Event Analytics via Discriminant Tensor Factorization

XIDAO WEN, University of Pittsburgh, USA
YU-RU LIN, University of Pittsburgh, USA
KONSTANTINOS PELECHRINIS, University of Pittsburgh, USA

Analyzing the impact of disastrous events has been central to understanding and responding to crises. Traditionally, the assessment of disaster impact has primarily relied on the manual collection and analysis of surveys and questionnaires as well as the review of authority reports. This can be costly and time-consuming whereas a timely assessment of an event's impact is critical for crisis management and humanitarian operations. In this work, we formulate the impact discovery as the problem to identify the shared and discriminative subspace via tensor factorization due to the multidimensional nature of mobility data. Existing work in mining the shared and discriminative subspaces typically requires the predefined number of either type of them. In the context of event impact discovery, this could be impractical, especially for those unprecedented events. To overcome this, we propose a new framework, called "`PairFac`," that jointly factorizes the multidimensional data to discover the latent mobility pattern along with its associated discriminative weight. This framework does not require splitting the shared and discriminative subspaces in advance and at the same time automatically captures the persistent and changing patterns from multidimensional behavioral data. Our work has important applications in crisis management and urban planning, which provides a timely assessment of impacts of major events in the urban environment.

CCS Concepts: • **Computing methodologies** → **Factorization methods**; • **Information systems** → *Wrappers (data mining)*;

Additional Key Words and Phrases: Urban Computing, Data Mining, Tensor Factorization , Event Analytics

ACM Reference Format:
Xidao Wen, Yu-Ru Lin, and Konstantinos Pelechrinis. 2018. Event Analytics via Discriminant Tensor Factorization. *ACM Trans. Knowl. Discov. Data.* 1, 1, Article 1 (January 2018), 40 pages.
https://doi.org/10.1145/3184455

1 INTRODUCTION

Analyzing the impact of disastrous events has been central to understanding and responding to crises. Effective crisis management requires not only careful planning and preparation for disaster relief operations, but also a timely assessment of an event's impact. The latter is important for facilitating actions that will bring the society back to its normal operations as

Correspondence: yurulin@pitt.edu.
Authors' addresses: Xidao Wen, University of Pittsburgh, 135 North Bellefield Avenue, Pittsburgh, PA, 15260, USA, xidao.wen@pitt.edu; Yu-Ru Lin, University of Pittsburgh, 135 North Bellefield Avenue, Pittsburgh, PA, 15260, USA, yurulin@pitt.edu; Konstantinos Pelechrinis, University of Pittsburgh, 135 North Bellefield Avenue, Pittsburgh, PA, 15260, USA, kpele@pitt.edu.

1556-4681/2018/1-ART1 $$15.00
https://doi.org/10.1145/3184455

fast as possible [24]. In this work, we introduce a novel event analysis framework that can automatically reveal the changes in human behavioral patterns associated with an event through mining context-rich, high-dimensional and potentially heterogeneous urban activity data.

Traditionally, the assessment of a (natural or artificial) disaster's impact has primarily relied on the manual administration and analysis of surveys and questionnaires, as well as the review of authority reports [29]. Both of these approaches are costly and time-consuming. Today, in the era of mobile and pervasive computing, rich digital human traces of routine transactions are generated by city-dwellers, businesses, and organizations that can be collected through online platforms (e.g., activities on social media), sensing technologies (e.g., mobile phones and wireless sensors) and other means (e.g., crowdsourcing platforms). These rich troves of human behavioral data provide an unprecedented opportunity to closely examine - both qualitatively and quantitatively - the changes in urban activity that follow events of interest (e.g., disasters). While much progress has been made in predictive event analytics, such as detecting and/or forecasting event outbreaks [2, 33, 35], automatically quantifying and capturing the impact of an event has been neglected despite its aforementioned importance.

Our objective in this work is to develop an automated method for understanding the impacts of major effects in the urban environment. To achieve our goal we design unsupervised learning techniques to uncover the changes in human mobility patterns before and after an event of interest. In particular, we formulate our objective as a problem of identifying common and discriminative subspaces between two datasets, the first one capturing the behavior of interest prior to the event and the second one capturing the behavior after the event. While there is literature on discriminant subspace learning [10, 14, 20], these solutions fall into the same generic framework that requires the split of shared and discriminative components before learning the subspaces. However, in the context of analyzing the impact of an event, this is not possible. The vast spectrum of disastrous events and the associated context under which they happen, make it extremely difficult to obtain this knowledge. Thus, most of the prior methods cannot be practically applied to disaster event analysis.

In this article, we introduce a novel approach that is able to automatically discover the impact of an exogenous event. While we are focusing on the impact of an event on urban mobility, our proposed method is generic and can be used to analyze multiple aspects of urban human activities. Our focus on the mobility will enable us to answer the question of how does the event change *when*, *where* and *what* citizens normally do in a city? To reiterate, our approach, called `PairFac`, formulates the problem as a discriminant tensor analysis problem and solves it through the joint factorization of a *pair* of tensors. Specifically, given two tensors capturing urban activity data *before* and *after* an event of interest, `PairFac` simultaneously learns the shared and discriminative latent subspaces of the tensor pairs. `PairFac` thus, reveals the patterns that both persist and change across multiple aspects of urban activity data.

The motivation for designing `PairFac` stems from the fact that understanding the impact of a disastrous event is a necessity in disaster management, while existing methods for discriminative subspace learning exhibit practical limitations in their applicability in reality. More specifically, in the context of disaster management, "impact assessment" plays a critical role in understanding the (social) consequences of an event. In this situation, "social impact" refers to the consequences an event has on human populations, altering the ways in which they live, work, entertain, relate to one another, organize to meet their needs, and cope as members of society [34]. The process of social impact assessment involves a number of steps, including among others "description of the relevant human environment and

zones of influence", "identification and investigation of probable impacts", and "estimation of secondary and cumulative impacts". These tasks are traditionally performed through manual, labor-intensive data collection, and comparison. For example, to describe the human environment and zones of influence, relevant data related to the event should be collected and reviewed through a baseline study or community profile. This approach has been limited in terms of scope and comprehensiveness, as it is not possible to scale to all potentially affected people, and is restricted by pre-defined assessment indices that do not necessarily universally apply. Therefore, a data-driven, generalizable, approach that can leverage the large volume of (detailed) data collected from various sources is needed and has the potential to revolutionize the traditional disaster impact assessment process.

Furthermore, existing approaches in analyzing events mostly focus on the impact discovery using one- or two- dimensional analysis (e.g., call activities volume changes [3], change in geographical location distributions [30], change in emotions [36]) and few are capable of discovering multi-dimensional (or multi-modal) impacts. This inevitably leads to significant loss of information associated with certain aspects that are either projected to a lower dimensionality that can be handled by the model used or eliminated all together. Moreover, the interplay between these multiple facets is not explicitly considered and could result in a false interpretation of the outcome. By formulating the problem event impact analysis as a tensor factorization problem, we are able to discover the changes that are correlated in multiple dimensions. For example, the change in the call volumes on top of any association in time can also vary depending on the location of these phone calls (e.g., their distance from the epicenter of the event).

Our method differs from existing work in discriminative subspace learning [10, 14, 20] by introducing the discriminative weight vector that allows for automatically aligning the common components while at the same time discerning the discriminative components. As shown in Fig. 1, we model the mobility data with two three-dimensional tensors[1], one describing the mobility before the event and one describing the behavior after the event of interest. As alluded to above, `PairFac` jointly factorizes the two tensors to identify the latent mobility patterns that both change, as well as persist, before and after the event of interest. Our comprehensive evaluations of `PairFac` on both synthetic and real-world event datasets clearly showcase its effectiveness.

The key contribution of this work includes:

- We formally introduce the problem of capturing the impact of an exogenous event on the normal operations of a system using discriminant tensor analysis. Given the multidimensional nature of the rich human behavioral data, we use tensor representation to preserve the interactions between different information layers, such as the temporal, spatial and human action layers (see Fig. 1).
- We propose `PairFac` (see Fig. 1), a novel joint tensor factorization framework that aims at simultaneously learning the shared and discriminative components from a pair of high-dimensional data sources. Our method can automatically identify the common components and at the same time discover the discriminative ones without a predefined number of either type, by formalizing an appropriate optimization problem.
- We provide an efficient iterative algorithm that guarantees convergence to a locally optimal solution for the aforementioned optimization problem. Furthermore, the algorithm is scalable with time complexity linear in the number of non-zero tensor

[1] `PairFac` can be extended to more dimensions. However, for illustrative purposes, as well as, due to the nature of our datasets, we design and evaluate our method with three-dimensional tensors.

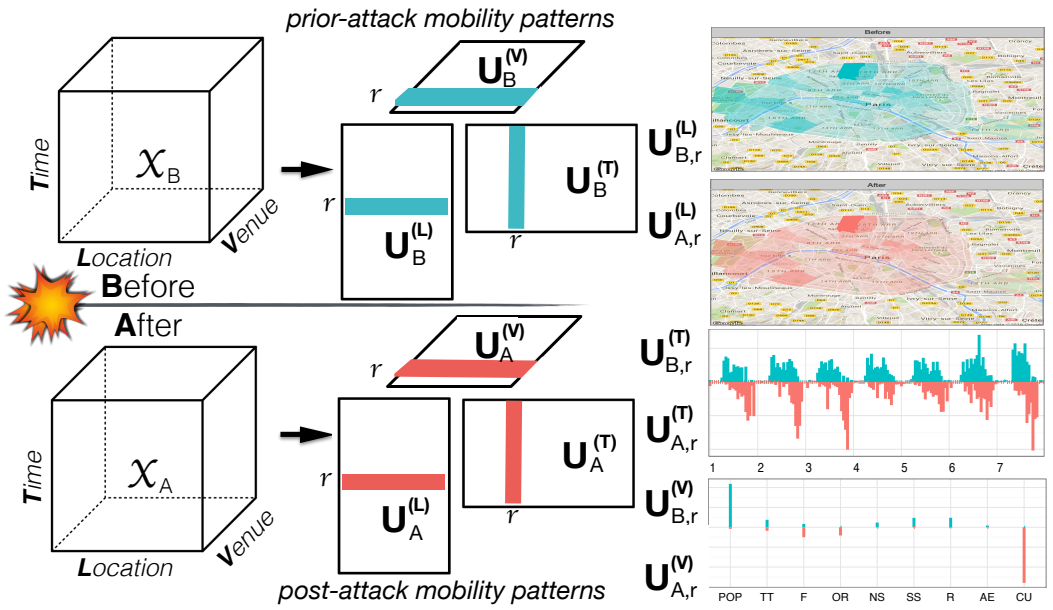


Fig. 1. **Problem illustration of the proposed discriminant tensor analysis.** $\mathcal{X}_B$ and $\mathcal{X}_A$ represents the data tensor ***B**efore* and ***A**fter* a specific event (Paris terrorist attack). Matrices $\mathbf{U}^{(L)}$, $\mathbf{U}^{(T)}$, and $\mathbf{U}^{(V)}$ represent the three factor matrices for ***L**ocation*, ***T**ime*, and ***V**enue*, respectively. The same-index columns in each triplet of factor matrix jointly represents a behavioral pattern. `PairFac` identifies similar and discriminative patterns before and after the event. For each pattern (e.g, colored in blue or red), we show the location distribution (e.g., $\mathbf{U}^{(L)}_{B,r}$, $\mathbf{U}^{(L)}_{A,r}$) in the city (of Paris), the time distribution (e.g., $\mathbf{U}^{(T)}_{B,r}$, $\mathbf{U}^{(T)}_{A,r}$) in a week (24×7) and the venue distribution (e.g., $\mathbf{U}^{(V)}_{B,r}$, $\mathbf{U}^{(V)}_{A,r}$) among different activities (e.g., Professional & Other Places (POP), Travel & Transportation (TT), Food (F); please refer to 6.1.2 for details.)

elements. In addition, we provide guidance on a parallel implementation of the algorithm based on Spark that can further speed up the optimization.

This article represents a significant extension of our prior work [37] and is our first full discussion on this subject. In this article, we include new solutions, algorithmic details, and proofs, as well as extensive experimental results. In particular, there are several major developments since our previous work [37]:

- We introduce a new algorithm that provides better interpretation of the discriminative weights while at the same time achieving component alignment. In particular, we introduce an additional auxiliary function to capture the commonalities between the pair of tensors. This additional information gives rise to an easier interpretation of the discriminative scores - i.e., higher scores represent unique patterns while lower scores indicate shared patterns. In addition, our prior work relies on a post-hoc analysis

of the learned components to determine the pair-wise alignment of the common components. We address this limitation by re-formulating our objective function with a new regularization term to enforce the similarity between common components.
- We provide a detailed algorithmic description and analysis in addition to a parallelized version of the algorithm. More specifically, we provide details for the solution of our formulated optimization problem, while at the same time providing a theoretical analysis of its convergence. In addition, we provide a scalable, distributed implementation of `PairFac` that speeds up the runtime, through the partition of tensors into mutually non-overlapped blocks. This allows the gradient update in each step to be computed via multiple nodes.
- We perform comprehensive experiments on the scalability and sensitivity of `PairFac`. We also apply `PairFac` to extensive case studies on real events. To better understand how to appropriately apply the algorithms to event analytics in practice, we systematically analyze the separability of data with respect to the ability of `PairFac` to segment the components into common and discriminative parts. We further employ our approach to discover the long-term impact of terrorist attacks in Paris using traffic sensor data and Twitter geo-tagged content. Another case study is conducted for discovering the changes in mobility patterns during the Thanksgiving week between 2014 and 2015. We demonstrate that our approach can not only distill the crowd activity patterns under exogenous shocks but also analyze long-term activity changes.

The rest of this paper is organized as follows. Section 2 discusses literature related to our study, while Section 3 presents the problem formulation and the essential background. In Section 4, We introduce multiple solutions to the tensor factorization problem, including a novel algorithm that automatically learns the discriminative weights of the components. Section 5 provides detailed quantitative results on synthetic datasets, while Section 6 presents our case studies. In Section 7 we discuss some open issues and future directions, while Section 8 concludes our work.

2 RELATED WORK

In this section, we describe literature relevant to our methodology and to event and urban analytics.

2.1 Shared and Discriminative Subspace Learning

The increasing availability of data from a diverse set of sources has given rise to the study of joint analysis of heterogeneous data. Our study closely relates to the area of discriminative tensor analysis. For example, GTDA (General Tensor Discriminant Analysis) [32] attempts to discover the discriminative features of a pair of tensors as a preprocessing step for subsequent topic discovery and classification tasks, while TCCA (Tensor Canonical Correlation Analysis) [21] generalizes Canonical Correlation Analysis (CCA) to handle the data of an arbitrary number of views or distinct feature sets and identifies a reliable common subspace shared by all views. Compared to these studies, `PairFac` attempts to simultaneously identify both common and discriminative subspace from the multi-dimensional dataset. The simultaneous discovery of common and discriminative subspace is not new. However, it is typically limited to two dimensions at most. For instance, Gupta et al. [9] propose a joint NNMF on two data sources through a shared subspace, while maintaining their unique variations through individual subspaces. While Gupta et al. [10] further impose mutually orthogonal regularizations to separate the common and discriminative subspaces to ensure that the

shared and the discriminative subspaces are mutually exclusive. Following the same idea, Kim et al. [14] relax the framework by requiring the shared subspaces to be *similar* while not necessarily being strictly identical. Regarding the shared and discriminative subspace learning in the context of tensor factorization, the framework by Liu et al. [20] - similar to [9] - separates the subspace into shared and individual subspaces. In contrast, `PairFac` imposes regularization on the shared and discriminative subspaces to automatically identify the number of either type of components, while offering scalability by enabling factorization of even higher dimensional data.

2.2 Event Analytics

During the last few years, there has been an increasing interest in the area of event analytics through microblogs (e.g., Twitter). Researchers have approached this field from three perspectives. One line of research is geared towards large-scale societal event detection and forecasting (e.g., civil unrest, disease outbreak, and elections). A common technique is to monitor the frequency of all words and look for a sudden burst in the frequency of (a subset of) them [23]. For instance, Ning et al. [25] develop a multiple instance learning based approach to identify evidence-based precursors and forecast events into the future.

The second line of research aims at sense making of an event's storyline through statistical analysis or visual analytics. For example, Diakopoulos et al. [8] design a visual analytics tool to help journalists and media professionals extract news-worthy content from a large volume of social media data.

Our work falls into the third line of research, which aims at studying the impact of an event on the affected population. E.g., Lin and Margolin [18] explore the emotional response of Twitter users in different cities to the bombing attacks in Boston, while, Bagrow et al. [3] provide a quantitative view of the behavioral changes in human activity under extreme (natural and man-made) conditions, such as bomb attacks and earthquakes, through the analysis of mobile phone records. In a similar direction, Song et al. [30] mined GPS traces of 1.6 million users and built a system to automatically discover, analyze, and simulate the mobility of a large population under severe disasters in Japan. The shortcoming of using only cell phone and GPS data is that the activity context is absent. Including information relevant to activity concept significantly complicates the analysis due to the increased dimensionality of the data.

