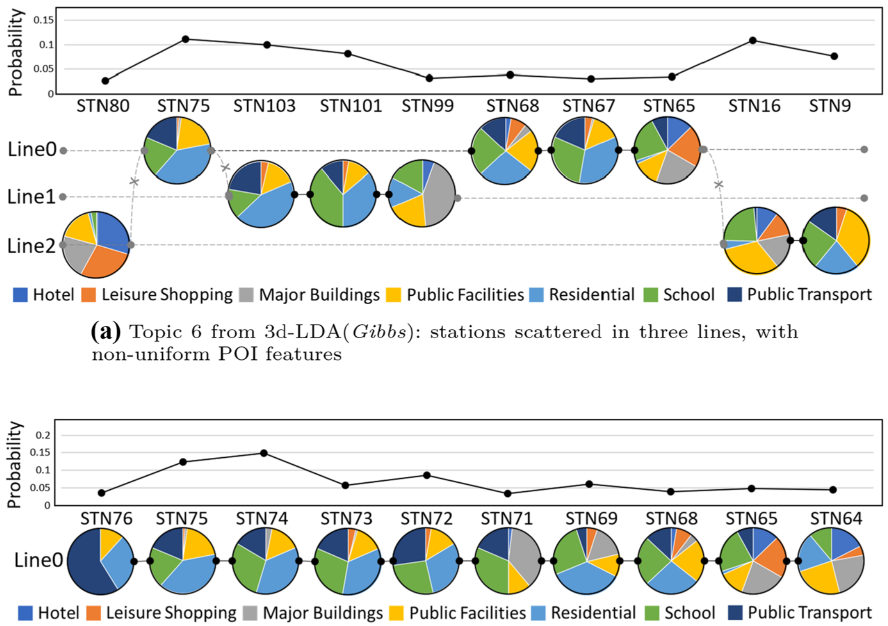
(b) Locations of Selected Four Topics. (b1): Topics from 3d-LDA without graphs scattered in different regions; (b2): Topics from GR-TensorLDA concentrated in same region/line. (Bigger bubble means higher weight).

Fig. 5 **(a)** POI features for origin topics; **(b)** The locations of top 10 stations from origin topics 4,5,6 and 8

line/region; however, topics without graphs are dispersed among different regions. Therefore, the topics learned from our proposed method have significant improvement in interpretability and reflect external knowledge.

(3) Station Components Analysis to check each topic's station component in terms of the station location and POI: A further detailed study about the top 10 station components with the highest weights inside each topic (here topic 6 is chosen) is conducted to check those stations' exact locations and the surrounding POIs. (1) In Fig. 6. (b), the top 10 stations of our topic 6 are all located in the same metro line, and the POI feature of each station is also mainly residential; (2) On the contrary, in Fig. 6. (a) the top 10 stations of topic 6 without graphs are scattered over different three lines, and those stations also have quite different POI features, such as station 80 in leisure shopping, station 99 in the major building.

To conclude, the identified topics from the proposed model have better physical meanings and interpretability.



(a) Topic 6 from 3d-LDA(*Gibbs*): stations scattered in three lines, with non-uniform POI features

(b) Topic 6 from GR-TensorLDA: all stations in the same line 0, with dominant POI as 'Residential' (light blue in pie chart)

Fig. 6 The top 10 stations from topic 6, with station weights presented as line charts on top, real metro lines plotted below and station POI features presented as pie charts

Applications of OD Clustering: Metro companies usually categorize stations by their expert knowledge, such that different station categories have different operational, marketing, and urban planning strategies. However, this categorization is usually out-of-date since a station's feature evolves over time. The learned station clustering is purely data-driven and updated with data, which provides better insights.