2.3 Urban Computing

In recent years, there has been a significant volume of research in the area of urban computing and informatics. Zheng et al. [42] summarize seven types of urban computing scenarios for urban planning, transportation, environment, energy, social issues, economy, and public safety and security. Our work, from an application perspective, falls into the last category, as we seek to obtain an understanding of the impact of exogenous events on urban space. Recently, there have been several inspiring studies looking at urban environments. For instance, Pan et al. [26] detect traffic anomalies based on drivers' routing behavior on road networks, while, Pang et al. [27] apply the likelihood ratio test (LRT) (widely used in epidemiological studies) to describe traffic patterns. Our research is geared more towards the area of disaster analytics in urban environments. Early forecasting and detection of disasters are critical for planning effective humanitarian interventions and disaster management. However, there is plenty of literature in this realm, and it is not the focus of our study. For example, Lee and Sumiya [17] propose to detect events such as environmental disasters from geo-tagged Twitter data, while Yu et al. [40] propose multiple Markov boundaries in local

causal discovery to identify the sets of precursors for tornado forecasting. In another study, Madaio [22] developed the *Firebird* framework to help municipal fire departments identify and prioritize commercial property fire inspections, with a combination of techniques from machine learning, geocoding and information visualization. Finally, the short- and long-term evacuation plans/behaviors in the case of a disaster have also been studied [30, 31].

The contribution of our work resides in the area of disaster impact discovery from multidimensional and heterogeneous data. `PairFac` is a generic framework that can be used to study the impact of various exogenous events – being either natural, man-made, or imposed by the local government (e.g., planning policies). For example, the impact of a long-term construction project on the inhabitants' mobility and activities can be quantified using `PairFac`. Identifying behavioral changes for a variety of "urban interventions" has been identified as an open problem pertaining particularly to urban computing [42].

3 PRELIMINARIES AND PROBLEM FORMULATION

In this section, we provide the necessary tensor theory background and notations for the design of `PairFac`, followed by the problem formulation.

3.1 Preliminaries

3.1.1 Tensors. A tensor is a mathematical representation of a multidimensional array. Table 1 presents the notation we use in the rest of the paper. We use x to represent a scalar, $\mathbf{x}$ a vector, $\mathbf{X}$ a matrix, and $\boldsymbol{\mathcal{X}}$ a tensor. We further use $\mathbf{x}_i$ to denote the i-th entry of vector $\mathbf{x}$, $\mathbf{X}_{ij}$ to denote the element of matrix $\mathbf{X}$ at position $\{i, j\}$ and $\boldsymbol{\mathcal{X}}_{ijk}$ to denote the element of tensor $\boldsymbol{\mathcal{X}}$ at position $\{i, j, k\}$. The *order* of a tensor is the number of dimensions (also referred to as modes, or ways). The *dimensionality* of a mode is the number of elements in that mode. We use I_q to denote the dimensionality of the q-th mode. For example, the three-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}_+^{I_1 \times I_2 \times I_3}$ has three modes with dimensionality of I_1, I_2, and I_3, respectively. $\mathbb{R}_+$ indicates that all the elements of $\boldsymbol{\mathcal{X}}$ obtain non-negative values.

Symbol	Description
x	a scalar (lower-case letter)
$\mathbf{x}$	a vector (boldface lower-case letter)
$\mathbf{X}$	a matrix (boldface capital letter)
$\boldsymbol{\mathcal{X}}$	a tensor (boldface Euler script letter)
$\mathbf{X}_{i,j}$	the scalar at the $\{i, j\}$ position of matrix $\mathbf{X}$
$\boldsymbol{\mathcal{X}}_{\boldsymbol{i,j,k,\ldots}}$	the scalar at the $\{i, j, k...\}$ position of $\boldsymbol{\mathcal{X}}$
$\mathbf{X}_{(n)}$	mode-n unfolding of tensor $\boldsymbol{\mathcal{X}}$
$\mathbf{U}^{(n)}$	mode-n factor matrix of tensor $\boldsymbol{\mathcal{X}}$
$\mathbf{U}_r^{(n)}$	the r-th column in mode-n factor matrix of tensor $\boldsymbol{\mathcal{X}}$
I_1,..., I_M	the dimensionality of mode 1, ..., M
R	the desired rank (Capital Italic script letter)

Table 1. Description of Notations.

3.1.2 Basic Operations. **Mode-n matricization or unfolding:** Matricization is the process of reordering the elements of an M-way array into a matrix. A mode-n matricization of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$ is denoted by $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times \prod_{q \neq n}^{M} I_q}$.

Mode-n product: The mode-n matrix product of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$ with a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ is denoted by $\boldsymbol{\mathcal{X}} \times_n U$ and is a new tensor of size $I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$ with $(\boldsymbol{\mathcal{X}} \times_n \mathbf{U})_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_N} u_{j i_n}$.

Tensor Decomposition: Given an input tensor, tensor factorization decomposes it into a smaller/core tensor multiplied by a matrix along each mode. For the case of a three-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, we have $\boldsymbol{\mathcal{X}} \approx \boldsymbol{\mathcal{Z}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$. Matrices $\mathbf{A} \in \mathbb{R}^{I_1 \times O}$, $\mathbf{B} \in \mathbb{R}^{I_2 \times P}$, and $\mathbf{C} \in \mathbb{R}^{I_3 \times Q}$ are called *factor matrices*, or *factors/components*, while tensor $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is called the *core tensor*. In this process, each element of the tensor $\boldsymbol{\mathcal{X}}$ is the product of the corresponding factor matrix elements multiplied by a weight $\boldsymbol{\mathcal{Z}}_{opq}$, i.e., $\boldsymbol{\mathcal{X}}_{i_1 i_2 i_3} \approx \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \sum_{i_3=1}^{I_3} \boldsymbol{\mathcal{Z}}_{opq} \mathbf{A}_{o i_1} \mathbf{B}_{p i_2} \mathbf{C}_{q i_3}$.

CP Decomposition: CANDECOMP/PARAFAC [11] decomposition is often referred to as CP. The CP decomposition of tensor $\boldsymbol{\mathcal{X}}$ could be expressed as $\boldsymbol{\mathcal{X}}_{opq} \approx \sum_{r=1}^{R} \mathbf{A}_{or} \mathbf{B}_{pr} \mathbf{C}_{qr}$. Let $[\mathbf{z}]$ denote a superdiagonal tensor, where $[\cdot]$ is the operation that transforms vector $\mathbf{z}$ to a superdiagonal tensor by setting tensor element $z_{k \ldots k} = \mathbf{z}_k$ and other elements as 0. Thus the CP decomposition of a three-way tensor can be written as $\boldsymbol{\mathcal{X}} \approx [\mathbf{z}] \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$. Following Kolda [15], the CP model can be concisely expressed as $\boldsymbol{\mathcal{X}} \approx [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \equiv \sum_{r=1}^{R} \mathbf{A}_r \circ \mathbf{B}_r \circ \mathbf{C}_r$.

3.2 Problem Formulation

Simultaneous Discovery of Common and Discriminative Activity Patterns:

Problem Definition 1. *Let us consider two non-negative tensors, $\boldsymbol{\mathcal{X}}_B \in \mathbb{R}^{I_L \times I_T \times I_V}$ and $\boldsymbol{\mathcal{X}}_A \in \mathbb{R}^{I_L \times I_T \times I_V}$ representing the urban activities **B**efore (B) and **A**fter (A) an exogenous shock, where the tensor modes represents the **L**ocation (L), **T**ime (T) and **V**enue (V) of the activities. We seek to obtain a non-negative tensor factorization (NTF) to approximate both input tensors, as $\boldsymbol{\mathcal{X}}_q \approx [\![\mathbf{U}_q^{(L)}, \mathbf{U}_q^{(T)}, \mathbf{U}_q^{(V)}]\!]$, $\forall q \in \{A, B\}$, and $\mathbf{U}_q^{(m)} \in \mathbb{R}_+^{I_m \times R}$, $\forall m \in \{L, T, V\}$, represents the factor matrices corresponding to each mode.*

Note, as alluded to above, that in this work we focus on three-mode tensors but `PairFac` can be used to deal with data with higher dimensionality. The location dimension corresponds to specific neighborhoods in the city, the time is quantized hourly, while the venue dimension captures the various types of establishments available (e.g., coffee shops, retail shops, etc.). The term "venue" refers to the kind of place people visited, e.g., restaurants, schools, etc. – that is, the semantics of the human activity, whereas "location" refers to the geographical location that certain activity occurs. In our data, "venue" information is available from Foursquare that describes the functionality or the activity provided by the point of interest (or location). As shown in Fig.1, the corresponding columns (red) of each factor matrix together define a mobility pattern that associates specific areas/neighborhoods, time, and types of venues. Disastrous events, such as terrorist attacks, can inject intensive psychological instabilities in the targeted population and as a result the mobility and/or behavioral patterns of this population are likely to change after the event. The goal of the problem described above is to discover the shared and discriminative components of the tensor structures describing their urban activities before and after an event of interest.

4 SOLUTIONS

In this section, we begin with providing solutions to Problem 1. We start by describing the current state-of-the-art approaches to solving similar problems [10, 14, 20] (Sections 4.1 and 4.2). We then discuss their limitations and introduce a new solution (Section 4.3). We further provide a parallel implementation of our solution in Section 4.4.

4.1 Shared and Discriminative Subspace Approach

To learn the shared and discriminative subspace, Liu et al. [20] proposed the **C**ommon and **D**iscriminative subspace **N**on-negative **T**ensor **F**actorization (CDNTF) which takes a set of tensors as its input and computes both their common and discriminative subspaces simultaneously as the output. Following their work, the objective of CDNTF can be re-written as the following simultaneous factorization of two input tensors: $\boldsymbol{\mathcal{X}}_q \approx [\![\mathbf{U}_q]\!]$, where $\mathbf{U}_q = [(\mathbf{U}^{(m)}_{q:C}, \mathbf{U}^{(m)}_{q:D_q})]$, $\forall q$, $\forall m$, and $\mathbf{U}^{(m)}_q \in \mathbb{R}^{I_m \times R}_+$. In this way, the columns of matrix $\mathbf{U}^{(m)}_q$ are segmented into two parts: $\mathbf{U}^{(m)}_{q:C}$ represents the common subspace, while $\mathbf{U}^{(m)}_{q:D_q}$ represents the discriminative components to each tensor. The above common and discriminative subspace discovery is the solution to the minimization of the following objective function:

$$J_0 = \sum_{q \in \{A,B\}} \frac{1}{n_q} \left\| \boldsymbol{\mathcal{X}}_q - [\![(\mathbf{U}^{(m)}_{q:C}, \mathbf{U}^{(m)}_{q:D_q})]\!] \right\|^2_F \tag{1}$$

where $\mathbf{U}^{(m)}_{q:C}$ and $\mathbf{U}^{(m)}_{q:D}$ are defined as above, n_q is the Frobenius norm of each tensor, and $\|\cdot\|^2_F$ stands for the Frobenious norm.

4.2 Regularized Shared and Discriminative Subspace Approach

Shared and discriminative subspace learning have also been explored in the context of nonnegative matrix factorization. In fact, CDNTF can be thought of as the extension of nonnegative shared subspace learning (JSNMF [9]) to higher dimensions. Under this framework, Gupta et al. [10] propose regularized nonnegative shared subspace learning that further imposes a mutual orthogonality constraint on the constituent subspace, which segregates the patterns. In the context of discovering common and discriminative mobility patterns, we extend the framework to **R**egularized **J**oint **S**ubspace **N**onnegative **T**ensor **F**actorization (RJSNTF) and with a slight abuse of notation, we derive the following minimization problem:

$$J_1 = J_0 + \sum_{m \in \{L,V,T\}} J_{R1}(\mathbf{U}^{(m)}_{q:C}, \mathbf{U}^{(m)}_{q:D_q}, \mathbf{U}^{(m)}_{q':D_{q'}}), \tag{2}$$

where $\mathbf{U}^{(m)}_{q:C}$ and $\mathbf{U}^{(m)}_{q:D_q}$ are defined as above. Focusing on our application, q' in Eq. 2 represents the time after the event that is different from q (time prior to the event) with $\mathbf{U}^{(m)}_{q:C} = \mathbf{U}^{(m)}_{q':C}$, and $\mathbf{U}^{(m)}_{q:D_q} \neq \mathbf{U}^{(m)}_{q':D'_q}$. Therefore, $J_{R1}(\cdot)$ is a regularization function used to penalize the "similarity" between subspaces spanned in $\{\mathbf{U}^{(m)}_q\}$ and $\{\mathbf{U}^{(m)}_{q'}\}$. Following [10], the mutually orthogonal constraints are defined as:

$$J_{R1} = \hat{\alpha} \left\| \mathbf{U}^{(m)T}_{q:C} \mathbf{U}^{(m)}_{q:D_q} \right\|^2 + \hat{\beta} \left\| \mathbf{U}^{(m)T}_{q:C} \mathbf{U}^{(m)}_{q':D_{q'}} \right\|^2 + \hat{\gamma} \left\| \mathbf{U}^{(m)T}_{q:D_q} \mathbf{U}^{(m)}_{q':D_{q'}} \right\|^2, \tag{3}$$

where $\hat{\alpha}$, $\hat{\beta}$ and $\hat{\gamma}$ are the regularization parameters. When $J_{R1} = 0$, the model becomes identical to CDNTF.

RJSNTF enforces the shared components to be strictly identical, which is a hard constraint and might result in distortion during the factorization. Kim et al. [14] have proposed the simultaneous discovery of common and discriminative topics via joint non-negative matrix factorization where this constraint is relaxed by redefining the regularization term. Their approach further emphasizes the similarities and differences of the common and

discriminative components. Following the same idea and replacing $\mathbf{U}^{(m)}_{q:C}$ with $\mathbf{U}^{(m)}_{q:C_q}$ and $\mathbf{U}^{(m)}_{q':C_{q'}}$ to represent the similar components of tensors $\boldsymbol{\mathcal{X}}_q$ and $\boldsymbol{\mathcal{X}}_{q'}$. we derive **S**imultaneous **D**iscovery of **C**ommon and **D**iscriminative **N**onnegative **T**ensor **F**actorization (SDCDNTF) as the following minimization function:

$$J_2 = J_0 + \sum_{m\in\{L,V,T\}} J_{R2}(\mathbf{U}^{(m)}_{q:C_q}, \mathbf{U}^{(m)}_{q:D_q}, \mathbf{U}^{(m)}_{q':C_{q'}}, \mathbf{U}^{(m)}_{q':D_{q'}}), \tag{4}$$

and

$$J_{R2} = \alpha \left\| \mathbf{U}^{(m)}_{q:C_q} - \mathbf{U}^{(m)}_{q':C_{q'}} \right\|^2 + \beta \left\| \mathbf{U}^{(m)}_{q:D_q}{}^{T} \mathbf{U}^{(m)}_{q':D_{q'}}) \right\|_{1,1}, \tag{5}$$

where $\|\cdot\|_{1,1}$ denotes the absolute sum of all the matrix entries.

4.3 Automatic Discovery of Discriminative Components

4.3.1 Our `PairFac` *Formulation.* The above approaches fall under the same framework that splits the tensors' components into common and discriminative parts in advance, discovering these components with different regularization. These approaches require the number of shared (or distinct) components to be determined beforehand, which is difficult in practice. In this paper, we propose a novel factorization method, which we term `PairFac`, that does not require manual input of the shared or distinct components. In order to achieve that, we assign a weight to each component that reflects the *discriminative coefficient* or *score* of the corresponding component.

For this purpose, we introduce two auxiliary data tensors $\boldsymbol{\mathcal{Z}}_B$ and $\boldsymbol{\mathcal{Z}}_A$ that represent the aggregated unique patterns found in each tensor respectively. We first define the following function to compute these auxiliary tensors.

Definition 2. *Given a data tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times I_2\times I_3}$, $G(\boldsymbol{\mathcal{X}})$ is a clamping function that outputs a tensor $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{I_1\times I_2\times I_3}$ that is derived from the input tensor $\boldsymbol{\mathcal{X}}$ with entries restricted to a given value range such that $\boldsymbol{\mathcal{Z}} = G(\boldsymbol{\mathcal{X}})$, where $G(\boldsymbol{\mathcal{X}})$ is defined as:*

$$G(\boldsymbol{\mathcal{X}}) = \begin{cases} \boldsymbol{\mathcal{X}}_{i_1i_2i_3}, \textit{if } \boldsymbol{\mathcal{X}}_{i_1i_2i_3} > \epsilon, \\ 0, \textit{otherwise}, \end{cases} \tag{6}$$

where ϵ is a constant that defines the minimum entry in the tensor $\boldsymbol{\mathcal{Z}}$. ϵ can be empirically chosen to control the sparsity of the auxiliary tensors (we use $\epsilon = 0$ in this work). Note that the clamping function $G(\cdot)$ can also work with vectors and matrices. Then we compute $\boldsymbol{\mathcal{Z}}_q$ that captures the unique variance in $\boldsymbol{\mathcal{X}}_q$ from $\boldsymbol{\mathcal{X}}_{q'}$ as:

$$\boldsymbol{\mathcal{Z}}_q = G\left(\boldsymbol{\mathcal{X}}_q - \boldsymbol{\mathcal{X}}_{q'}\right), q \in \{A, B\} \tag{7}$$

Definition 3. *Given two data tensors $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1\times I_2\times I_3}$, $G'(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}})$ is a binary clamping function defined as:*

$$G'(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}) = \begin{cases} 1, \textit{if } |\boldsymbol{\mathcal{X}}_{i_1i_2i_3} - \boldsymbol{\mathcal{Y}}_{i_1i_2i_3}| < \epsilon', \\ 0, \textit{otherwise}, \end{cases} \tag{8}$$

where ϵ' is a constant that defines the minimum entry in the tensor $\boldsymbol{\mathcal{Z}}$. ϵ' can also be empirically chosen to control the sparsity of the auxiliary tensors (we use $\epsilon' = 0$ as well).

Now with function $G^{'}$, we derive another auxiliary tensor $\mathcal{S}$ defined as:

$$\boldsymbol{\mathcal{S}_q} = G^{'}\left(\boldsymbol{\mathcal{X}}_q, \boldsymbol{\mathcal{X}}_{q^{'}}\right), q \in \{A, B\} \tag{9}$$

We further introduce the weight vectors $\mathbf{w}_q \in \mathbb{R}_+^R$ to capture the discriminative coefficient of each component. Given $\mathcal{S}$, we want to enforce $(1 - \mathbf{w}_q)$ to represent the contribution of the corresponding components to the common parts of the two tensors. Note that in Equation 9 we use a binary clamping function to infer the $\mathcal{S}$ tensor. This function captures the common factors as the ones whose differences are no larger than $\epsilon^{'}$. The choice of $\epsilon^{'}$ allows for imposing the degree of sparsity in the $\mathcal{S}$ tensor, which the stochastic version of optimization benefits from (introduced in Section 4.4). Besides, the binary clamping function enforces the non-common part to be zero such that the weights $(1 - \mathbf{w}_q)$ gives a clearer interpretation directly related to the degree of how much the pattern contributes towards to the common tensor. Our intuition is that while factorizing the original tensors into its latent patterns, we would like to find a discriminative score for each pattern that corresponds to its unique contribution in each tensor. At the same time, we want to find a score that represents the commonality of a component in the two tensors. With the notations presented, we formally derive the minimization objective of `PairFac` as:

$$J_3 = J_0^{'} + J_{R3} + J_{R4}, \tag{10}$$

where $J_0^{'}$ differs from J_0 in that it does not require the manual split of common and discriminative parts in the factor matrix, and J_{R3} is a function to factorize the auxiliary tensors, defined as:

$$J_{R3} = \alpha \sum_{q \in \{A,B\}} \left\| \boldsymbol{\mathcal{Z}}_q - [\![\mathbf{w}_q; [\mathbf{U}_q]]\!] \right\|^2 + \beta \sum_{q \in \{A,B\}} \left\| \boldsymbol{\mathcal{S}}_q - [\![(1 - \mathbf{w}_q); [\mathbf{U}_q]]\!] \right\|^2, \tag{11}$$

where $\mathbf{w}_q$ is the *level of discriminativeness* associated with component $\mathbf{U}_q$. According to Eq. 11, the degree of which $\mathbf{U}_q$ contributes towards the reconstruction of $\mathcal{Z}_q$ is determined by $\mathbf{w}_q$, and $\mathcal{Z}_q$ captures the information of predominant "discriminative" part between the two (before- and after-) tensors. Similarly, (1-$\mathbf{w}_q$) can be thought of as the degree of commonality associated with $\mathbf{U}_q$ as the reconstruction of $\mathcal{S}_q$ depends on $\mathbf{U}_q$ but weighted by (1-$\mathbf{w}_q$). Unlike $\mathcal{Z}_q$, $\mathcal{S}_q$ captures the information of predominant "common" part shared in both (before- and after-) tensors.

J_{R4} in Eq. 10 is a function to align the components in the order such that similar components should share similar weights as the result of the factorization:

$$J_{R4} = \gamma \sum_{m \in \{L,V,T\}} \sum_{j}^{R} \left\| (1 - \mathbf{w}_{q_j}) \mathbf{U}_{qj}^{(m)} - (1 - \mathbf{w}_{q^{'}_j}) \mathbf{U}_{q^{'}j}^{(m)} \right\|^2, \tag{12}$$

where $\mathbf{U}_{qj}^{(m)}$ is the jth column of factor matrix $\mathbf{U}_{qj}^{(m)}$ and $\mathbf{w}_{q_j}$ is its associated score.

Note that Eq. 10 differs from the objective defined in our prior work [37]. In [37], the objective is given as:

$$J_3^* = J_0^{'} + J_{R3}^*, \tag{13}$$

where

$$J_{R3}^* = \alpha \sum_{q \in \{A,B\}} \left\| \boldsymbol{\mathcal{Z}}_q - [\![\mathbf{w}_q; [\mathbf{U}_q]]\!] \right\|^2. \tag{14}$$

Compared to J^*_{R3} in Eq. 13, J_{R3} in Eq. 11 has the addition of a second term, which uses the auxiliary tensor $\mathcal{S}$. Our prior work attempts to model the level of uniqueness of each component i captured by the weight w_i. With the addition of $(1 - w_i)$, we can interpret it as the level of contribution to the commonality between the two tensors. Moreover, the output of Eq. 13 in our prior work [37] splits the tensor factors into common and discriminative components but is not able to identify directly the pair-wise common components across tensors. Previously, we addressed this problem through post-hoc analysis on examining the pair-wise similarity of the components, which could be cumbersome. In this study, we expand on our prior work by introducing J_{R4} to automatically align the common components in order.

To solve Eq. 10 we use the block coordinate descent method. Consider the updating of $\mathbf{U}_q^{(m)}$ at iteration k. Using the fact that if $\boldsymbol{\mathcal{X}}_q = \mathbf{U}_q^{(m)} \circ \mathbf{U}_q^{(m')} \circ \mathbf{U}_q^{(m'')}$, then $\boldsymbol{X}_{q(m)} = \mathbf{U}_q^{(m)}(\mathbf{U}_q^{(m'')} \odot \mathbf{U}_q^{(m')})^T$, where $\boldsymbol{X}_{q(m)}$ is the unfolded matrix of $\boldsymbol{\mathcal{X}}_{\boldsymbol{q}}$ m-th mode. The objective (J_3) can be then re-written as:

$$\begin{aligned}
\text{minimize } \frac{1}{2} \sum_{q\in\{A,B\}} \Big(& \frac{1}{n_q} \left\| \boldsymbol{X}_{q(m)} - \mathbf{U}_q^{(m)}(\mathbf{U}_q^{(m'')} \odot \mathbf{U}_q^{(m')})^T \right\|^2 \\
& + \alpha \left\| \boldsymbol{Z}_{q(m)} - \mathbf{U}_q^{(m)} \Lambda_{\mathbf{w}_q} (\mathbf{U}_q^{(m'')} \odot \mathbf{U}_q^{(m')})^T \right\|^2 \\
& + \beta \left\| \boldsymbol{S}_{q(m)} - \mathbf{U}_q^{(m)}(\mathbf{I} - \Lambda_{\mathbf{w}_q})(\mathbf{U}_q^{(m'')} \odot \mathbf{U}_q^{(m')})^T \right\|^2 \Big) \\
& + \gamma \sum_{m\in\{L,V,T\}} \left\| \mathbf{U}_q^{(m)}(\mathbf{I} - \Lambda_{\mathbf{w}_q}) - \mathbf{U}_{q'}^{(m)}(\mathbf{I} - \Lambda_{\mathbf{w}_{q'}}) \right\|^2,
\end{aligned} \tag{15}$$

where $\odot$ denotes the Khatri-Rao product, $\mathbf{I} \in \mathbb{R}^{R\times R}$ is the identity matrix, $\Lambda_{\mathbf{w}_q}$ is a diagonal matrix with $\mathbf{w}_q$ as its diagonal entries, and m' and m'' are used to index factor matrices other than $\mathbf{U}_q^{(m)}$. The gradient with respect to $\mathbf{U}_q^{(m)}$ is given as:

$$\begin{aligned}
\nabla_{\mathbf{U}_q^{(m)}} J_3 = & \frac{1}{n_q} \left(\mathbf{U}_q^{(m)}(\mathbf{U}_q^{(m')} \odot \mathbf{U}_q^{(m'')})^T - \boldsymbol{X}_{q(m)} \right) (\mathbf{U}_q^{(m')} \odot \mathbf{U}_q^{(m'')}) \\
& + \alpha \left(\mathbf{U}_q^{(m)} \Lambda_{\mathbf{w}_q} (\mathbf{U}_q^{(m')} \odot \mathbf{U}_q^{(m'')})^T - \boldsymbol{Z}_{q(m)} \right) (\mathbf{U}_q^{(m')} \odot \mathbf{U}_q^{(m'')}) \Lambda_{\mathbf{w}_q}^T \\
& + \beta \left(\mathbf{U}_q^{(m)}(\mathbf{I} - \Lambda_{\mathbf{w}_q})(\mathbf{U}_q^{(m')} \odot \mathbf{U}_q^{(m'')})^T - \boldsymbol{S}_{q(m)} \right) (\mathbf{U}_q^{(m')} \odot \mathbf{U}_q^{(m'')})(\mathbf{I} - \Lambda_{\mathbf{w}_q})^T \\
& + \gamma \left(\mathbf{U}_q^{(m)}(\mathbf{I} - \Lambda_{\mathbf{w}_q}) - \mathbf{U}_{q'}^{(m)}(\mathbf{I} - \Lambda_{\mathbf{w}_{q'}}) \right) (\mathbf{I} - \Lambda_{\mathbf{w}_q}).
\end{aligned} \tag{16}$$

Let

$$\mathbf{F}^{k-1}_{\mathbf{U}_q^{(m)}} = \mathbf{U}^{(m')}_{q,k-1} \odot \mathbf{U}^{(m'')}_{q,k-1}. \tag{17}$$

We take

$$\mathcal{L}^{k-1}_{\mathbf{U}_q^{(m)}} = \left\| {\mathbf{F}^{k-1}_{\mathbf{U}_q^{(m)}}}^T \mathbf{F}^{k-1}_{\mathbf{U}_q^{(m)}} \right\|^2, \tag{18}$$

and

$$\omega^{k-1} = \frac{\alpha_{k-1} - 1}{\alpha_k}, \tag{19}$$

with $\alpha_0 = 1$ and

$$\alpha_k = \frac{1 + \sqrt{4\alpha_{k-1}^2 + 1}}{2}. \tag{20}$$

Furthermore, let

$$\hat{\mathbf{U}}_{q,k-1}^{(m)} = \mathbf{U}_{q,k-1}^{(m)} + \omega_n^{k-1} \left(\mathbf{U}_{q,k-1}^{(m)} - \mathbf{U}_{q,k-2}^{(m)} \right), \tag{21}$$

and

$$\begin{aligned} \hat{\mathbf{G}}_{\mathbf{U}_q^{(m)}}^{k-1} = \frac{1}{n_q} & \left(\hat{\mathbf{U}}_{q,k-1}^{(m)} \mathbf{F}_{\mathbf{U}_q^{(m)}}^{k-1\,T} - \boldsymbol{X}_{q_{(m)}} \right) \mathbf{F}_{\mathbf{U}_q^{(m)}}^{k-1} \\ & + \alpha \left(\hat{\mathbf{U}}_{q,k-1}^{(m)} \Lambda_{\mathbf{w}_q} \mathbf{F}_{\mathbf{U}_q^{(m)}}^{k-1\,T} - \boldsymbol{Z}_{q_{(m)}} \right) \mathbf{F}_{\mathbf{U}_q^{(m)}}^{k-1} \Lambda_{\mathbf{w}_q}^T \\ & + \beta \left(\hat{\mathbf{U}}_{q,k-1}^{(m)} (\mathbf{I} - \Lambda_{\mathbf{w}_q}) \mathbf{F}_{\mathbf{U}_q^{(m)}}^{k-1\,T} - \boldsymbol{S}_{q_{(m)}} \right) (\mathbf{1} - \Lambda_{\mathbf{w}_q}) \mathbf{F}_{\mathbf{U}_q^{(m)}}^{k-1} \\ & + \gamma \left(\hat{\mathbf{U}}_{q,k-1}^{(m)} (\mathbf{I} - \Lambda_{\mathbf{w}_q}) - \hat{\mathbf{U}}_{q',k-1}^{(m)} (\mathbf{I} - \Lambda_{\mathbf{w}_{q'}}) \right) (\mathbf{I} - \Lambda_{\mathbf{w}_q}) \end{aligned} \tag{22}$$

be the gradient. Then we can derive the update based on [38]:

$$\mathbf{U}_{q,k}^{(m)} = \underset{\mathbf{U}_q^{(m)} \geq 0}{\operatorname{argmin}} \langle \hat{\mathbf{G}}_{\mathbf{U}_q^{(m)}}^{k-1}, \mathbf{U}_{q,k}^{(m)} - \hat{\mathbf{U}}_{q,k-1}^{(m)} \rangle + \frac{\mathcal{L}_{\mathbf{U}_q^{(m)}}^{k-1}}{2} \left\| \mathbf{U}_{q,k}^{(m)} - \hat{\mathbf{U}}_{q,k-1}^{(m)} \right\|_F, \tag{23}$$

which can be written in the closed form as

$$\mathbf{U}_{q,k}^{(m)} = \max \left(0, \hat{\mathbf{U}}_{q,k-1}^{(m)} - \hat{\mathbf{G}}_{\mathbf{U}_q^{(m)}}^{k-1} / \mathcal{L}_{\mathbf{U}_q^{(m)}}^{k-1} \right). \tag{24}$$

Similarly, let

$$\hat{\mathbf{w}}_{q,k-1} = \mathbf{w}_{q,k-1} + \omega^{k-1} \left(\mathbf{w}_{q,k-1} - \mathbf{w}_{q,k-2} \right), \tag{25}$$

and

$$\begin{aligned} \hat{\mathbf{G}}_{\mathbf{w}_q}^{k-1} = & \left(\hat{\mathbf{w}}_q \mathbf{F}_{\mathbf{w}_q}^{k-1\,T} - \boldsymbol{\mathcal{Z}}_q \right) \mathbf{F}_{\mathbf{w}_q}^{k-1} - \left((\mathbf{1} - \hat{\mathbf{w}}_q) \mathbf{F}_{\mathbf{w}_q}^{k-1\,T} - \boldsymbol{\mathcal{S}}_q \right) \mathbf{F}_{\mathbf{w}_q}^{k-1} \\ & - \sum_m \left(\mathbf{U}_{q,k-1}^{(m)\,T} ((\mathbf{1} - \hat{\mathbf{w}}_q) \mathbf{U}_{q,k-1}^{(m)} - (\mathbf{1} - \Lambda_{\mathbf{w}_{q'}}) \mathbf{U}_{q',k-1}^{(m)}) \right), \end{aligned} \tag{26}$$

where

$$\mathbf{F}_{\mathbf{w}_q}^{k-1} = \mathbf{U}_{q,k-1}^{(m)} \odot \mathbf{U}_{q,k-1}^{(m')} \odot \mathbf{U}_{q,k-1}^{(m'')}. \tag{27}$$

Let

$$\mathcal{L}_{\mathbf{w}_q}^{k-1} = \left\| \mathbf{F}_{\mathbf{w}_q}^{k-1\,T} \mathbf{F}_{\mathbf{w}_q}^{k-1} \right\|^2, \tag{28}$$

we can write the closed form of the update for $\mathbf{w}_q$

$$\mathbf{w}_{q,k} = \max \left(0, \hat{\mathbf{w}}_{q,k-1} - \hat{\mathbf{G}}_{\mathbf{w}_q}^{k-1} / \mathcal{L}_{\mathbf{w}_q}^{k-1} \right). \tag{29}$$

ALGORITHM 1: `PairFac` algorithm for discovering the shared and discriminative subspace from tensor pairs.

Input : original tensors $\boldsymbol{\mathcal{X}}_B$ and $\boldsymbol{\mathcal{X}}_A$, and R.
Output: $\{w_q\}$, $\{U_q^{(m)}\}$ for $q \in \{A, B\}$ and $m \in \{L, V, T\}$

Compute $\boldsymbol{\mathcal{Z}}_q$ and $\boldsymbol{\mathcal{S}}_q$ by Eq. 7 and Eq. 9, $\forall m$;
Randomly initialize $\mathbf{U}_{q,-1}^{(m)} = \mathbf{U}_{q,0}^{(m)}$ and set $\mathbf{w}_{q,-1} = \mathbf{w}_{q,0} = [\frac{\mathbf{1}}{R}]$, $\forall q$ and $\forall m$;
Set $\alpha_0 = 1$ and $k = 0$;
while *not converged* **do**
    $k = k + 1$;
    Compute $\mathcal{L}_{\mathbf{w}_q}^{k-1}$, $\mathcal{L}_{\mathbf{U}_q^{(m)}}^{k-1}$, and set ω^{k-1}, $\forall q$ and $\forall m$, according to Eq. 18, 28, 19;
    Compute $\hat{\mathbf{U}}_{q,k}^{(m)}$ and $\hat{\mathbf{w}}_{q,k}$, $\forall q$ and $\forall m$, according to Eq. 21, and 25;
    Update $\mathbf{U}_{q,k}^{(m)}$ and $\mathbf{w}_{q,k}$, $\forall q$ and $\forall m$, according to Eq. 23, and 29;
end

Algorithm 1 summarizes the above updating rules for solving Eq. 10 [2].

Convergence analysis We provide the convergence analysis of Algorithm 1. The convergence of alternating proximal gradient method is analyzed in [4].

LEMMA 1. *(Sufficient decrease property [6]). Let $f : \mathbb{R}^m \to \mathbb{R}$ be a continuously differentiable function with gradient $\bigtriangledown f$ assumed L_f-Lipschitz continuous and let $\sigma : \mathbb{R}^m \to (-\infty, +\infty]$ be a proper and lower semicontinuous function with $inf_{\mathbb{R}^m}\sigma > -\infty$. For any $t > L_f$ and $u \in dom\sigma$, define*

$$u^+ = arg\ \min_x \{\langle\ x - u, \bigtriangledown f(u)\rangle + \frac{t}{2}\|x - u\|^2 + \sigma\ (u)\}. \tag{30}$$

Then we have that

$$f(u) + \sigma(u) - (f(u^+) + \sigma(u^+)) \geq \frac{1}{2}(t - L_f)\|u^+ - u\|^2. \tag{31}$$

LEMMA 2. *Let $\Psi(\rho)$ be the objective function J_3 ,where $\rho = (\mathbf{U}_{q,k}^{(m)}, \mathbf{w}_{q,k})_{k\in\mathbb{N}}$ and $(\mathcal{L}_{\mathbf{U}_q^{(m)}}^k, \mathcal{L}_{\mathbf{w}_q}^k)_{k\in\mathbb{N}}$ are generated by our* `PairFac` *algorithm, we have that*

$$\Psi(\mathbf{U}_{q,k}^{(m)}, \mathbf{w}_{q,k}) - \Psi(\mathbf{U}_{q,k+1}^{(m)}, \mathbf{w}_{q,k}) \geq \frac{\mathcal{L}_{\mathbf{U}_q^{(m)}}^k}{2}\|\mathbf{U}_{q,k}^{(m)} - \mathbf{U}_{q,k+1}^{(m)}\|^2, \forall m, \forall q,$$

$$\Psi(\mathbf{U}_{q,k+1}^{(m)}, \mathbf{w}_{q,k}) - \Psi(\mathbf{U}_{q,k+1}^{(m)}, \mathbf{w}_{q,k+1}) \geq \frac{\mathcal{L}_{\mathbf{w}_q}^k}{2}\|\mathbf{w}_{q,k} - \mathbf{w}_{q,k+1}\|^2, \forall q$$

Proof. The above inequalities can be obtained by using Lemma 1.□

In the following we show that the value of $\Psi(\rho)$ monotonically decreases on the sequence $(\rho^k)_k \in \mathbb{N}$, which is generated by `PairFac`.