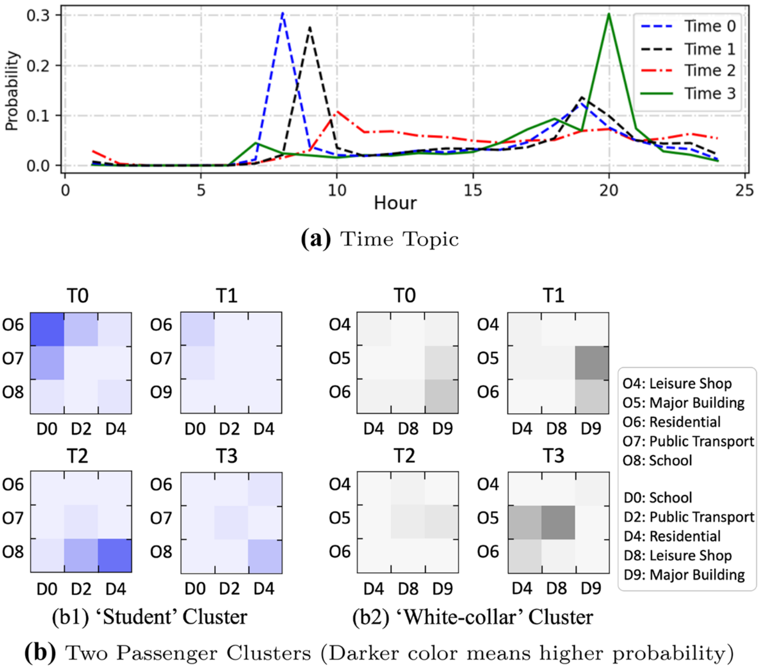
5.7 Time topics

The time topics from our method are shown in Fig. 7(a), Topic 0 captures the first morning peak (7–8 AM); Topic 1 captures the second morning peak (9–10 AM); Topic 2 captures the mid-day trend, and Topic 3 captures the evening peak (8 PM).

Applications of Time Clustering: The learned time topics could offer clear insights about the peak hours and enable crowd management.

5.8 Passenger clustering

The passenger cluster could be learned from each passenger's tensor topic distribution parameter $\mathcal{C}_u$. The distance between two passengers' topic distributions could be measured by the Euclidean distance, the Jensen Shannon (JS) divergence



(a) Time Topic

(b1) 'Student' Cluster (b2) 'White-collar' Cluster

(b) Two Passenger Clusters (Darker color means higher probability)

Fig. 7 **(a)** Time topics, **(b)** Passenger clusters

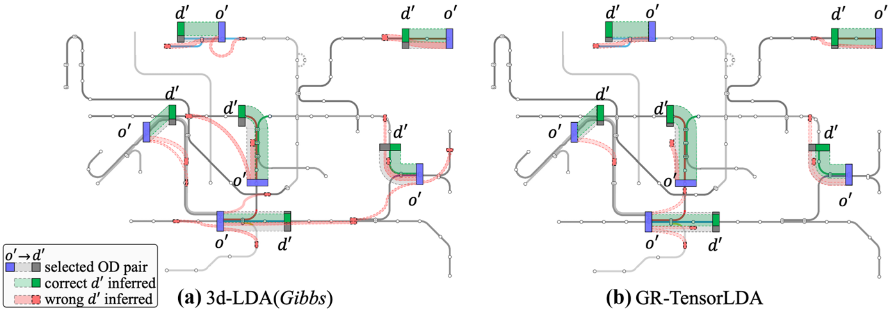
(Cheng et al. 2020) and so on. We choose the JS divergence since it is a symmetric measure for distributions. Then clustering methods such as K-means could be applied to cluster passengers.

Two passenger clusters are shown: In Fig. 7(b1) 'student' cluster travels from O6 (residential) to D0 (school) at T0 (7–8 AM) and travels back from O8 (school) to D4 (residential) at T2 (mid-day); In Fig. 7(b2) 'white-collar' cluster usually travels from O5 (major building) to D9 (major building) at T1 (9–10 AM) and after work travels from O5 (major building) to D8 (leisure shopping) at T3 (8 PM).

Applications of passenger clustering: (1) **Customized Services**: Passenger clustering could help public transport companies better understand passenger demographics, enabling customized travel reward plans or tailored advertisements for different passenger clusters. (2) **Destination Inference**: Moreover, the conditional probability based on the latent parameters $\mathbf{B}^O$, $\mathbf{B}^D$, $\mathbf{B}^T$, and $\mathcal{C}_u$ enables more potential applications. For example, given the passenger u', origin o', and entry time t', the destination could be predicted by: $P(w^D = d \mid w^O = o', w^T = t', u') \propto P(w^O = o', w^D = d, w^T = t', u') = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{l=1}^{L} c_{u',j,k,l} \times \beta^O_{jo'} \times \beta^D_{kd} \times \beta^T_{lt'}$. As a result, a destination crowd warning message in the mobile application could then be directed towards each passenger. With $\mathbf{B}^O$, $\mathbf{B}^D$ better estimated with graphs, for passengers who travel between two stations (accounts for 83.8% of the population in our dataset), the destination inference accuracy is improved by 18% compared with 3d-LDA(*Gibbs*), as shown in Table 6. In Fig. 8, the most popular OD pairs (i.e., $o' \rightarrow d'$) are selected to visualize destination inference. With the input of passenger u', who has