[2]Our codes are publicly available at https://github.com/picsolab/pairfac.

LEMMA 3. *Let $\Psi(\rho)$ be the objective function defined in J_3, where $\rho = (\mathbf{U}_{q,k}^{(m)}, \mathbf{w}_{q,k})$ and there exists $L > 0$ such that $\mathcal{L}^k_{\mathbf{U}_q^{(m)}} \geq L$ and $\mathcal{L}^k_{\mathbf{w}_q} \geq L$, then (i) The sequence $\{\Psi(\rho)\}_{k\in\mathbb{N}}$ is nonincreasing and for any $k \in \mathbb{N}$, there is a scalar $\beta > 0$ such that*

$$\Psi(\rho^k) - \Psi(\rho^{k+1}) \geq \beta \|\rho^k - \rho^{k+1}\|_F^2, \forall k \geq 0.$$

(ii) We have

$$\sum_{k=1}^{\infty} (\|\mathbf{U}_{q,k+1}^{(m)} - \mathbf{U}_{q,k}^{(m)}\|^2 + \|\mathbf{w}_{q,k+1} - \mathbf{w}_{q,k}\|^2) = \sum_{k=1}^{\infty} \|\rho^{k+1} - \rho^k\|^2 < \infty, \tag{32}$$

and therefore the sequence $\{\Psi(\rho)\}_{k\in\mathbb{N}}$ is bounded.

Proof. Adding the inequalities from Lemma 2, we have

$$\Psi(\mathbf{U}_{q,k}^{(m)}, \mathbf{w}_{q,k}) - \Psi(\mathbf{U}_{q,k+1}^{(m)}, \mathbf{w}_{q,k+1}) \geq \frac{\mathcal{L}^k_{\mathbf{U}_q^{(m)}}}{2} \|\mathbf{U}_{q,k}^{(m)} - \mathbf{U}_{q,k+1}^{(m)}\|^2 + \frac{\mathcal{L}^k_{\mathbf{w}_q}}{2} \|\mathbf{w}_{q,k} - \mathbf{w}_{q,k+1}\|^2. \tag{33}$$

In `PairFac`, the Lipschitz constants $\mathcal{L}^k_{\mathbf{U}_q^{(m)}} \geq L$, $\mathcal{L}^k_{\mathbf{w}_q} \geq L$. Therefore, we have

$$\frac{\mathcal{L}^k_{\mathbf{U}_q^{(m)}}}{2} \|\mathbf{U}_{q,k}^{(m)} - \mathbf{U}_{q,k+1}^{(m)}\|^2 + \frac{\mathcal{L}^k_{\mathbf{w}_q}}{2} \|\mathbf{w}_{q,k} - \mathbf{w}_{q,k+1}\|^2 \geq \frac{L}{2} (\|\mathbf{U}_{q,k}^{(m)} - \mathbf{U}_{q,k+1}^{(m)}\|^2 + \|\mathbf{w}_{q,k} - \mathbf{w}_{q,k+1}\|^2). \tag{34}$$

Combining inequality 33 and 34 yields the following

$$\Psi(\rho^k) - \Psi(\rho^{k+1}) \geq \frac{L}{2} \|\rho^k - \rho^{k+1}\|^2. \tag{35}$$

Hence with $\beta = \min\{L/2, 1/2\}$, we prove (i).

From Eq. 33 we obtain that the sequence $\{\Psi(\rho)\}_{k\in\mathbb{N}}$ is nonincreasing. Since Ψ is assumed to be bounded from below by *zero*, it converges to some real number $\bar{\Psi}$. Let N be a positive integer. Summing up all $k \geq 1$ for inequality 35, we have

$$\begin{aligned} \Psi(\rho^0) - \bar{\Psi} &\geq \Psi(\rho^0) - \Psi(\rho^N) \\ &\geq \frac{L}{2} \sum_{k=1}^{N} \|\rho^k - \rho^{k+1}\|^2 \\ &= \frac{L}{2} \sum_{k=1}^{N} (\|\mathbf{U}_{q,k}^{(m)} - \mathbf{U}_{q,k+1}^{(m)}\|^2 + \|\mathbf{w}_{q,k} - \mathbf{w}_{q,k+1}\|^2) \end{aligned} \tag{36}$$

Taking the limit as $N \to \infty$, we prove the assertion (ii).

Based on this lemma, we then provide a convergence result of algorithm 1 under certain assumptions. Let $\rho = (\mathbf{U}_q^{(m)}, \mathbf{w}_q)$. A point ρ satisfies the KKT-condition for the solution to Eq. 10 if

$$\begin{aligned}
&\frac{1}{n_q}\mathbf{U}_q^{(m)} \star \Bigg(\left(\mathbf{U}_q^{(m)}(\mathbf{U}_q^{(m')} \odot \mathbf{U}_q^{(m'')})^T - \boldsymbol{X}_{q_{(m)}}\right)(\mathbf{U}_q^{(m')} \odot \mathbf{U}_q^{(m'')}) \\
&+\alpha \left(\mathbf{U}_q^{(m)}\Lambda_{\mathbf{w}_q}(\mathbf{U}_q^{(m')} \odot \mathbf{U}_q^{(m'')})^T - \boldsymbol{Z}_{q_{(m)}}\right)(\mathbf{U}_q^{(m')} \odot \mathbf{U}_q^{(m'')})\Lambda_{\mathbf{w}_q}^T \\
&+\beta \left(\mathbf{U}_q^{(m)}(\mathbf{I} - \Lambda_{\mathbf{w}_q})(\mathbf{U}_q^{(m')} \odot \mathbf{U}_q^{(m'')})^T - \boldsymbol{S}_{q_{(m)}}\right)(\mathbf{U}_q^{(m')} \odot \mathbf{U}_q^{(m'')})(\mathbf{I} - \Lambda_{\mathbf{w}_q})^T \\
&+\gamma \left(\mathbf{U}_q^{(m)}(\mathbf{I} - \Lambda_{\mathbf{w}_q}) - \mathbf{U}_{q'}^{(m)}(\mathbf{I} - \Lambda_{\mathbf{w}_{q'}})\right)(\mathbf{I} - \Lambda_{\mathbf{w}_q})\Bigg) = 0, \\
&\mathbf{w}_q \star \Bigg(\left(\mathbf{w}_q {\mathbf{F}_{\mathbf{w}_q}}^T - \boldsymbol{\mathcal{Z}}_q\right)\mathbf{F}_{\mathbf{w}_q} - \left((\mathbf{1} - \mathbf{w}_q){\mathbf{F}_{\mathbf{w}_q}}^T - \boldsymbol{\mathcal{S}}_q\right)\mathbf{F}_{\mathbf{w}_q} \\
&- \sum_m \left({\mathbf{U}_q^{(m)}}^T((\mathbf{1} - \mathbf{w}_q)\mathbf{U}_q^{(m)} - (\mathbf{1} - \Lambda_{\mathbf{w}_{q'}})\mathbf{U}_{q'}^{(m)})\right)\Bigg) = 0,
\end{aligned} \tag{37}$$

where $\star$ denotes component-wise product and $\mathbf{F}_{\mathbf{w}_q}$ is defined in Eq. 27.

THEOREM 1. *Suppose the sequence* $\{\rho = (\mathbf{U}_{q,k}^{(m)}, \mathbf{w}_{q,k})\}$ *generated by algorithm* `PairFac` *is uniformly away from zero, i.e., there exists* $\mathcal{L} > 0$ *such that* $\mathcal{L}^k_{\mathbf{U}_q^{(m)}} \geq L$ *and* $\mathcal{L}^k_{\mathbf{w}_q} \geq L$, $\forall q$ *and* $\forall m$. *Then any limit point of* $\{\rho\}$ *satisfies the KKT-conditions 37.*

The proof of Theorem 1 is provided in Appendix.

4.4 Parallel Implementation

In this section, we provide a scalable implementation for the `PairFac` algorithm. Our method is based on FlexiFaCT [5], which is a MapReduce algorithm for PARAFAC and coupled PARAFAC decompositions.

The key idea of FlexiFaCT is to split the tensor data into multiple blocks, each of which is further split into smaller blocks with no shared rows or columns. Given the complex nature of tensorial computation, researchers have initiated efforts in devising more efficient algorithms for tensor computations, e.g., GigaTensor [13], FlexiFaCT [5], MET [16], Turbo-SMT [28], and Haten2 [12]. We adopt the scheme introduced in [5] due to its simplicity in implementation as well as its ability for coupled tensor matrix factorization. The parallelization implementation involves three steps:

Step 1: Blocking for Parallelization. This step is to partition the data tensors into certain blocks so that each block could run in parallel. Following [5], we term one set of independent blocks in the corresponding tensor a *stratum*, and then we denote the number of blocks in each stratum by d. To have full coverage of the whole tensor, we require d^2 strata. For a stratum s we have blocks $P_i^{(s)}$ for $i = 0...d-1$. Let each block P be the tensor that contains all data observations in (b_i, b_j, b_k) where b_i, b_j, b_k are ranges in I, J, and K: $b_i = (i\lceil I/d\rceil, (i+1)\lceil I/d\rceil)$, $b_j = (j\lceil I/d\rceil, (j+1)\lceil I/d\rceil)$, $b_k = (k\lceil I/d\rceil, (k+1)\lceil I/d\rceil)$. With this we define the blocks for stratum s as

$$\begin{aligned}
P_i^{(s)} &= (b_i, b_{j_{s,i}}, b_{k_{s,i}}) \\
j_{s,i} &= (i+s) \bmod d \\
k_{s,i} &= \lfloor (i + s/d) \rfloor \bmod d,
\end{aligned}$$

for $i = 0...d-1$.

Step 2: Parallelizing the Computation. We partition the original tensors as well as the three auxiliary tensors with the same schema so that $P_i^{(s)}$ denotes the same block across different tensors. With this partition schema, we run the strata sequentially, but for each stratum we compute the gradient with respect to $\mathbf{U}_q^{(m)}$ by Eq. 22 and to $\mathbf{w}_q$ by Eq. 26 based on sparse tensors constructed from (b_i, b_j, b_k) in parallel on d machines.

Step 3: Gradient Summation. Now we have temporary gradient values computed by each machine. These values are sent the partial gradients to the centralized master server. Lastly, the final gradients in Eq. 22 and Eq. 26 are the summation of all the partial gradients.

In practice, step 1 can be regarded as a prepossessing step to index the observations in the tensors to certain blocks for parallelization. We can run step 2 and 3 repeatedly, iteratively updating $\mathbf{U}_q^{(m)}$ and $\mathbf{w}_q$, $\forall m$ and $\forall q$, until the algorithm converges.

5 EVALUATION

In this section, we provide the evaluation of `PairFac` based on a synthetic dataset. Section 5.1 describes the synthetic dataset, while Section 5.2 illustrates the exemplary output of `PairFac`. Section 5.3 provides the quantitative comparison with existing baselines. Since `PairFac` outputs components with a list of associated weights instead, Section 5.4 discusses several approaches to identify the common and discriminative components based on the weights. Finally, in Section 5.5, we provide guidance on the parallelized implementation of `PairFac`.

5.1 Synthetic Data Setup

The synthetic dataset generation aims to provide multidimensional datasets that share some signals in common. To this end, we want to generate two three-way tensors $\boldsymbol{\mathcal{X}}_B \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and $\boldsymbol{\mathcal{X}}_A \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ according to the equation $\boldsymbol{\mathcal{X}}_B = \sum_{r=1}^{R} \mathbf{U}_{B,r}^{(L)} \circ \mathbf{U}_{B,r}^{(T)} \circ \mathbf{U}_{B,r}^{(V)}$ and $\boldsymbol{\mathcal{X}}_A = \sum_{r=1}^{R} \mathbf{U}_{A,r}^{(L)} \circ \mathbf{U}_{A,r}^{(T)} \circ \mathbf{U}_{A,r}^{(V)}$, where $\boldsymbol{\mathcal{X}}_B$ and $\boldsymbol{\mathcal{X}}_A$ share the first K components among the total R components in the first factor matrix and have exactly the same columns in the second and third factor matrices. K is a parameter that controls the extent to which the two generated tensors are similar to each other. Our generation rules of the synthetic dataset follow the idea in [14]. The shared part in the first factor matrix are generated as:

$$\mathbf{U}_{C,r}^{(L)} = \begin{cases} 1, sr \leqslant m < s(r+1), \\ 0, \text{otherwise}, \end{cases}$$

where $s = I/\ (R + (R - K))$, r is the column index for each matrix and m is the row index. We generate the discriminative parts in the first factor matrix as:

$$\mathbf{U}_{D:B,r}^{(L)} = \begin{cases} 1, sK + sr \leqslant m < sK + s(r+1), \\ 0, \text{otherwise}, \end{cases}$$

and

$$\mathbf{U}_{D:A,r}^{(L)} = \begin{cases} 1, sR + sr \leqslant m < sR + s(r+1), \\ 0, \text{otherwise}. \end{cases}$$

In addition, each row of $\{\mathbf{U}^{(T)}\}$ and $\{\mathbf{U}^{(V)}\}$ is set to be a unit vector with only one non-zero entry at a randomly selected dimension. We further add sparse Gaussian noise $\mathcal{N}(0, \sigma^2)$ with different levels of variance to 20% of the entries in $\mathbf{U}_B^{(L)}$ and $\mathbf{U}_A^{(L)}$.

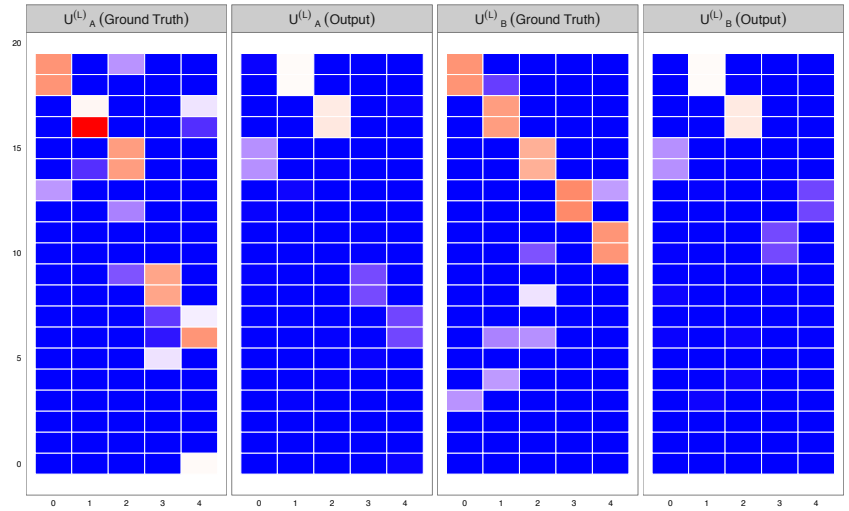


Fig. 2. **Illustration of the output from our approach.** We reorder the components of each output factor matrix by its associated weight in ascending order from left to right. The weight vector $\mathbf{w}_A$ = [.0001, .0000, .0000, .4874, .5124] and the weight vector $\mathbf{w}_B$ = [.0000, .0003, .0013, .4877, .5107].

5.2 Algorithm Output Illustration

We first provide the illustration of the output from our approach with the synthetic dataset generated by setting $I_1 = I_2 = I_3 = 20$, $R = 5$, $K = 3$, $\sigma^2 = 0.1$ and $\alpha = 10^{-5}$, $\beta = 2$, and $\gamma = 10^{-4}$. Fig. 2 shows an illustrative example of the factor matrices obtained from our method in comparison with the ground truth factor matrices. Each column of the output factor matrices is associated with a discriminative score (i.e., $\mathbf{w}_q$ as in Eq. 11). To reiterate, a higher score represents a greater level of discriminativeness for the corresponding component in comparison with the components in the factor matrix in the second tensor. As explained in Eq. 11, the value of each $\mathbf{w}_q$ reflects the extent to which its corresponding component contributes to the reconstruction of the "differing" part between the two (before and after) tensors. We observe that our method nicely segments each output factor into two parts based on the learned weights. The weights of the common components are almost zero while the discriminative components contribute equally to the overall discriminative power. One outstanding property of this model compared to our prior work [37] is its ability to align the similar components in the corresponding order. For instance, we observe that the first three columns are common components in $\mathbf{U}_A^{(L)}$(Output) and $\mathbf{U}_B^{(L)}$(Output). Among these three columns, the first columns in these two matrices correspond to one common component (the third column) in $\mathbf{U}_A^{(L)}$(Ground Truth) and $\mathbf{U}_B^{(L)}$(Ground Truth). Similarly, we could find that the second and the third columns in the output matrices concur with themselves and can also find their matches in the ground truth matrices.

5.3 Comparisons with Baselines

As discussed in Section 4, there are three existing models that we adopt for comparisons, including CDNTF [20], our extension of RSJNMF [10] to RSJNTF, and our extension of SDCDNMF [14] to SDCDNTF.

5.3.1 Baselines. We include three baselines and one modification of our method for comparative studies:

- CDNTF [20] takes an input K and splits the factor matrix into K common components and $(R-K)$ discriminative components by solving Eq. 1 with multiplicative updating rules.
- RSJNTF is our tensor extension of RSJNMF [10]. It also requires the number of common components K as input K and is based on a similar framework with CDNTF where additional mutually orthogonal constraints on the common and discriminative components are added. We develop multiplicative updating rules to solve Eq. 2.
- SDCDNTF is our tensor extension of SDCDNMF [14], which also requires K as input. It can be classified under the same framework as RSJNTF, where there is a relaxation to the constraints on the shared components. We extend the block coordinate descent framework to SDCDNTF to solve Eq. 4.
- `PairFac` does not require the specification of K. Instead, it generates two weight vectors that represent the discriminative scores for each of its components.

5.3.2 Evaluation Metrics. To quantitatively evaluate the performance of our proposed approach in comparison with existing literature, we use three measures, namely, (a) the relative reconstruction error, (b) the quality of the recovered discriminative components and (c) the quality of the recovered common components. To measure the quality of the reconstruction, we compute the relative reconstruction error as:

$$\frac{1}{2}\left(\frac{\left\|\boldsymbol{\mathcal{X}}_B - [\![\mathbf{U}_B^{(L)}, \mathbf{U}_B^{(T)}, \mathbf{U}_B^{(V)}]\!]\right\|^2}{\|\boldsymbol{\mathcal{X}}_B\|^2} + \frac{\left\|\boldsymbol{\mathcal{X}}_A - [\![\mathbf{U}_A^{(L)}, \mathbf{U}_A^{(T)}, \mathbf{U}_A^{(V)}]\!]\right\|^2}{\|\boldsymbol{\mathcal{X}}_A\|^2}\right).$$

The quality of the recovered discriminative part of the factor matrix is computed as the similarity between the output factor matrix and the ground truth factor matrix: $sim_D(\mathbf{U}, \bar{\mathbf{U}}) = \frac{1}{R-K}\sum_{r>K}^{R} \cos(\mathbf{U}_r, \bar{\mathbf{U}}_r) = \frac{\mathbf{U}_r \cdot \bar{\mathbf{U}}_r}{\|\mathbf{U}_r\|\|\bar{\mathbf{U}}_r\|}$, where $\mathbf{U}_r$ is the r-th discriminative component in the ground truth factor matrix and $\bar{\mathbf{U}}_r$ is the output of the r-th discriminative component. Because there is an ambiguity in the column ordering [1], we try out all possible permutations of $R-\kappa$ components and compute the maximum similarity. Furthermore, we compute the maximum similarity score of the common components as: $sim_C(\mathbf{U}, \bar{\mathbf{U}}) = \frac{1}{R}\sum_{r\leqslant K}^{R} \cos(\mathbf{U}_r, \bar{\mathbf{U}}_r) = \frac{\mathbf{U}_r \cdot \bar{\mathbf{U}}_r}{\|\mathbf{U}_r\|\|\bar{\mathbf{U}}_r\|}$.

5.3.3 Experiment Setup. Following the setup introduced in section 5.1, we generate another synthetic dataset by setting $I_1 = 100$, $I_2 = 10$, $I_3 = 20$, $\sigma^2 = 0.5$, $R = 10$, and $K = 5$. For SDCDNTF, we experiment with α and $\beta \in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. For RSJNTF, following [10], we set a super parameter α in the same range. Finally, for `PairFac`, we set α and $\beta \in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$, and $\gamma \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$. We plot the average reconstruction error versus the average similarity score on the discriminative components as well as on the common components from 30 runs of each method on every set of parameters.

5.3.4 Results. Fig. 3 presents the comparison of the various methods from 30 independent trials for each combination of parameter settings. The x-axis and y-axis show the quality of recovered discriminative components and the quality of recovered common components. Each point represents the average result of 30 runs. The size of each point is proportional to the reconstruction error. We observe that `PairFac` has comparable reconstruction quality with that of SDCDNTF. We also notice that most of the points from `PairFac` lay on the top-right region in the figure, exhibiting higher quality in both recovered discriminative and common

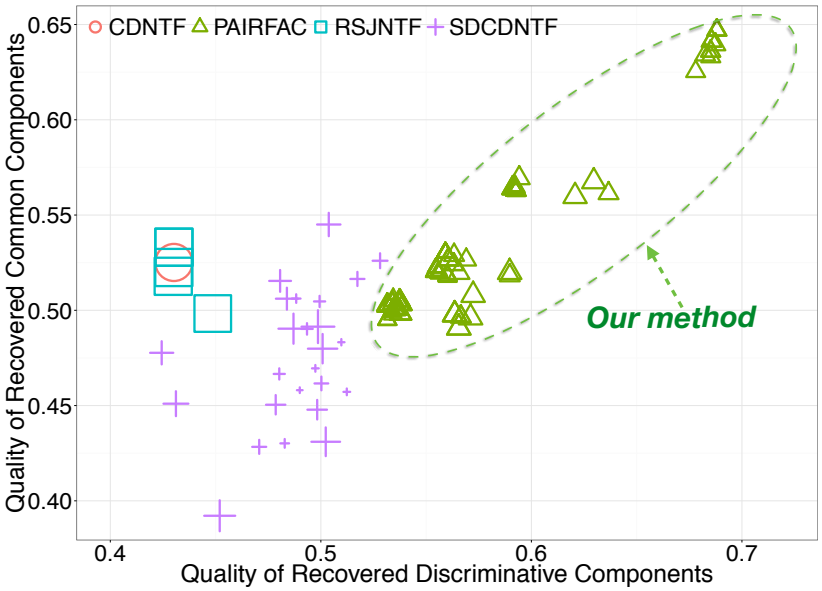


Fig. 3. **Comparison of our approach with existing methods.** Each point represents the average score of 30 runs for each combination of the parameter setting. The size of points represents the reconstruction error.

components. We conducted additional experiments on cases where the data have a varying number of modes that are similar or different. Our results show that `PairFac` consistently achieves better recovery quality in both the common and discriminative components. The results are included in the appendix.

5.4 Identification of Common and Discriminative Patterns

`PairFac` learns the ranked components based on their discriminative scores. Components that have higher similarities associate with low weights. In this section, we show how to identify common and discriminative patterns.

Given a vector of ranked numerical values in the range of $(0, 1)$ generated by `PairFac`, the problem of identifying common and discriminative components is equivalent to searching for a proper threshold θ, such that components with $w < \theta$ would be regarded as common components, while the rest can be regarded as discriminative components. We experimented with four approaches for the selection of a cutoff threshold:

Fixed threshold. The simplest approach is to define a fixed threshold, regardless how many common components are in the tensors. We can set $\theta = \frac{1}{R}$, which essentially makes the assumption that every component (from the R components in total) has equal probability of being discriminative.

Largest Difference. We could also define θ as the maximum difference between two consecutive (ordered) weights.

Two Clusters. The weights learned from `PairFac` tend to fall into two natural groups. Therefore small weights and large weights are likely to be separated by a simple one-dimensional clustering with two clusters.

Bimodal Density. Given that the weights tend to fall into two natural groups, we could model the distribution of weights using a kernel density function and set θ equal to the local minimum of the area between two peaks.

5.4.1 Experimental Setup. In this experiment, we aim into evaluating the number of common components identified by different heuristics. Following the setup introduced in section 5.1, we generated another synthetic dataset by setting $I_1 = 100$, $I_2 = 10$, $I_3 = 20$, $\sigma^2 = 0.5$, $R = 10$, and $K \in \{1, 2, 4, 5, 6, 7, 8, 9\}$. We perform five runs with each value of K and reported the run with the best results.

5.4.2 Results. In Fig. 4, we present the number of common components identified based on the value θ defined by the different heuristics aforementioned. Ideally, for a *perfect* choice of θ, we expect the results to lay on the line $y = x$. Of the four approaches attempted, we observe that the value of θ defined by *bimodal density* and *largest differences* are the closest to the *optimal* solution.

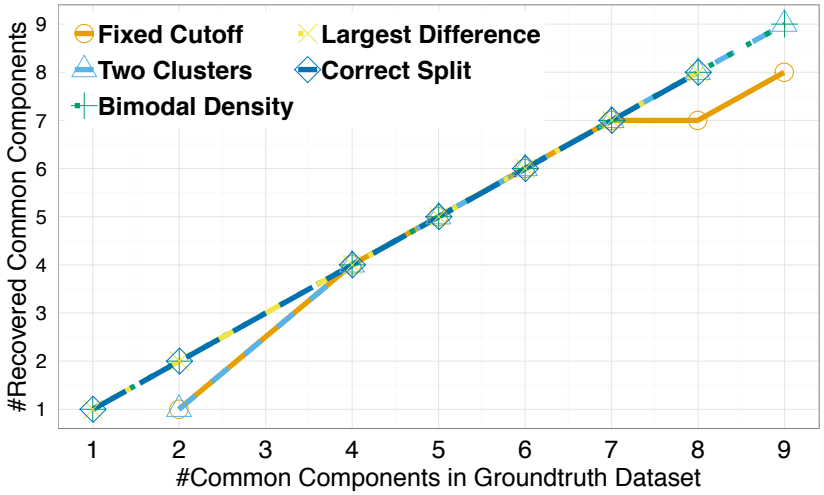


Fig. 4. **Number of common components identified by different heuristic approaches.** Dark blue line with diamond-shaped points denotes the perfect split between the common and discriminative components; the cutoff defined by Bimodal Density (green line with cross-shaped points) has the closest split with the optimal split.

5.5 Parameter Sensitivity

In our approach, parameters α and β control the weight placed on identifying the discriminative or common components, and γ controls the extent to which common components could be aligned together. In this section, we evaluate the sensitivity of our approach with regards to these parameters.

5.5.1 Experimental Setup. We follow the same experimental setup as introduced in Section 5.2 for `PairFac`. For each experiment, we vary one of the parameters α, β, and γ in `PairFac`, while keeping the remaining parameters constant.

5.5.2 Evaluation Metrics. In Eq. 10, we introduced auxiliary tensors to capture the common as well as the unique parts of both tensors. α and β control the importance of the discriminative and common components respectively. As `PairFac` learns the discriminative weights of each component we label them in order to classify them as common or unique. During this process, we need to identify a cutoff point for the (ranked) weights. The components that have discriminative power higher than this cutoff would be regarded as unique patterns to each tensor. Section 5.4 suggests that the distribution of weights follow a bi-modal distribution and the local minimum of the pit is the optimal cutoff for the split. Hence, we separate the components using a bimodal distribution for the weights. To measure the extent to which the bimodal distribution could reach a clear separation, we compute the bimodal separation index [41]. Furthermore, the third term in Eq. 10 enforces that similar components should be aligned together. γ is expected to control the degree to which the factorization should be constrained by the component similarity regularization.

5.5.3 Results. For evaluating the sensitivity of α and β, we calculate their impact on the separability and the relative reconstruction error, with a fixed value of γ. For evaluating the sensitivity of γ, we calculate its effect on the similarity of the common components and the relative reconstruction error, with α and β fixed. We run `PairFac` with each parameter setting for 30 runs and report the average measures with standard errors.

Effect of auxiliary tensors. We vary the weight of factorizing the auxiliary tensors by setting α and $\beta \in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$, and $\gamma = 1$. Fig. 5 shows the average relative reconstruction given different settings of α and β. The results suggest that with the increase of the weight for factorizing the auxiliary tensors, the reconstruction quality degrades. One exception is shown in Fig. 5 (a), where the relative reconstruction error decreases while α becomes larger. However, we expect the factorization quality would eventually go up with larger α values. Fig. 6 shows the average separability with different parameter settings of α and β when γ is fixed to 1. When β is fixed, the separability becomes larger when α increases, except when β is equal to 1. When α is fixed, we observe that the separability decreases first and then increases.

Effect of column regularization. We vary the weight of enforcing the column similarity regularization by setting $\gamma \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$ and $\alpha = \beta = 10^{-6}$. Fig. 7 (a) shows the average relative reconstruction error with different values of γ. As we observe, when $\gamma \leqslant 10^{-2}$, the column regularization barely draws any impacts on the reconstruction error, although we have gains in the similarity between the resultant common components as shown in Fig. 7 (b). When $\gamma \geq 10^{-2}$, the reconstruction error first decreases and increases again, while the similarity scores seem to continue rising with the increase of γ. It is possible that a reasonably large choice of γ can give rise to the importance of column regularization in the factorization steps. However, when γ is set to be too large, the factorization result would bias towards making excessive agreements between the common components, while losing its quality on the true discriminative patterns.

To summarize, we demonstrated that, in practice, the "relative reconstruction error" can be used to observe the appropriate range of the parameter settings. For example, in our experiments, we found that the reconstruction errors are relatively stable for a wide range of α and β values, except for a very large value in either of the two parameters (Fig. 5 and 6). γ controls the level of "similarity" in common components, which is a parameter that allows the algorithm to adapt to different application scenarios (Fig. 7(b)). A too large value of γ (too much tolerance of "similarity") may degrade the reconstruction results, which can be easily discovered from plotting the reconstruction error against γ (Fig. 7(a)).

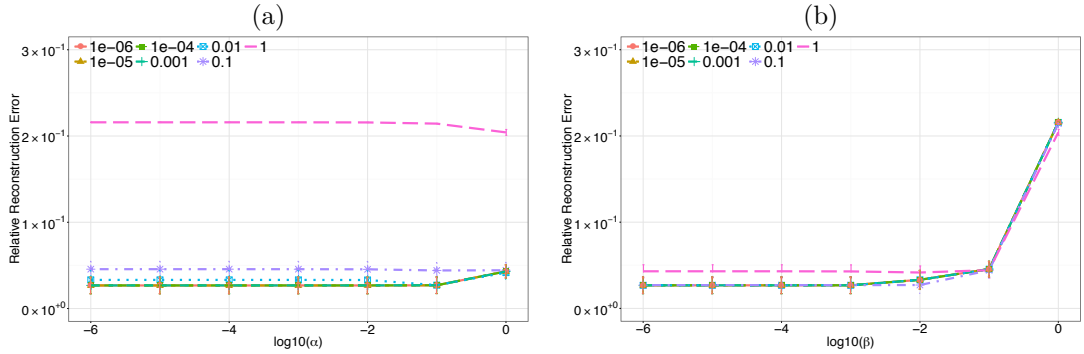


Fig. 5. **(a) α vs. relative construction error (b) β vs. relative reconstruction error.** Different lines represent the settings of different α or β values. (a) shows that as α goes large, we have higher reconstruction errors except when $\beta = 1$; (b) shows that as β larger tend to lead to higher reconstruction errors.

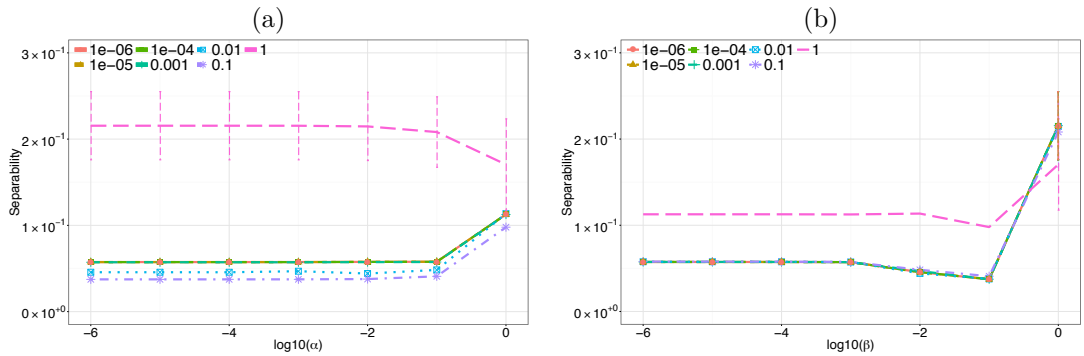


Fig. 6. **(a) α vs. separability (b) β vs. separability.** Different lines represent the settings of different α or β values. (a) shows that as α goes larger, the separability becomes larger except when $\beta = 1$; (b) shows that as β becomes larger, however followed by in increasing trend.

5.6 Scalability

In this section, we provide the scalability analysis of our proposed method in terms of parallel and non-parallel implementations. The purpose of the experiments on the synthetic data is to demonstrate the run-time efficiency of the proposed method as well as the speedup of the parallelization strategy. To understand how different tensor properties affect the computation time, we perform a set of experiments with varying conditions. There are three sets of parameters involved in this analysis: observations N is the number of nonzero elements in the tensor; dimensionality I is the size of a mode; and rank R is the minimal number of rank one tensors, which generate the tensor as their sum.