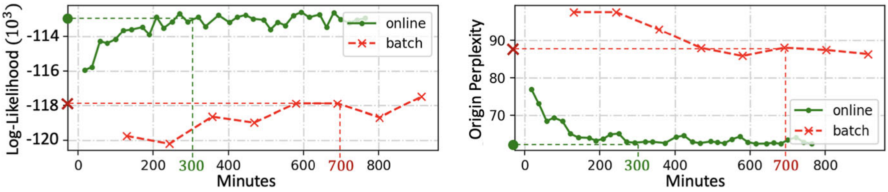
Table 6 Destination inference accuracy

Route Type (Percent in data)	3d-LDA(*Gibbs*)	GR-TensorLDA
A $\rightleftharpoons$ B (83.8%)	65.73%	77.21%
A $\rightleftharpoons$ C $\rightleftharpoons$ B(10.5%)	59.11%	66.19%
$\geq$ 4 stations (5.7%)	50.21%	51.48%



(a) 3d-LDA(*Gibbs*) **(b)** GR-TensorLDA

Fig. 8 Destination inference for selected OD pairs: o' in blue, ground-truth d' in grey, correct d in green, wrong d in red



(a) Log-Likelihood Evolution **(b)** Origin Perplexity Evolution

Fig. 9 Convergence comparison of **(a)** log-likelihood evolution and **(b)** origin perplexity evolution between Online Algorithm 2 in green and Batch Algorithm 1 in red. Each point marker '·' in online version denotes 10 iterations, and each cross marker '×' in batch version denotes 1 iteration

these OD pairs, and the time t' when he/she enters o', the destination is inferred by our method with higher accuracy (correct d' in green); However, the method without graphs makes more mistakes (wrong d in red).

5.9 Faster convergence from online algorithm

Last but not least, as shown in Fig. 9, we compare the convergence speed of the algorithm's batch version and its online version in the same computation environment. The proposed online stochastic algorithm needs more iterations to converge but 60% less time: The online version (shown in color green) converges at $t \approx 300$, more than twice faster than the batch algorithm (shown in color red), which converges at $t \approx 700$.

Moreover, the online algorithm also converges with with better parameter estimates: (1) Higher log-likelihood: As shown in Fig. 9. (a), online version converges at log-likelihood $\approx -113 \times 10^3$, higher than batch version's convergence at log-likelihood

$\approx -118 \times 10^3$; (2) Lower perplexity: as shown in Fig. 9. (b), online version converges at origin perplexity ≈ 60, lower than batch version's convergence at origin perplexity ≈ 89.

6 Conclusion

In this paper, we studied the ODT-P multi-clustering problem for the individual passenger travel pattern to achieve meaningful clusters on each dimension (origin, destination, and time) and on the individual passenger by incorporating external information on the origin and destination stations. To solve this challenge, we proposed a novel graph-regularized tensor Latent Dirichlet Allocation model, which applies to the travel data of each passenger with the consideration of the external information as the graph regularization. We proposed a tensorized variational EM-algorithm to estimate parameters. To improve the scalability, an online learning algorithm is further proposed. In the case study based on the Hong Kong metro system, we demonstrate our superiority over state-of-the-art methods in terms of two times higher topic coherence, 50% lower distance on graph, and better interpretability. Our improvement is also reflected on its 20% more accurate individual destination inference. The proposed online learning method can also converge twice faster with the same good performance as the batch learning method.

Future work

Our model will be extended to cover trip duration, which is a continuous variable. The generative process will be extended to handle continuous distribution correspondingly. Besides, due to the independent assumption of Dirichlet distribution, correlations between passengers will also be further examined.

Generalization

This work could also be applied to non-metro data such as bus and sharing rides if the ODT information is recorded. In the road traffic, the nodes in $\mathbf{G}_{net}$ and $\mathbf{G}_{poi}$ could be defined as different grids, road segments, or zip code zones, and the edges could be defined similarly if distance and POI are available.

Acknowledgements This work is collaborative research with Hong Kong MTR Co. under the RGC GRF 16201718 and 16216119. The authors appreciate the help from MTR research, marketing, and customer service teams. The preliminary idea of this work has been presented IEEE ICDE 2021 Ph.D. Symposium (Li et al. 2021). The authors would like to give their appreciations to the comments and advice from the symposium.