5.6.1 Experiments. We construct two synthetic tensors following the dataset setup introduced in Section 5.1, with a varying set of parameters to test the scalability with respect to each of them. To this end, we fix two of three parameters N, I, R and vary the remaining one. We conducted three experiments for the sake of validating the scalability of our method

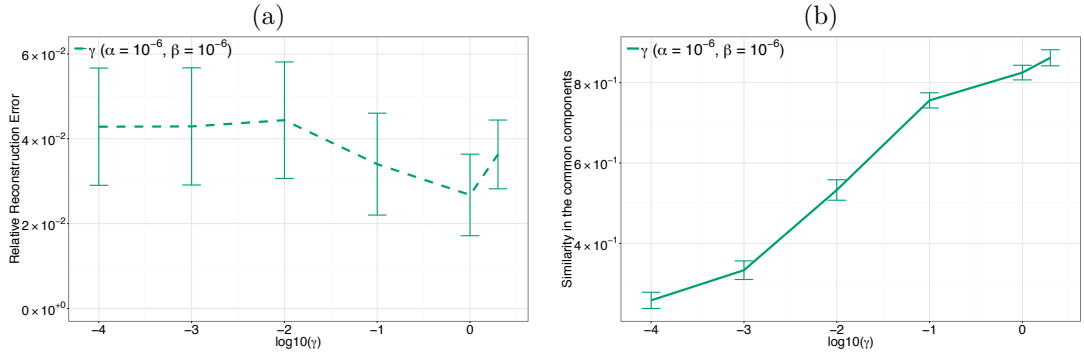


Fig. 7. **(a) γ vs. relative reconstruction error (b) γ vs. similarity in the common components.** (a) shows that as γ goes large, the relative reconstruction error decreases and then goes up after taking certain larger values; (b) shows that as γ increases, we have higher similarities in the common components.

concerning the number of observations, the dimensionality of the tensors, and the tensor rank. Unless otherwise stated, we set the convergence criteria as either reaching 10,000 iterations or the relative reconstruction is below 10^{-4}.

Observations. We generate a synthetic dataset with $I_1 = I_2 = I_3 = 1000$, $R = 30$, $K = 10$, following section 5.1. Then we take the top N largest elements from each tensor to construct the sparse tensor, where N varies in the range of $\{10^2, 10^3, 10^4, 10^5, 10^6\}$. We set $R = 10$ as the number of components after the factorization. In Fig. 8 (a), we show the running time of our algorithm against the number of observations.

Dimensionality. We generate a synthetic dataset with $I_1 = I_2 = I_3 \in \{400, 500, 600, 700, 800\}$, $R = 30$, $K = 10$. Then we take the top 10^4 largest elements from each tensor to construct the sparse tensors and set $R = 10$ as the number of components after the factorization. In Fig. 8 (b), we show the running time of our algorithm against the dimensionality.

Rank. We generate a synthetic dataset with $I_1 = I_2 = I_3 = 1000$, $R = 30$, $K = 10$. Then we take the top 10^5 largest elements from each tensor to construct the sparse tensors and set R in the range of $\{10, 20, 30, 40, 50\}$ as the number of components after the factorization. In Fig. 8 (c), we show the running time of our algorithm against the rank of the tensor decomposition.

5.6.2 Results. The results show that the running time of `PairFac` scales reasonably well with the growth of the number of observations, the dimensionality of the tensors, and the number of components. Furthermore, with the stratum split mechanism introduced in Section 4.4, we could reach better scalability with the help of multi-threading processing of `PairFac`. The yellow lines in Fig 8 show the running time with two threads in comparison to single-threaded `PairFac`.

6 CASE STUDIES

In this section, we illustrate the application of our method in two case studies, which showcases the effects of specific events in the urban space, including the Paris terrorist attacks and the Thanksgiving holiday weeks comparison in New York City (NYC).

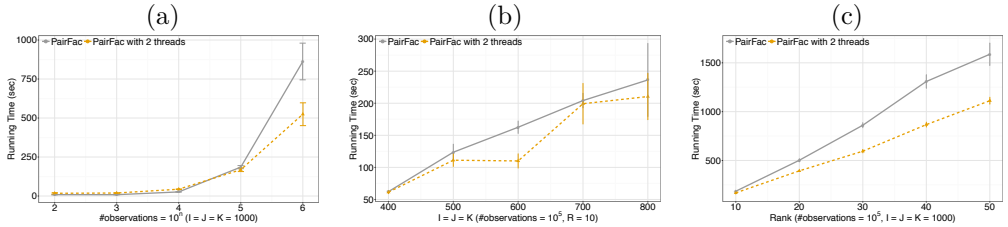


Fig. 8. **(a) number of observations vs. Running Time (b) Dimensionality vs. Running Time and (c) Rank vs. Running Time.**

Type of Data	Dimensions extracted	Volume of raw data extracted
Traffic Sensor databases	Location, Time	10,915,272 hourly occupancy rate from 2,885 road sensors
Check-ins and POI database	Location, Time, Activity	86,033 check-ins with 15,375 POI information
Geo-tagged Tweets	Location, Time	121,631 tweets

Table 2. Data sources used in the case study of Paris terrorist attacks.

6.1 Paris Attacks

In this section, we use `PairFac` to analyze the effects of the Paris terrorist attacks in the surrounding urban space. In our previous study, we investigated the immediate impact of urban mobility in the following week of the attacks. In this study, we collect Twitter check-ins and traffic sensor data in the month following the attacks from the Paris area and apply our approach to study the long-term impacts on urban mobility.

6.1.1 Dataset. Table 2 summarizes the data sources we used for our case study. The first dataset is the geo-tagged tweets from Paris collected through the Twitter API between the period of Oct 16th, 2015 and Dec 18, 2015. The region is defined by a rectangle boundary [3] that covers the Paris area. 121,631 geo-located tweets were extracted during the period covered. The second dataset includes approximately 10.9 million records of traffic sensor data [7]. It provides the hourly occupancy rate of 2,889 road segments in the area of Paris and covers the same period as above. Our third dataset is from Foursquare collected by Yang et al. [39] and it contains 86,033 check-ins from 15,375 POIs in the area of Paris between April 2012 and September 2013.

6.1.2 Case Study Setup. In our previous study, we used grid-cell based city partition to study the immediate impact of the terrorist attacks. We constructed three-mode tensors, where the three dimensions are location, time, and venue type, respectively. While the spatial locations can be represented via a two-dimensional variable, e.g., (x, y) or (latitude, longitude), they can also be represented as a list of locations indexed by the two-dimensional variable. We use the latter representation in our experiment to facilitate the interpretation of discovered impact in terms of "location mode" and to compare it with other modes. In our case studies, we used the neighborhoods to construct a list of locations as one mode in the input tensors, where each entry in the location dimension represents one neighborhood

[3] N 48° 54′ 32.6118″, E 2° 24′ 33.7104″, N 48° 48′ 56.361″, E 2° 14′ 36.7794″.

location. We extract 80 quartiers from 20 arrondissements in Paris as the possible values of the location dimension. For the temporal dimension, we segment a week into $24 \times 7 = 168$ hourly intervals. Finally, for the venue dimension, we extract the nine primary categories in the Foursquare venue hierarchy that includes *Professional & Other Places* (POP), *Travel & Transport* (TT), *Food* (F), *Outdoors & Recreation* (OR), *Nightlife Spot* (NS), *Shop & Service* (SS), *Residence* (R), *Arts & Entertainment* (AE), and *College & University* (CU). For the data tensor of geo-tagged tweets, we first construct a matrix LT, where LT_{ij} is the number of geo-tagged tweets that fall in the i-th district at the j-th hour in the week. Similarly, we construct the LT matrix based on the traffic sensor data, where LT_{ij} is the average occupancy rate in i-th district at the j-th hour in the week. Then, we construct a matrix FTV, where FTV_{ijk} is the probability of Foursquare check-ins in the k-th venue category that falls in the i-th district at the j-th hour in the week. Thus, for each cell at a given hour in the week, we know from the matrix FTV the probability distribution of activities over the nine categories. Finally, the entries in the data tensor are computed as:

$$\mathcal{X}_{ijk} = \frac{LT_{ij} \times FTV_{ijk}}{\sum_{ijk} \mathcal{X}_{ijk}}, \tag{38}$$

for both $\boldsymbol{\mathcal{X}}_B$ and $\boldsymbol{\mathcal{X}}_A$. $\boldsymbol{\mathcal{X}}_B$ contains the normalized aggregated values over four weeks between Oct. 16th, 2015 (Friday) and Nov. 12th, 2015, and $\boldsymbol{\mathcal{X}}_A$ is constructed based on the normalized values in the following month, between Nov. 20th, 2015 (Friday) and Dec. 18th, 2015.

In our study we set $\alpha = \beta = 10^{-8}$, $\gamma = 5 \times 10^{-7}$ for social media dataset and $\gamma = 10^{-7}$ for traffic sensor dataset. Finally we set $R = 20$ for both datasets.

6.1.3 Results. The advantage of `PairFac` is that it aligns the respective components of each tensor which share high similarities. This is realized through the fact that the output of `PairFac` is the mobility components as ranked by their associated discriminative scores, with similar components sharing similar scores. It is therefore straightforward to identify the common patterns as well as those discriminative ones. In the following, we pick two common patterns and one discriminative pattern from each dataset to illustrate the advantage of our proposed `PairFac` method. We first show the patterns from the geo-tagged tweets dataset, followed by the ones from the traffic sensors.

Patterns from social media data: Since the *largest difference* method has been shown to best find the split the patterns as in Section 5.4, we use it to separate the common components and the discriminative components in all case studies. There are 19 pairs of common components with small discriminative scores (M = .032, SD = .034) and one set of discriminative components with discriminative scores as .38 and .39, respectively. Below we show several interesting patterns among them all:

Common Pattern 1. Fig. 9 shows the 3rd component one month before (with discriminative score as .0035) and one month after (with discriminative score as .0038) the Paris attacks. We observe that the patterns from each tensor are virtually identical in all three dimensions. This set of patterns primarily corresponds to the activities in professional places. The time usage of this pattern typically falls during the daytime, although we observe a spike of activities on Thursday nights. This might be due to the small portion of nightlife activities mixed in this pattern. The pattern is heavily geographically distributed in the 16th arrondissement, where four Fortune Global 500 companies (PSA Peugeot Citroën, Kering, Lafarge, and Veolia) have their headquarters, which might explain the periodical distribution of professional workplaces activities.

Fig. 9. **Common Pattern 1 from social media data.** $3rd$ component before the attacks and $3rd$ component after the attacks. Two maps show the probability distribution of check-ins in different neighborhoods of Paris before (right) and after (left), where dark red (right) stands for a higher probability. The bottom-left figure shows the distribution of traffic over the week (24×7), where blue lines represent the distribution before the attacks and the red lines represent the one after. The bottom-right figure features the distribution of check-ins over different types of venues (defined in section 6.1.2).

Fig. 10. **Common pattern 2 from social media.** $14th$ component before the attacks and $14th$ component after the attacks.

Common Pattern 2. Fig. 10 shows the 14th component one month before (with discriminative score as .026) and one month after (with discriminative score as .053) the attacks. We see that the patterns from each tensor are almost identical, especially in their location and

Fig. 11. **Unique patterns from social media data.** 20th component before the attacks and 20th component after the attacks.

activity distribution. The time associated with this pattern starts from the morning and keeps active for almost the entire day. It is interesting to note that on Sundays, Parisians tend to start this pattern late and then gradually increase its usage. We observe the most geographically highlighted areas are the one that is very close to the 11th arrondissement, which is regarded as the hub of new food scene[4]. Other areas include the 8th and 10th arrondissement are also among the top three popular food places.

Unique Patterns. Fig. 11 shows the 20th component one month before (with discriminative score as .38) and 20th component from one month after (with discriminative score as .039) the attacks. We select these two as they share similar time distribution, along with similar location distribution, while their associated time of the week is different. This set of patterns features the activities around outdoor recreations. The area associated with these components is at the upper corner of 19th arrondissement, which is featured by Parc de la Villette, the third largest park in Paris. This could explain why the activity is centered around the outdoor recreations and the time mostly focuses on the second half of the days or over the weekends. We notice that before the attacks, the time distribution follows a fairly periodical pattern, with activities mostly taking place during the day-time and then shifting to afternoons or nights during the weekends. However, after the attacks, the volume of activities becomes less regular and also shrinks during most of the weekdays.

Patterns from Traffic Sensors: The largest difference method leads to 18 pairs of common components with small discriminative scores (M = .042, SD = .031) and two sets of discriminative components with large discriminative scores (M = .12, SD = .10), respectively. Below we show several interesting patterns among them all. The first two sets of common patterns are similar to the ones from the social media data in their respective distributions, while the last one differs from each.

Common Pattern 1. Fig. 12 shows the 6th patterns one month before (with discriminative score as .037) and one month after (with discriminative score as .000) the attacks related

[4]https://www.thrillist.com/eat/paris/paris-arrondissements-ranked-by-their-food-and-drink

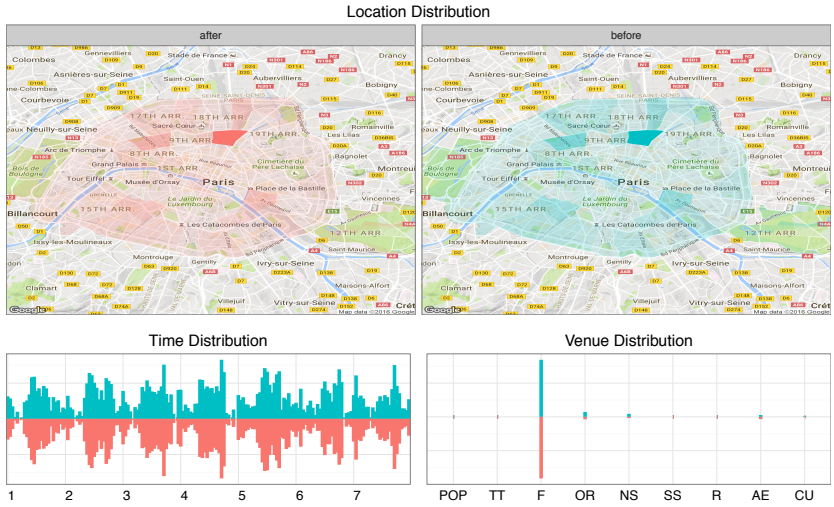


Fig. 12. **Common Pattern 1 from Paris traffic sensors.** $6th$ component before the attacks and $6th$ component after the attacks.

to the activities of food. We observe that the patterns from each tensor are practically the same in all three dimensions. This set of patterns spans across multiple districts in Paris, while mostly from the 10th arrondissement. In the time dimension, this pattern reaches its peak during the day in the weekdays and tends to peak during the night on the weekends.

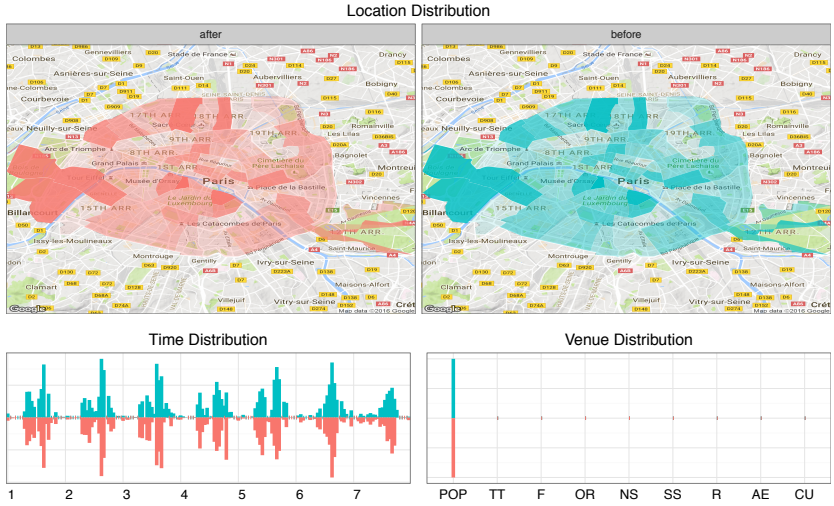


Fig. 13. **Common pattern 2 from Paris traffic sensors.** $14th$ component before the attacks and $14th$ component after the attacks.

Common Pattern 2. Fig.13 shows the 14th components one month before (with discriminative score as .037) and one month after (with discriminative score as .084) the attacks, corresponding to the activities of professional places. The patterns from each tensor are very

similar in all three dimensions. This set of patterns spans across multiple districts in Paris. We can observe two peaks during the day-time, for which we conjecture each of them can relate to the rush hour for work. The weekend traffic, however, is more centralized during the day.

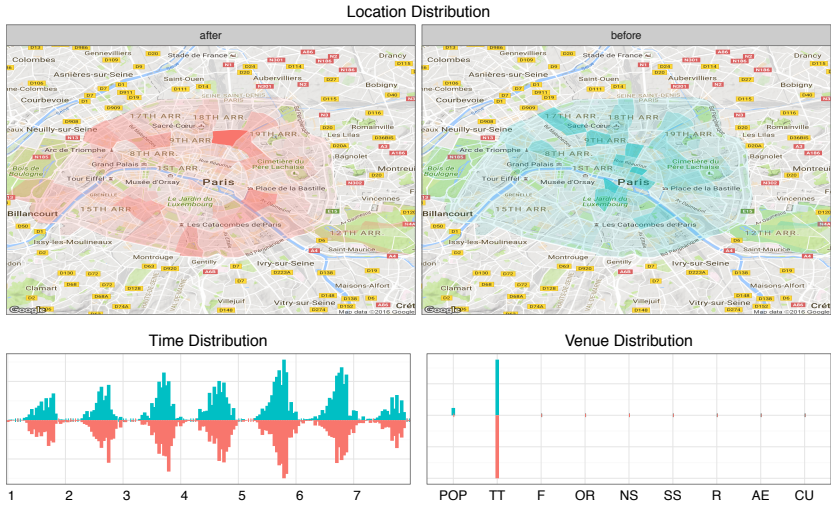


Fig. 14. **Unique patterns from Paris traffic sensors.** $20th$ component before the attacks and $19th$ component after the attacks.

Unique Patterns. Fig. 14 shows the 20th (with discriminative score as .169) component for one month before and 19th (with discriminative score as .096) component for one month after the attacks. We select these two because they exhibit similar distributions both in time and activities, as both of them show daily travel and transportation patterns while being very distinct regarding their location distribution in the city. Prior to the attacks, the destination of travel and transportation seems to fall around multiple locations, while several of them are close the attack sites (e.g., 3rd, 4th, and 11th arrondissement). However, in the following month, the traffic appears to have been more centralized to the 10th arrondissement and also tend to be more spread out from the affected areas. We suspect the difference in the location distribution could be because of road-blocks in those places after the attacks.

6.2 Thanksgiving in NYC

In this section, we demonstrate `PairFac` as a general urban analysis tool to uncover the changes in mobility patterns during holidays. Thanksgiving is a major national holiday in the United States. In this case study, we want to understand the differences in the mobility patterns revealed in the Thanksgiving holiday week over two consecutive years.

6.2.1 Dataset. We collected the taxi trips during Thanksgiving week of 2014 and 2015, respectively, which accumulates to 4,845,322 trips. Table 3 lists the dataset used in this case study. The information about each trip includes the pick-up location, drop-off location, pick-up time, drop-off time, the number of passengers. Similar to the previous case study, we also supply taxi trips with Foursquare data to model the location-time venue distribution. It contains 554,791 check-ins from 62,120 POIs in the NYC area between April 2012 and September 2013.

Type of Data	Dimensions extracted	Volume of raw data extracted
Taxi Trips	Location, Time	4,845,322 Trips
Check-ins and POI database	Location, Time, Activity	554,791 check-ins with 62,120 POI information

Table 3. Data sources used in the case study of Thanksgiving holiday week in NYC.

Component Index	1	2	3	4	5	6	7	8	9	10
Before	$1.1e-02$	$1.2e-02$	0.027	0.00508	0.039	0.021	0.087	0.18	0.19	0.43
After	$2.2e-05$	$1.2e-07$	0.000	0.02174	0.000	0.060	0.098	0.18	0.20	0.45
Difference	$0.0e+00$	$9.8e-04$	0.014	0.00019	0.012	0.042	0.103	0.17	0.04	0.48

Table 4. The discriminative scores associated with each component in NYC case study. The third row shows the difference between the components with the consecutive indexes.

6.2.2 Case Study Setup. Again, we construct three-mode tensors, where the three dimensions are location, time, and venue type, respectively. We keep the time and venue dimension the same as the Paris attacks case study. For the location dimension, we extract 193 neighborhoods in NYC. For the data tensors, we first construct a matrix LT, where LT_{ij} is the total number of passengers that are dropped off in the i-th neighborhood at the j-th hour in the week. Then, we construct a matrix FTV, where FTV_{ijk} is the probability of Foursquare check-ins in the k-th venue category that falls in the i-th neighborhood at the j-th hour in the week. Thus, for each cell at a given hour in the week, we know from the matrix FTV the probability distribution of activities over the nine categories. Finally, the entries in the data tensor are computed following Eq. 38. In our experiments, we set $\alpha = \beta = 10^{-8}$, $\gamma = 10^{-7}$, and $R = 10$.

6.2.3 Results. The largest difference method suggests only one set of discriminative components 10th component before (with discriminative score as .43) and 10th after (with discriminative score as .45), with the rest being common components. However, our observation is that components starting from 8th have already shown different degrees of differences in their distributions. This suggests that using a single cut-off to differentiate common and discriminative components might be too simplified a measure to determine the categories of the components. On the other hand, the discriminative score provided by our model can potentially provide a more accurate measurement of the similarity of the components.

In this section, we show several interesting patterns revealed by our method. Again, we first show two common patterns, followed by two sets of discriminative patterns:

Common Pattern 1. Fig. 15 shows the first components from 2014 (with discriminative score as .001) and 2015 (with discriminative score as .000), respectively. Their activity distributions reveal a pattern of mixed functions including professional places, outdoor recreations, travel, and transportation, etc. The activities mostly center during the day-time and the areas associated with this pattern (e.g., Times Square and Central Park) suggest the associated activities (e.g., Times Square for professional places and transportation hubs). We observe that these two patterns have almost identical distributions in all three dimensions with Thursday (the day of Thanksgiving) being the least active day.

Common Pattern 2. Fig 16 shows the comparison between the 2nd components from Thanksgiving week in 2014 (with discriminative score as .0.001) and 2015 (with discriminative

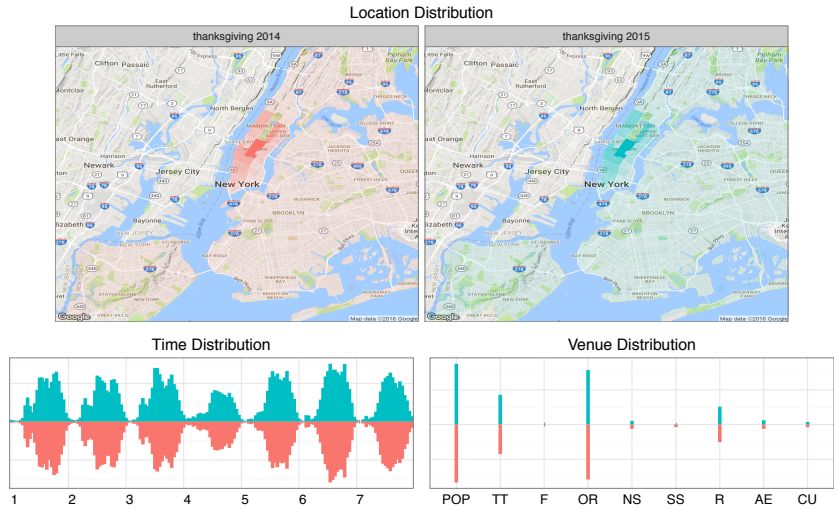


Fig. 15. **The first common patterns from NYC taxi trip.** 1st component from 2014 Thanksgiving week and 1st component from 2015 Thanksgiving week.

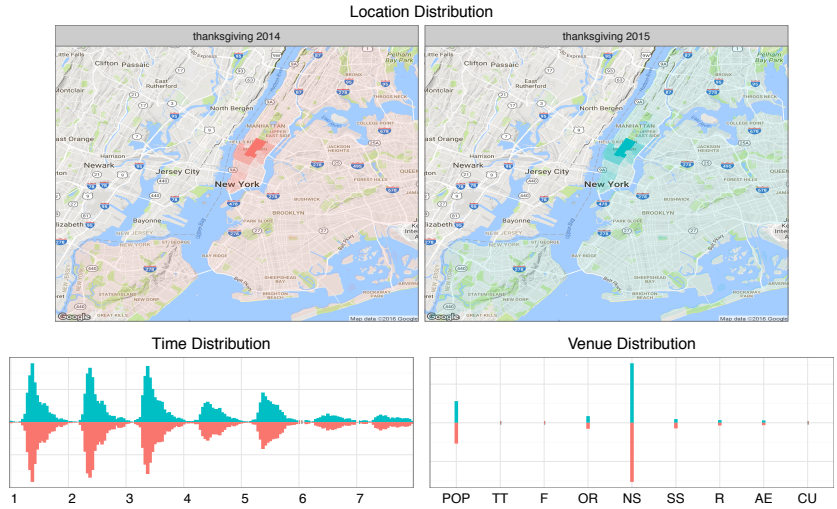


Fig. 16. **The second common patterns from NYC taxi trips.** $2nd$ component from 2014 Thanksgiving week (red) and $2nd$ component from 2015 Thanksgiving week (blue).

score as .0.000). We observe that the patterns in 2014 and 2015 are almost indifferentiable in all three dimensions. This set of patterns focuses on the nightlife spots activities around Times Square with their peaks spanning from Mondays to Wednesdays while decreasing on.

Unique Patterns 1. Fig.17 shows both 8th component of 2014 (with discriminative score as .18) and 8th component of 2015 (with discriminative score as .18) center their activities around Midtown. However, during the Thanksgiving week of 2014, the focus of the activities from this pattern is related to professional places or colleges and universities, while the focus moves to food and professional places in 2015. For the time dimension, we observe there is

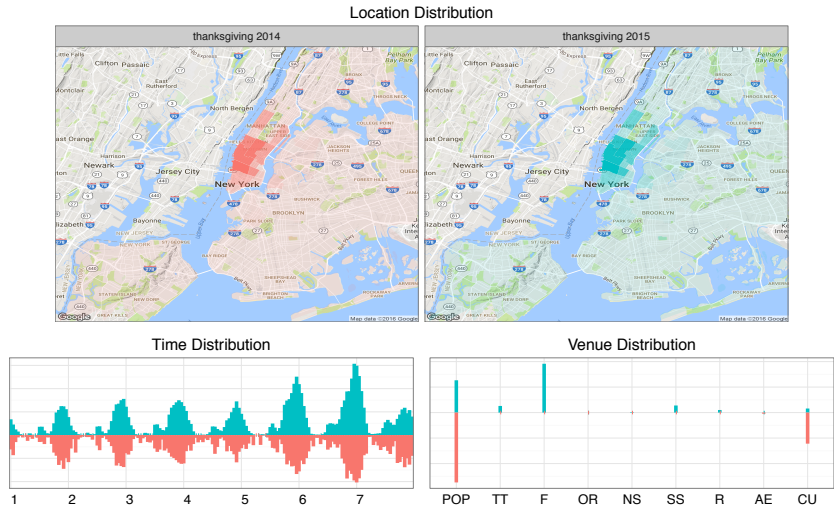


Fig. 17. **The unique patterns from NYC taxi trips.** $8th$ component from 2014 Thanksgiving week and $8th$ component from 2015 Thanksgiving week.

a higher volume of activities over the weekend in 2015 and slightly more activities on the Thanksgiving day, comparing to the one in 2014.

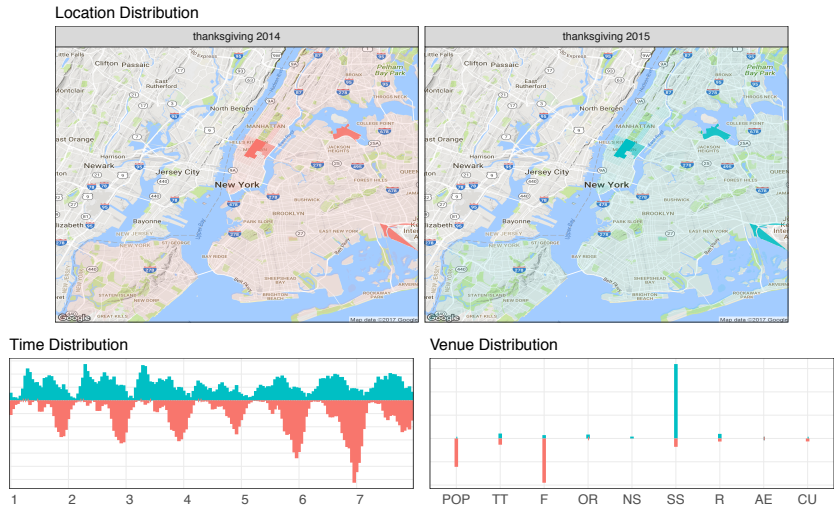


Fig. 18. **The unique patterns from NYC taxi trips.** $10th$ component from 2014 Thanksgiving week and $10th$ component from 2015 Thanksgiving week.

Unique Patterns 2. Fig.17 shows both 10th component of 2014 (with discriminative score as .43) and the 10th component of 2015 (with discriminative score as .45) that center their activities around Midtown, LaGuardia, and JFK. Although two patterns have almost identical location preferences, the time and the venue associated differ. While the pattern in 2014 has an array of activities (food, professional places, shopping and services and travel

& transportation), the one in 2015 primarily focuses on the shopping and services (e.g., shopping in 5th Ave). In the time mode, the pattern in 2014 has a relative low volume of activities during the weekdays with a higher volume over the weekend, while in 2015 the pattern seems to posses relatively less changes between the weekdays and the weekends. This could be due to the effect of the weather conditions during these two periods of time. In Thanksgiving holiday week 2014, there was a significant winter storm and it wasn't until the weekend that the temperature were back in the high 40's[5]. However, the Thanksgiving holiday week in 2015 has seen sunny weather all week with high temperatures from the upper 40's to the mid-60's.

7 DISCUSSION

In what follows we discuss some open issues with our study as well as our future directions:

(1) As the first step of this study, we undertook the effort of removing the need to manually pre-determine the number of common and discriminative components. Despite the advancments we made, there still exists the challenging question of the choice of the number components for `PairFac`. Although it is a common question for tensor factorization, and dimensionality reduction tasks in general, it is also an essential step towards a more robust discovery of latent patterns, and the impact of an event in our study in particular. As part of our future work, we plan to investigate a more systematic way of determining the number of components, particularly for the application of event analytics.

(2) `PairFac` is useful in discovering the changes in multidimensional data during two time periods. There are two potential issues with the current model: a) compared to existing literature [19], it does not offer insights on the changes over multiple periods of time; b) the current model focuses on finding the changes in the whole subspace rather than in any particular dimension. The latter could potentially be tackled through a dimension-specific regularization term in `PairFac`'s optimization objective. However, the task of analyzing changes over multiple time-periods could be more challenging, since `PairFac` requires the computation of the auxiliary tensors that host the pair-wise common and discriminant signals. The number of auxiliary tensors needed would increase dramatically as the number of original tensors (i.e., time periods) increases. Hence, more research is required to determine how to scalably model persistent and changing patterns over multiple time periods.

(3) The impact of disasters can be measured from different aspects based on the availability of different datasets. By investigating the disasters using multiple datasets, it is possible to discover the impact that might otherwise be obscured in isolated datasets. The construction of input tensors and the interpretation of the output from the different data sources would depend on the nature of the datasets (i.e., their meanings and granularities). For example, Twitter data contains specific information regarding activities such as locations, times, and content, while sensor data provides broad information about traffic flow as measured by the vehicles. These two datasets provide complementary aspects of human mobility – the kinds of places they visited and tweeted about or how they use vehicles to move around the city. For example, in our previous study of the immediate impact of urban mobility after the Paris attacks [37], we observed more Twitter activity close to night entertainment areas, but much less traffic. In a scenario of disaster aftermath, Twitter data could help identify how people went out to the streets to show solidarity, or commemorate the victims, whereas the traffic sensor data could show how people's activities on the streets subsequently blocked

[5]https://www.nbcnewyork.com/news/local/New-York-City-New-Jersey-Snow-Thanksgiving-Travel-Delays-Roads-Forecast-283718461.html

road segments and reduced the automotive traffic in the same region. As different datasets illuminate distinctive aspects of city dynamics, it is an interesting next step to investigate the correlations among different datasets in order to devise models that can be utilized to discover patterns.

(4) The case studies presented in Section 6 construct tensors with three dimensions (locations, time, venues), where each employs a pre-determined level of details in the corresponding mode revealed from `PairFac`. For example, we use neighborhoods for the location dimension. We proceed to this level because it enables us to further study the potential factors that lead to the observed changes in different neighborhoods. These factors can be obtained from readily available data, such as demographics, which are usually aggregated at the level of city neighborhoods. It is important to note that, with different settings, we might obtain understandings of the urban dynamics in different resolutions. The choice of resolutions at this moment is rather application dependent. In our future work, we aim to develop an extension of our method that can automatically disclose the most interesting details.

(5) Despite the issues and limitations of acknowledged as above, `PairFac` provides interesting insights in evaluating the impact of events in the city. In our first case study of the Paris attacks, we reveal the changes in the mobility patterns based on two datasets, social media data (geo-tagged Twitter content) and traffic sensors, separately. Compared to [37], the results show that most of the patterns resumed to the same orders as they were before the attacks (e.g., Work-related patterns in Fig. 9 and 13, Food-related patterns in Fig. 10 and 12). However, according to the Twitter data, the outdoor-recreation pattern in the northern part of Paris has not been as exercised as much as it was before. Particularly, Thursdays see one of most reduced activities. We guess this might be due to the police raid in the northern suburb of Paris, Saint-Denis, which is close to Parc de la Villette, on November 18th. Although the siege was ended in the morning of November 18th, it wasn't until the next day that French officials announced the primary suspect in the Paris attacks was killed in the raid [6]. On the other hand, from the traffic sensor data, we show the transportation pattern has seen the distinct focus of regions, where people tend to alternate their choices of transportation to the areas that are away from the attack sites. This change in transportation patterns was only observed from the traffic sensor data. We guess this could be due to the road blocks in the areas close to the attack sites where the access could be limited to foot traffic. The results from the NYC Thanksgiving case study (comparing 2014 to 2015 are contradicting to our expectations as we observe almost identical patterns of Outdoor, Transportation (Fig. 15) and Nightlife activities (Fig. 16), and surprisingly more food activities (Fig. 17). Although FBI has warned that the media officer of ISIS had called the Macy's Thanksgiving Day parade an "excellent target," our analysis shows that the mobility patterns do not vary much over the two years even under the influence of potential and imminent terror attack. However, this could also because of reinforced security due to the terror threat as 2,500 police officers were deployed on the ground for the Thanksgiving [7].

8 CONCLUSION

In this work, we propose a new analytic approach `PairFac` that aims to discover the impact of an exogenous event on multiple aspects of human activities in the urban environment.

[6] https://en.wikipedia.org/wiki/2015_Saint-Denis_raid
[7] http://www.nydailynews.com/new-york/hundreds-turn-thanksgiving-parade-balloon-inflation-article-1.2447267

With the multidimensional nature of the mobility/behavioral data, we formulate the impact discovery as the problem of identifying common and discriminative subspaces from these datasets. Compared to the existing methods, our approach has the advantage of automatically distinguishing the common and discriminative components. This is realized through the introduction of auxiliary tensors and additional column regularization for the learning optimization objective of discriminative weights. We conduct extensive experiments with synthetic data to demonstrate `PairFac`'s effectiveness and scalability.