Author Contributions (1) Z. Li: ideation, model formulation, programming and conducting experiments, paper writing; (2) H. Yan and C. Zhang: ideation, model formulation, paper writing; (3) F. Tsung: ideation, supervision.

Funding Open Access funding enabled and organized by Projekt DEAL. This work is funded by the Hong Kong Research Grants Council (RGC) under the grants RGC GRF 16201718 and 16216119.

Data availibility The data will be publicly available under https://github.com/bonaldli/GR-TensorLDA.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Code availability The code is publicly accessible by https://github.com/bonaldli/GR-TensorLDA.

Ethics approval This paper satisfies the compliance with ethical standards. There is no potential conflicts of interest; The research does not involve Human Participants and/or Animals; The data in this paper has been anonymized to protect data privacy; Informed consent was obtained from all individual participants.

Consent to participate Informed consent was obtained from all individual participants included in the study.

Consent for publication All individual participants have consented to the submission of the regular paper to the journal.

Appendix A tensorized variational EM algorithm

In this appendix, the proposed Tensorized Variational EM Algorithm will be derived in detail. For simplicity, we replace $\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{L}$ with $\sum_{jkl}$ to avoid some long expression.

A.1 Computing $E_q[\log(c_{j,k,l} \mid \mathcal{A})]$

The three-dimensional Dirichlet distribution could be written as an exponential family: $P(\mathcal{C} \mid \mathcal{A}) = \exp\Big\{\sum_{jkl}(\alpha_{j,k,l} - 1)\log c_{j,k,l} + \log\Gamma(\sum_{jkl}\alpha_{j,k,l}) - \sum_{jkl}\log\Gamma(\alpha_{j,k,l})\Big\}$, where $\alpha_{j,k,l} - 1$ is the natural parameter, $\log c_{j,k,l}$ is the sufficient statistic for $c_{j,k,l}$, and $\log\Gamma(\sum_{jkl}\alpha_{j,k,l}) - \sum_{jkl}\log\Gamma(\alpha_{j,k,l})$ is the log of the normalization factor. Since the derivative of the log of the normalization factor with respect to the natural parameter is equal to the expectation of the sufficient statistic, we can get:

$$E[\log(c_{j,k,l} \mid \boldsymbol{\mathcal{A}})] = \Psi(\alpha_{j,k,l}) - \Psi\left(\sum_{jkl} \alpha_{j,k,l}\right)$$

A.2 Variational inference

The lower bound (LB) of the log likelihood of a document is obtained by Jensen's inequality:

$$\begin{aligned}
&\log P(\mathbf{d}_u \mid \boldsymbol{\mathcal{A}}, \mathbf{B}^O, \mathbf{B}^D, \mathbf{B}^T) = \log \int \sum_{z_{j,k,l}} P(\boldsymbol{\mathcal{C}}, z, \mathbf{d}_u \mid \boldsymbol{\mathcal{A}}, \mathbf{B}^O, \mathbf{B}^D, \mathbf{B}^T) d\boldsymbol{\mathcal{C}} \\
&\quad = \log \int \sum_{z_{j,k,l}} \frac{P(\boldsymbol{\mathcal{C}}, z, \mathbf{d}_u \mid \boldsymbol{\mathcal{A}}, \mathbf{B}^O, \mathbf{B}^D, \mathbf{B}^T) q(\boldsymbol{\mathcal{C}}, z)}{q(\boldsymbol{\mathcal{C}}, z)} d\boldsymbol{\mathcal{C}} \\
&\quad \geq E_q[\log P(\boldsymbol{\mathcal{C}}, z, \mathbf{d}_u \mid \boldsymbol{\mathcal{A}}, \mathbf{B}^O, \mathbf{B}^D, \mathbf{B}^T)] - E_q[\log q(\boldsymbol{\mathcal{C}}, z)] = LB
\end{aligned}$$

LB can be expanded by using the factorizations of p and q:

$$\begin{aligned}
LB &= E_q[\log P(\boldsymbol{\mathcal{C}}, z, \mathbf{d}_u \mid \boldsymbol{\mathcal{A}}, \mathbf{B}^O, \mathbf{B}^D, \mathbf{B}^T)] - E_q[\log q(\boldsymbol{\mathcal{C}}, z)] \\
&= E_q[\log P(\boldsymbol{\mathcal{C}} \mid \boldsymbol{\mathcal{A}})] + E_q[\log P(z \mid \boldsymbol{\mathcal{C}})] + E_q[\log P(\mathbf{d}_u \mid z, \mathbf{B}^O, \mathbf{B}^D, \mathbf{B}^T)] \\
&\quad - E_q[\log q(\boldsymbol{\mathcal{C}})] - E_q[\log q(z)]
\end{aligned}$$

The five terms are further expanded using Appendix A.1 result:

$$\begin{aligned}
LB = {} & \log \Gamma\left(\sum_{jkl} \alpha_{j,k,l}\right) - \sum_{jkl} \log \Gamma(\alpha_{j,k,l}) + \sum_{jkl} (\alpha_{j,k,l} - 1)\left(\Psi(\epsilon_{j,k,l}) - \Psi\left(\sum_{jkl} \epsilon_{j,k,l}\right)\right) \\
& + \sum_{\underline{i}=1}^{N_u} \sum_{jkl} \phi^O_{\underline{i}j} \phi^D_{\underline{i}k} \phi^T_{\underline{i}l} \left(\Psi(\epsilon_{j,k,l}) - \Psi\left(\sum_{jkl} \epsilon_{j,k,l}\right)\right) \\
& + \sum_{\underline{i}=1}^{N_u} \left(\sum_{j=1}^{J} \sum_{o=1}^{V^O} \phi^O_{\underline{i}j} w^O_{\underline{i}o} \log \beta^O_{jo} + \sum_{k=1}^{K} \sum_{d=1}^{V^D} \phi^D_{\underline{i}k} w^D_{\underline{i}d} \log \beta^D_{kd} + \sum_{l=1}^{L} \sum_{t=1}^{V^T} \phi^T_{\underline{i}l} w^T_{\underline{i}t} \log \beta^T_{lt}\right) \\
& - \log \Gamma\left(\sum_{jkl} \epsilon_{j,k,l}\right) + \sum_{jkl} \log \Gamma(\epsilon_{j,k,l}) - \sum_{jkl} (\epsilon_{j,k,l} - 1)\left(\Psi(\epsilon_{j,k,l}) - \Psi\left(\sum_{jkl} \epsilon_{j,k,l}\right)\right) \\
& - \sum_{\underline{i}=1}^{N_u} \left(\sum_{j=1}^{J} \phi^O_{\underline{i}j} \log \phi^O_{\underline{i}j} + \sum_{k=1}^{K} \phi^D_{\underline{i}k} \log \phi^D_{\underline{i}k} + \sum_{t=1}^{L} \phi^T_{\underline{i}t} \log \phi^T_{\underline{i}t}\right)
\end{aligned}$$

A.2.1 Variational multinomial

We keep the terms in lower bound containing $\phi^O_{\underline{i}j}$ for example:

$$\begin{aligned} LB_{[\phi^O_{\underline{i}j}]} &= \phi^O_{\underline{i}j}\sum_{k=1}^{K}\sum_{l=1}^{L}\phi^D_{\underline{i}k}\phi^T_{\underline{i}k}(\Psi(\epsilon_{j,k,l}) - \Psi\left(\sum_{jkl}\epsilon_{j,k,l})\right) \\ &\quad + \phi^O_{\underline{i}j}\log\beta^O_{jo} - \phi^O_{\underline{i}j}\log\phi^O_{\underline{i}j} + \lambda_{\underline{i}}\left(\sum_{j=1}^{J}\phi^O_{\underline{i}j} - 1\right) \end{aligned}$$

To maximize it with respect to $\phi^O_{\underline{i}j}$, derivative is calculated and set to be zero:

$$\begin{aligned} \frac{\partial LB}{\partial\phi^O_{\underline{i}j}} &= \sum_{k=1}^{K}\sum_{l=1}^{L}\phi^D_{\underline{i}k}\phi^T_{\underline{i}k}\left(\Psi(\epsilon_{j,k,l}) - \Psi\left(\sum_{jkl}\epsilon_{j,k,l}\right)\right) \\ &\quad + \log\beta^O_{jo} - \log\phi^O_{\underline{i}j} - 1 + \lambda = 0 \end{aligned}$$

Then we have:

$$\phi^O_{\underline{i}j} \propto \beta^O_{jo}\exp\left[\sum_{k=1}^{K}\sum_{l=1}^{L}\phi^D_{\underline{i}k}\phi^T_{\underline{i}k}\left(\Psi(\epsilon_{j,k,l}) - \Psi\left(\sum_{jkl}\epsilon_{j,k,l}\right)\right)\right]$$

Same steps are followed to get $\phi^D_{\underline{i}k}$ and $\phi^T_{\underline{i}l}$.