We apply `PairFac` in two case studies and demonstrate its capability to reveal persistent and changing mobility patterns with respect to events of interest. For example, in our first case study using data from the terrorist attacks in Paris of 2015, we see that activities around professional life and food venues experienced the least changes. Using `PairFac` process results, they appear to have identical location and time distribution over the course of the period of study. The most dramatic change was seen in outdoor recreation activities in the 1st and 19th Arrondissements. Although they share the same location distribution, we observe that their associated times became irregular.

`PairFac` is not only for use in determining disaster impact, as seen in the Pairs terrorist attacks case study, but also as a general urban analysis tool to identify changes in the activities of the city's inhabitants over the different periods of time. This use of `PairFac` can be seen in the example of our second case study. We applied `PairFac` to the data regarding taxi travel during the Thanksgiving holiday week in 2014 and 2015, in order to investigate the changes, if any, in mobility patterns. The results suggest that most of the patterns remain consistent and reveal the unique attributes of mobility in NYC during this major American holiday. For example, in the Times Square area, both nightlife and professional activities decrease between Thanksgiving Thursday through Sunday. When we compare the two Thanksgivings, there are some differences in activities. Specifically, in Thanksgiving 2015, in midtown Manhattan, there are less professional and academic activities, but a greater number of food related activities in the same area with similar time distribution. One potential explanation for the increase in food related activities for Thanksgiving 2015 is that people were not influenced to change their behavior by the conceivable increase in risk of terrorist attack. Additionally, the police took extra precautions and placed heavy police force on the ground to safeguard the areas of the city most at risk[8].

There are several future directions for this work. (1) In this study, we present the case studies from the perspectives of two datasets with distinct nature of their origins and representations separately (e.g., Twitter check-ins and traffic sensors from Paris). In our next step, we also would like to investigate what stands in common for these two data sources. We believe this would shed light on how we could better understand the phenomenon that originates from different data sources. (2) It is also our desire to study how the patterns differ and evolve over a period of time instead of only considering before and after the events. We should note that pairwise computation of common and discriminative tensors as introduced in this study make sense for the purpose of probing the shifts during these two periods. However, such design should be used with caution since it could be too computationally expansive for a sequence of tensors over time. In this case, a different design for the computation of the common and discriminative signals might be required. We believe that using the mean of a sequence of tensors could be a more natural way to capture the common signals over time. However, future work is needed to comprehensively understand the problem and to explore potential solutions. (3) Another natural extension of our work is to investigate the driving

[8]https://nypost.com/2015/11/26/nypd-beefs-up-security-ahead-of-thanksgiving-day-parade/

factors that direct the observed changes. This can be particularly insightful in building disaster impact predictions. (4) As the current output of our algorithm ties to the choice of the number of components, we are not guaranteed to obtain meaningful patterns with a certain designated number of components. To resolve this issue, we want to extend `PairFac` under the framework of hierarchical impact discovery by including a component ranking approach across multiple levels.

ACKNOWLEDGEMENT

This work is part of the research supported from NSF #1634944, #1637067, and #1739413, the CRDF at the University of Pittsburgh, and the ARO Young Investigator Award W911NF-15-1-0599 (67192-NS-YIP). Any opinions, findings, and conclusions or recommendations expressed in this material do not necessarily reflect the views of the funding sources.

APPENDIX A PROOF OF THEOREM 2

Proof. Suppose $\bar{\rho} = (\bar{\mathbf{U}}_q^{(m)}, \bar{\mathbf{w}}_q)$ is a limit point of $\{\rho = (\mathbf{U}_{q,k}^{(m)}, \mathbf{w}_{q,k})\}$. Then there exists a subsequence $\{\rho^{k_j}\}$ converging to $\bar{\rho}$. Based on the above analysis, the sequence $\{\rho^{k_j+1}\}$ also converges to $\bar{\rho}$. According to the update rule 23, it holds that:

$$\mathbf{U}_{q,k_j+1}^{(m)} = \max\left(0, \mathbf{U}_{q,k_j}^{(m)} - \mathbf{G}_{\mathbf{U}_q^{(m)}}^{k_j} / \mathcal{L}_{\mathbf{U}_q^{(m)}}^{k_j}\right), \forall q \text{ and } \forall m.$$

Letting $j \to \infty$, we have

$$\bar{\mathbf{U}}_q^{(m)} = \max\left(0, \bar{\mathbf{U}}_q^{(m)} - \mathbf{G}_{\bar{\mathbf{U}}_q^{(m)}} / \mathcal{L}_{\bar{\mathbf{U}}_q^{(m)}}\right), \forall q \text{ and } \forall m,$$

where $\mathcal{L}_{\bar{\mathbf{U}}_q^{(m)}} = \lambda_{max}({\mathbf{F}_{\mathbf{U}_q^{(m)}}^{k-1}}^T \mathbf{F}_{\mathbf{U}_q^{(m)}}^{k-1})$. This implies $\bar{\mathbf{U}}_q^{(m)}$ is a minimizer of the linearized proximal regularization of $\Psi(\mathbf{U}_{q,k}^{(m)})$ regarded as a function with respect to $\mathbf{U}_q^{(m)}$ at point $\bar{\mathbf{U}}_q^{(m)}$, i. e.,

$$\mathbf{U}_q^{(m)} = \underset{\mathbf{U}_q^{(m)} \geq 0}{\operatorname{argmin}} \langle \bar{\mathbf{G}}_{\mathbf{U}_q^{(m)}}, \mathbf{U}_q^{(m)} - \bar{\mathbf{U}}_q^{(m)} \rangle + \frac{\mathcal{L}_{\mathbf{U}_q^{(m)}}}{2} \left\| \mathbf{U}_q^{(m)} - \bar{\mathbf{U}}_q^{(m)} \right\|_F. \tag{39}$$

Assume $\mathbf{U}_{q,*}^{(m)}$ is a solution of problem

$$\underset{\mathbf{U}_q^{(m)} \geq 0}{\operatorname{argmin}} \Psi(\mathbf{U}_q^{(m)}, \bar{\mathbf{w}}_q). \tag{40}$$

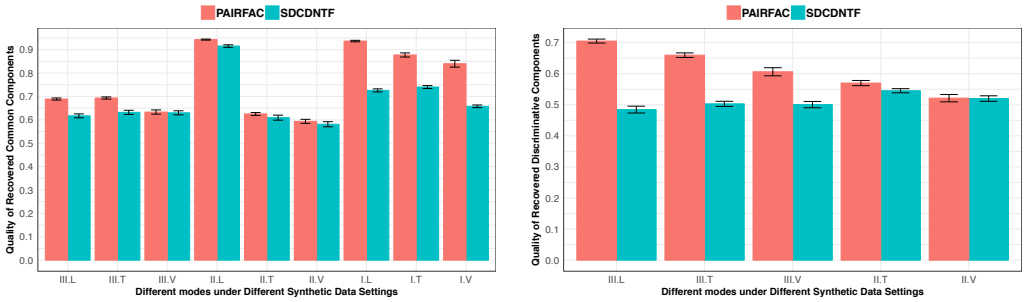
Using Lemma 2, we have $\Psi(\mathbf{U}_{q,*}^{(m)}, \bar{\mathbf{w}}_q) - \Psi(\bar{\mathbf{U}}_q^{(m)}, \bar{\mathbf{w}}_q) \geq 0$, which implies $\bar{\mathbf{U}}_q^{(m)}$ is also a solution of 40. Therefore $\bar{\mathbf{U}}_q^{(m)}$ must satisfy the KKT-conditions of 40. It is easy to see that Eq. 37 is the KKT conditions for J_3. This completes the proof. The proof of the KKT conditions for $\mathbf{w}_q$ can be obtained similarly and is thus omitted.□

APPENDIX B ADDITIONAL EXPERIMENTS ON THE SYNTHETIC DATASETS

Experiment Setting. We conducted additional experiments with three more synthetic data generation settings with a varying number of different modes:

I None of the three modes are different (all modes are similar: similar in $\mathbf{U}^L$, $\mathbf{U}^T$, and $\mathbf{U}^V$);

II Two modes are different with one being similar (similar in $\mathbf{U}^L$; but different in $\mathbf{U}^T$ and $\mathbf{U}^V$);



(a) Common Components (b) Discriminative Components

Fig. 19. **Experiment results on various settings of synthetic data generation.** The x-axis shows different modes (L, T, V) under three settings of synthetic data generation (I, II, III). The y-axis shows the quality of recovered common components. Note that the modes where the synthetic data tensors shared are not included in (b) since they do not have discriminative components.

III All three modes are different (different in $\mathbf{U}^L$, $\mathbf{U}^T$, and $\mathbf{U}^V$).

We added random sparse Gaussian noises to all three modes and followed the same parameter settings as in our manuscript. In terms of baseline, as SDCDNTF performs best in all the baselines. In this set of additional experiments, we compared PairFac with SDCDNTF. SDCDNTF can be considered as a representative approach that requires the pre-defined numbers of common and discriminative components, whereas PairFac automatically identifies the common and discriminative components in a data-driven manner.

Results. Fig. 19 (a) and (b) show the results the quality or recovered common and discriminative components under the three experiment settings, respectively. The x-axis shows different modes (L, T, V) under different synthetic data settings (I, II, III) and the y-axis shows the quality of the recovered components if applicable. We observe that `PairFac` consistently achieves superior performance compared to `SDCDNTF` in all settings.

REFERENCES

[1] Evrim Acar, Daniel M Dunlavy, Tamara G Kolda, and Morten Mørup. 2010. Scalable tensor factorizations with missing data. In *Proceedings of the 2010 SIAM international conference on data mining*. SIAM, 701–712.

[2] Harshavardhan Achrekar, Avinash Gandhe, Ross Lazarus, Ssu-Hsin Yu, and Benyuan Liu. 2011. Predicting flu trends using twitter data. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*. IEEE, 702–707.

[3] James P Bagrow, Dashun Wang, and Albert-Laszlo Barabasi. 2011. Collective response of human populations to large-scale emergencies. *PloS one* (2011).

[4] Amir Beck and Marc Teboulle. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences* 2, 1 (2009), 183–202.

[5] Alex Beutel, Partha Pratim Talukdar, Abhimanu Kumar, Christos Faloutsos, Evangelos E Papalexakis, and Eric P Xing. 2014. FlexiFaCT: Scalable Flexible Factorization of Coupled Tensors on Hadoop.. In *SDM*. SIAM, 109–117.

[6] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. 2014. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming* 146, 1-2 (2014), 459–494.

[7] Direction de la Voirie et des déplacements Service des Déplacements. [n. d.]. Données trafic issues des capteurs permanents. http://opendata.paris.fr/explore/dataset/comptages-routiers-permanents/. ([n. d.]).

[8] Nicholas Diakopoulos, Mor Naaman, and Funda Kivran-Swaine. 2010. Diamonds in the rough: Social media visual analytics for journalistic inquiry. In *Visual Analytics Science and Technology (VAST),*

2010 IEEE Symposium on. IEEE, 115–122.

[9] Sunil Kumar Gupta, Dinh Phung, Brett Adams, Truyen Tran, and Svetha Venkatesh. 2010. Nonnegative shared subspace learning and its application to social media retrieval. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1169–1178.

[10] Sunil Kumar Gupta, Dinh Phung, Brett Adams, and Svetha Venkatesh. 2013. Regularized nonnegative shared subspace learning. *Data mining and knowledge discovery* 26, 1 (2013), 57–97.

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

[12] Inah Jeon, Evangelos E Papalexakis, U Kang, and Christos Faloutsos. 2015. Haten2: Billion-scale tensor decompositions. In *2015 IEEE 31st International Conference on Data Engineering*. IEEE, 1047–1058.

[13] U Kang, Evangelos Papalexakis, Abhay Harpale, and Christos Faloutsos. 2012. Gigatensor: scaling tensor analysis up by 100 times-algorithms and discoveries. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 316–324.

[14] Hannah Kim, Jaegul Choo, Jingu Kim, Chandan K Reddy, and Haesun Park. 2015. Simultaneous discovery of common and discriminative topics via joint nonnegative matrix factorization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 567–576.

[15] Tamara Gibson Kolda. 2006. *Multilinear operators for higher-order decompositions.* Technical Report. Sandia National Laboratories.

[16] Tamara G Kolda and Jimeng Sun. 2008. Scalable tensor decompositions for multi-aspect data mining. In *2008 Eighth IEEE international conference on data mining*. IEEE, 363–372.

[17] Ryong Lee and Kazutoshi Sumiya. 2010. Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection. In *Proceedings of the 2nd ACM SIGSPATIAL international workshop on location based social networks*. ACM, 1–10.

[18] Yu-Ru Lin and Drew Margolin. 2014. The ripple of fear, sympathy and solidarity during the Boston bombings. *EPJ Data Science* (2014).

[19] Yu-Ru Lin, Jimeng Sun, Hari Sundaram, Aisling Kelliher, Paul Castro, and Ravi Konuru. 2011. Community discovery via metagraph factorization. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 5, 3 (2011), 17.

[20] Wei Liu, Jeffrey Chan, James Bailey, Christopher Leckie, and Kotagiri Ramamohanarao. 2013. Mining labelled tensors by discovering both their common and discriminative subspaces. In *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 614–622.

[21] Yong Luo, Dacheng Tao, Kotagiri Ramamohanarao, Chao Xu, and Yonggang Wen. 2015. Tensor canonical correlation analysis for multi-view dimension reduction. *IEEE transactions on Knowledge and Data Engineering* 27, 11 (2015), 3111–3124.

[22] Michael Madaio, Shang-Tse Chen, Oliver L Haimson, Wenwen Zhang, Xiang Cheng, Matthew Hinds-Aldrich, Duen Horng Chau, and Bistra Dilkina. 2016. Firebird: Predicting Fire Risk and Prioritizing Fire Inspections in Atlanta. *arXiv preprint arXiv:1602.09067* (2016).

[23] Michael Mathioudakis and Nick Koudas. 2010. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 1155–1158.

[24] Sharad Mehrotra, Xiaogang Qiu, Zhidong Cao, and Austin Tate. 2013. Technological Challenges in Emergency Response. *IEEE Intelligent Systems* (2013).

[25] Yue Ning, Sathappan Muthiah, Huzefa Rangwala, and Naren Ramakrishnan. 2016. Modeling Precursors for Event Forecasting via Nested Multi-Instance Learning. In *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

[26] Bei Pan, Yu Zheng, David Wilkie, and Cyrus Shahabi. 2013. Crowd sensing of traffic anomalies based on human mobility and social media. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 344–353.

[27] Linsey Xiaolin Pang, Sanjay Chawla, Wei Liu, and Yu Zheng. 2013. On detection of emerging anomalous traffic patterns using GPS data. *Data & Knowledge Engineering* 87 (2013), 357–373.

[28] Evangelos E Papalexakis, Christos Faloutsos, Tom M Mitchell, Partha Pratim Talukdar, Nicholas D Sidiropoulos, and Brian Murphy. 2014. Turbo-smt: Accelerating coupled sparse matrix-tensor factorizations by 200x. In *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 118–126.

[29] William E Schlenger, Juesta M Caddell, Lori Ebert, B Kathleen Jordan, Kathryn M Rourke, David Wilson, Lisa Thalji, J Michael Dennis, John A Fairbank, and Richard A Kulka. 2002. Psychological

reactions to terrorist attacks: findings from the National Study of Americans' Reactions to September 11. *Jama* 288, 5 (2002), 581–588.

[30] Xuan Song, Quanshi Zhang, Yoshihide Sekimoto, Teerayut Horanont, Satoshi Ueyama, and Ryosuke Shibasaki. 2013. Intelligent system for human behavior analysis and reasoning following large-scale disasters. *IEEE Intelligent Systems* (2013).

[31] Xuan Song, Quanshi Zhang, Yoshihide Sekimoto, Teerayut Horanont, Satoshi Ueyama, and Ryosuke Shibasaki. 2013. Modeling and probabilistic reasoning of population evacuation during large-scale disaster. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

[32] Dacheng Tao, Xuelong Li, Xindong Wu, and Stephen J Maybank. 2007. General tensor discriminant analysis and gabor features for gait recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 10 (2007).

[33] Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welpe. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. *ICWSM* (2010).

[34] Raheem A Usman, FB Olorunfemi, GP Awotayo, AM Tunde, and BA Usman. 2013. Disaster Risk Management and Social Impact Assessment: Understanding Preparedness, Response and Recovery in Community Projects. In *Environmental Change and Sustainability*. InTech.

[35] Xiaofeng Wang, Matthew S Gerber, and Donald E Brown. 2012. Automatic crime prediction using events extracted from twitter posts. In *International conference on social computing, behavioral-cultural modeling, and prediction*. Springer, 231–238.

[36] Xidao Wen and Yu-Ru Lin. 2016. Sensing Distress Following a Terrorist Event. In *Social, Cultural, and Behavioral Modeling: 9th International Conference, SBP-BRiMS 2016, Washington, DC, USA, June 28-July 1, 2016, Proceedings 9*. Springer, 377–388.

[37] Xidao Wen, Yu-Ru Lin, and Konstantinos Pelechrinis. 2016. PairFac: Event Analytics through Discriminant Tensor Factorization. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 519–528.

[38] Yangyang Xu and Wotao Yin. 2013. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on imaging sciences* (2013).

[39] Dingqi Yang, Daqing Zhang, and Bingqing Qu. 2016. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Transactions on Intelligent Systems and Technology (TIST)* 7, 3 (2016), 30.

[40] Kui Yu, Dawei Wang, Wei Ding, Jian Pei, David L Small, Shafiqul Islam, and Xindong Wu. 2015. Tornado Forecasting with Multiple Markov Boundaries. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2237–2246.

[41] Chidong Zhang, Brian E Mapes, and Brian J Soden. 2003. Bimodality in tropical water vapour. *Quarterly Journal of the Royal Meteorological Society* 129, 594 (2003), 2847–2866.

[42] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 3 (2014), 38.