A.2.2 Variational dirichlet

The term containing $\epsilon_{j,k,l}$ is simplified as follows:

$$\begin{aligned} LB_{[\epsilon]} &= \sum_{j,k,l}(\Psi(\epsilon_{j,k,l}) - \Psi\left(\sum_{j,k,l}\epsilon_{j,k,l})\right) \times \left(\alpha_{j,k,l} + \sum_{\underline{i}=1}^{N_u}\phi^O_{\underline{i}j}\phi^D_{\underline{i}k}\phi^T_{\underline{i}l} - \epsilon_{j,k,l}\right) \\ &\quad - \log\Gamma\left(\sum_{jkl}\epsilon_{j,k,l}\right) + \sum_{jkl}\log\Gamma(\epsilon_{j,k,l}) \end{aligned}$$

Similarly, to maximize the lower bound, derivative is calculated:

$$\frac{\partial LB}{\partial\epsilon_{j,k,l}} = \left(\alpha_{j,k,l} + \sum_{\underline{i}=1}^{N_u}\phi^O_{\underline{i}j}\phi^D_{\underline{i}k}\phi^T_{\underline{i}l} - \epsilon_{j,k,l}\right) \times \left(\Psi'(\epsilon_{j,k,l}) - \Psi'(\sum_{jkl}\epsilon_{j,k,l})\right) = 0$$

By setting the derivative as zero we have:

$$\epsilon_{j,k,l} = \alpha_{j,k,l} + \sum_{\underline{i}=1}^{N_u} \phi^O_{\underline{i}j}\phi^D_{\underline{i}k}\phi^T_{\underline{i}l}$$

A.3 Parameter estimation

A.3.1 Conditional multinomials $\mathbf{B}^O$, $\mathbf{B}^D$

This will be solved by gradient ascend algorithm due to the graph regularization. Terms in lower bound containing β^O_{jo} with $R(\mathbf{G}^O)$ and the constraint $\sum_{o=1}^{V^O}\beta^O_{jo} = 1$ considered are:

$$\begin{aligned} LB_{[\beta^O_{jo}]} = {} & \lambda \sum_{u=1}^{M}\sum_{\underline{i}=1}^{N_u}\sum_{j=1}^{J}\sum_{o=1}^{V^O} \phi^O_{u,\underline{i}j} w^O_{u,\underline{i}o}\log\beta^O_{jo} - (1-\lambda)R(\mathbf{G}^O) \\ & + \sum_{j=1}^{J} a^O_j \left(\sum_{o=1}^{V^O}\beta^O_{jo} - 1\right) \end{aligned}$$

Gradient $\nabla L(\boldsymbol{\beta}^O_j)$ with respect to β^O_{jo} is as follows:

$$\begin{aligned} \nabla L(\boldsymbol{\beta}^O_j) = \frac{\partial LB}{\partial \beta^O_{jo_1}} = {} & \lambda \frac{1}{\beta^O_{jo_1}} \sum_{u=1}^{M}\sum_{\underline{i}=1}^{N_u} \phi^O_{u,\underline{i}j}\mathbf{1}(w^O_{u\underline{i}} = o_1) \\ & - (1-\lambda)\sum_{o_2}(\mu\kappa^{G_{net}}_{o_1o_2} + (1-\mu)\kappa^{G_{poi}}_{o_1o_2})(\beta^O_{jo_1} - \beta^O_{jo_2}) + a^O_j . \end{aligned}$$

As mentioned in Section 4.3, $\mathbf{B}^O$ will be updated in an online stochastic manner, where gradient is calculated with only one observation s repeated M times, thus, gradient $\nabla L_s(\tilde{\boldsymbol{\beta}}^O_j)$ with respect to β^O_{jo} is

$$\begin{aligned} \nabla L_s(\tilde{\boldsymbol{\beta}}^O_j) = \frac{\partial LB_s}{\partial \beta^O_{jo_1}} = {} & \lambda \frac{M}{\beta^O_{jo_1}} \sum_{\underline{i}=1}^{N_s} \phi^O_{s,\underline{i}j}\mathbf{1}(w^O_{s\underline{i}} = o_1) \\ & - (1-\lambda)\sum_{o_2}(\mu\kappa^{G_{net}}_{o_1o_2} + (1-\mu)\kappa^{G_{poi}}_{o_1o_2})(\beta^O_{jo_1} - \beta^O_{jo_2}) + a^O_j . \end{aligned}$$

$\mathbf{B}^D$ will be updated same way.

A.3.2 Conditional multinomials B^T

There is no regularization in T dimension, so the estimation is similar with the work in (Blei et al. 2003).

$$\beta^T_{l,t} = \sum_{u=1}^{M}\sum_{i=1}^{N_u} \phi^T_{u,\underline{i},l}\mathbf{1}(w^T_{u\underline{i}} = t)$$

A.3.3 Dirichlet $\boldsymbol{\mathcal{A}}$

$\alpha_{j,k,l}$ will also be updated with Newton-Raphson method. The gradient $g_{j,k,l}$ with respect to $\alpha_{j,k,l}$ is as below:

$$\begin{aligned} g_{j,k,l} = \frac{\partial L}{\partial \alpha_{j,k,l}} &= M\left(\Psi(\sum_{jkl}\alpha_{j,k,l}) - \Psi(\alpha_{j,k,l})\right) \\ &+ \sum_{u=1}^{M}\left(\Psi(\epsilon_{u,j,k,l}) - \Psi\left(\sum_{jkl}\epsilon_{u,j,k,l}\right)\right) \end{aligned}$$

Then $\boldsymbol{\mathcal{A}}$ is updated as follows:

$$\alpha^{s+1}_{j,k,l} = \alpha^{s}_{j,k,l} - \left\{H^{-1}(\boldsymbol{\mathcal{A}})g(\boldsymbol{\mathcal{A}})\right\}_{j,k,l} = \alpha^{s}_{j,k,l} - \frac{g_{j,k,l} - c}{h_{j,k,l}}$$

where $c = \frac{\sum_{jkl} g_{j,k,l}/h_{j,k,l}}{z^{-1} + \sum_{jkl} h_{j,k,l}}$, $h_{j,k,l} = -M\Psi'(\alpha_{j,k,l})$, and $z = M\Psi'(\sum_{jkl}\alpha_{j,k,l})$ (Blei et al. 2003).

References

Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3(Jan):993–1022

Briand AS, Côme E, Trépanier M et al (2017) Analyzing year-to-year changes in public transport passenger behaviour using smart card data. Transp Res Part C: Emerg Technol 79:274–289

Chang J, Gerrish S, Wang C, et al (2009) Reading tea leaves: How humans interpret topic models. In: Advances in neural information processing systems, pp 288–296

Chen L, Jose JM, Yu H, et al (2016) A semantic graph based topic model for question retrieval in community question answering. In: Proceedings of the ninth ACM international conference on web search and data mining, pp 287–296

Cheng Z, Trépanier M, Sun L (2020) Probabilistic model for destination inference and travel pattern mining from smart card data. Transportation 48(4):2035–2053

Elhamifar E, Vidal R (2013) Sparse subspace clustering: Algorithm, theory, and applications. IEEE Trans Pattern Anal Mach Intell 35(11):2765–2781

Gao H, Nie F, Li X, et al (2015) Multi-view subspace clustering. In: Proceedings of the IEEE international conference on computer vision, pp 4238–4246

Geng X, Li Y, Wang L, et al (2019) Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 3656–3663

Griffiths TL, Steyvers M (2004) Finding scientific topics. Proc Natl Acad Sci 101(suppl 1):5228–5235

Guo S, Lin Y, Feng N, et al (2019) Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 922–929

Hoffman M, Bach FR, Blei DM (2010) Online learning for latent dirichlet allocation. In: advances in neural information processing systems, Citeseer, pp 856–864

Hu H, Lin Z, Feng J, et al (2014) Smooth representation clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3834–3841

Kolda TG, Bader BW (2009) Tensor decompositions and applications. SIAM Rev 51(3):455–500

Li D, Zamani S, Zhang J, et al (2019a) Integration of knowledge graph embedding into topic modeling with hierarchical dirichlet process. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp 940–950

Li X, Zhang J, Ouyang J (2019b) Dirichlet multinomial mixture with variational manifold regularization: Topic modeling over short texts. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 7884–7891

Li Z, Sergin ND, Yan H, et al (2020) Tensor completion for weakly-dependent data on graph for metro passenger flow prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 4804–4810

Li Z, Yan H, Zhang C, et al (2021) Tensor topic models with graphs and applications on individualized travel patterns. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE), IEEE, pp 2756–2761

Liu G, Lin Z, Yan S et al (2012) Robust recovery of subspace structures by low-rank representation. IEEE Trans Pattern Anal Mach Intell 35(1):171–184

Liu H, Tong Y, Zhang P, et al (2019) Hydra: A personalized and context-aware multi-modal transportation recommendation system. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 2314–2324

Mei Q, Cai D, Zhang D, et al (2008) Topic modeling with network regularization. In: Proceedings of the 17th international conference on World Wide Web, pp 101–110

Mohamed K, Côme E, Oukhellou L et al (2016) Clustering smart card data for urban mobility analysis. IEEE Trans Intell Transp Syst 18(3):712–728

Newman D, Lau JH, Grieser K, et al (2010) Automatic evaluation of topic coherence. In: Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics, pp 100–108

Parsons L, Haque E, Liu H (2004) Subspace clustering for high dimensional data: a review. ACM SIGKDD Explorations Newsl 6(1):90–105

Porteous I, Newman D, Ihler A, et al (2008) Fast collapsed gibbs sampling for latent dirichlet allocation. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 569–577

Ren J, Xie Q (2017) Efficient od trip matrix prediction based on tensor decomposition. In: 2017 18th IEEE International Conference on Mobile Data Management (MDM), IEEE, pp 180–185

Shi H, Yao Q, Guo Q, et al (2020) Predicting origin-destination flow via multi-perspective graph convolutional network. In: 2020 IEEE 36th International Conference on Data Engineering (ICDE), pp 1818–1821

Sun L, Axhausen KW (2016) Understanding urban mobility patterns with a probabilistic tensor factorization framework. Transp Res Part B: Methodol 91:511–524

Tang K, Chen S, Liu Z et al (2018) A tensor-based bayesian probabilistic model for citywide personalized travel time estimation. Transp Res Part C: Emerg Technol 90:260–280

Tang Y, Jiang Y, Yang H et al (2020) Modeling and optimizing a fare incentive strategy to manage queuing and crowding in mass transit systems: Modeling and optimizing a fare incentive strategy to manage queuing and crowding in mass transit systems. Transp Res Part B: Methodol 138:247–267

Teh YW, Newman D, Welling M (2007) A collapsed variational bayesian inference algorithm for latent dirichlet allocation. Tech. rep., CALIFORNIA UNIV IRVINE SCHOOL OF INFORMATION AND COMPUTER SCIENCE

Wang S, He L, Stenneth L, et al (2015) Citywide traffic congestion estimation with social media. In: Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, p 34

Wang Y, Yin H, Chen H, et al (2019) Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 1227–1235

Xiao H, Stibor T (2010) Efficient collapsed gibbs sampling for latent dirichlet allocation. In: Proceedings of 2nd asian conference on machine learning, JMLR Workshop and Conference Proceedings, pp 63–78

Yao L, Zhang Y, Wei B, et al (2017) Incorporating knowledge graph embeddings into topic modeling. In: Thirty-first AAAI conference on artificial intelligence

Yi D, Su J, Liu C et al (2019) A machine learning based personalized system for driving state recognition. Transp Res Part C: Emerg Technol 105:241–261

Yu B, Yin H, Zhu Z (2018) Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, pp 3634–3640

Yu K, He L, Philip SY et al (2019) Coupled tensor decomposition for user clustering in mobile internet traffic interaction pattern. IEEE Access 7:18,113-18,124

Zhang C, Hu Q, Fu H, et al (2017) Latent multi-view subspace clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4279–4287

Zhang C, Fu H, Hu Q et al (2018) Generalized latent multi-view subspace clustering. IEEE Trans Pattern Anal Mach Intell 42(1):86–99

Zhao J, Qu Q, Zhang F et al (2017) Spatio-temporal analysis of passenger travel patterns in massive smart card data. IEEE Trans Intell Transp Syst 18(11):3135–3146

Zhao Z, Koutsopoulos HN, Zhao J (2020) Discovering latent activity patterns from transit smart card data: A spatiotemporal topic model. Transp Res Part C: Emerg Technol 116(102):627

Zhong R, Lv W, Du B et al (2017) Spatiotemporal multi-task learning for citywide passenger flow prediction. 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), pp 1–8

Zhou M (2018) Nonparametric bayesian negative binomial factor analysis. Bayesian Anal 13(4):1065–1093

Zhu J, Ahmed A, Xing EP (2012) Medlda: maximum margin supervised topic models. J. Mach Learn Res 13(1):2237–2278

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.