Computing Urban Traffic Congestions by Incorporating Sparse GPS Probe Data and Social Media Data

SENZHANG WANG, Nanjing University of Aeronautics and Astronautics; Collaboration Innovation Center of Novel Software Technology and Industrialization
XIAOMING ZHANG, Beihang University
JIANPING CAO, National University of Defense Technology
LIFANG HE, Shenzhen University
LEON STENNETH, BMW, Audia, and Daimler's HERE Connected Driving
PHILIP S. YU, University of Illinois at Chicago; Tsinghua University
ZHOUJUN LI, Beihang University
ZHIQIU HUANG, Nanjing University of Aeronautics and Astronautics

Estimating urban traffic conditions of an arterial network with GPS probe data is a practically important while substantially challenging problem, and has attracted increasing research interests recently. Although GPS probe data is becoming a ubiquitous data source for various traffic related applications currently, they are usually insufficient for fully estimating traffic conditions of a large arterial network due to the low sampling frequency. To explore other data sources for more effectively computing urban traffic conditions, we propose to collect various traffic events such as traffic accident and jam from social media as complementary information. In addition, to further explore other factors that might affect traffic conditions, we also extract rich auxiliary information including social events, road features, Point of Interest (POI), and weather. With the enriched traffic data and auxiliary information collected from different sources, we first study the traffic co-congestion pattern mining problem with the aim of discovering which road segments geographically close to each other are likely to co-occur traffic congestion. A search tree based approach is proposed to efficiently discover the co-congestion patterns. These patterns are then used to help estimate traffic congestions and detect anomalies in a transportation network. To fuse the multisourced data, we finally propose a coupled matrix and tensor factorization model named TCE_R to more accurately complete the sparse traffic congestion matrix by collaboratively factorizing it with other matrices and tensors formed by other data. We evaluate the proposed model on the arterial network of downtown Chicago with 1,257 road segments whose total length is nearly 700 miles. The results demonstrate the superior performance of TCE_R by comprehensive comparison with existing approaches.

This work is supported by the NSFC under grants 61272083, 61370126, 61672081, 61602237, 61503253, and U1636211; the National High-tech R&D Program of China under grants 2015AA105303 and 2015AA016004; the NSF under grants IIS-1526499 and CNS-1626432; and Beijing Advanced Innovation Center for Imaging Technology (No. BAICIT-2016001).

This work is an extension of Wang et al. [2016a].

Authors' addresses: S. Wang and Z. Huang, College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics; emails: {szwang, zqhuang}@nuaa.edu.cn; X. Zhang and Z. Li (corresponding author), State Key Laboratory of Software Development Environment, Beihang University; emails: {yolixs, lizj}@buaa.edu.cn; J. Cao, National University of Defence Technology, email: caojianping@nudt.edu.cn; L. He, Shenzhen University; email: lifanghescut@gmail.com; L. Stenneth, BMW, Audi, and Daimler's HERE Connected Driving; email: lstenneth@gmail.com; P. S. Yu, School of Computer Science, University of Illinois at Chicago and Institute for Data Science, Tsinghua University; email: psyu@uic.edu.

DOI: http://dx.doi.org/10.1145/3057281

CCS Concepts: • **Information systems → Spatial-temporal systems**; **Data mining**

Additional Key Words and Phrases: Social media, traffic congestion, matrix factorization, data fusion

ACM Reference Format:
Senzhang Wang, Xiaoming Zhang, Jianping Cao, Lifang He, Leon Stenneth, Philip S. Yu, Zhoujun Li, and Zhiqiu Huang. 2017. Computing urban traffic congestions by incorporating sparse GPS probe data and social media data. ACM Trans. Inf. Syst. 35, 4, Article 40 (July 2017), 30 pages.
DOI: http://dx.doi.org/10.1145/3057281

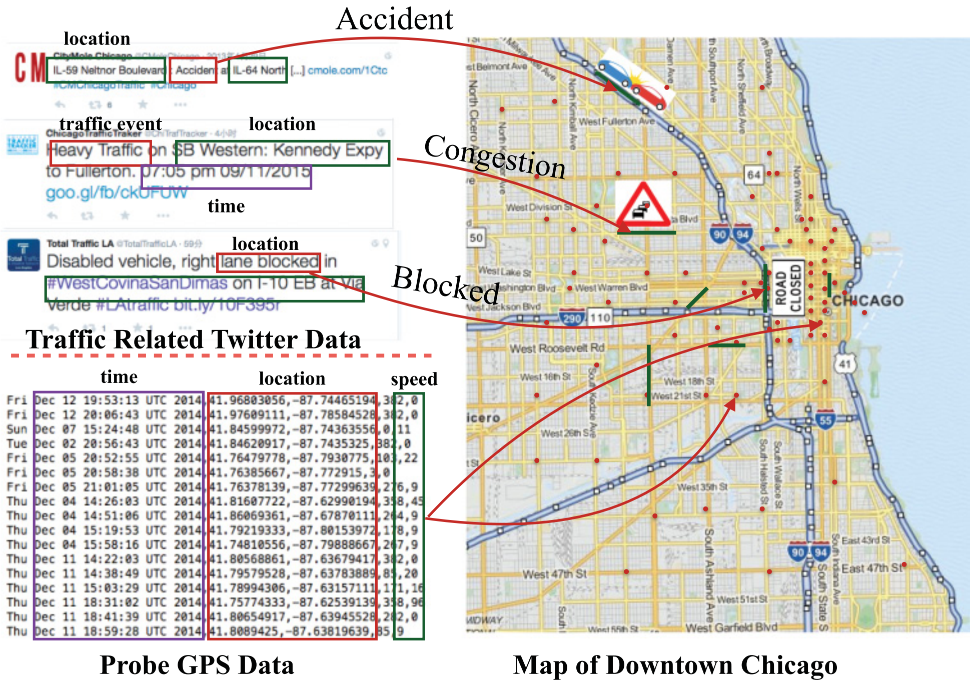
1. INTRODUCTION

Traffic congestion estimation in urban area is a practically important while substantially challenging problem. According to the report *2015 Mobility Scorecard* released by Texas A&M Transportation Institute, the yearly delayed time during traffic congestion is about 6.9 billion hours in the United States alone in 2014 [Schrank et al. 2015]. The estimated total cost due to traffic jam is over 160 billion dollars, and the figure is forecasted to grow to 192 billion by the year 2020. Traffic jam has become an urgent issue which can remarkably restrict urban development, and the condition is becoming worse with the urban expansion and population explosion in many developing countries [Kenworthy 2006; Tsekeris and Geroliminis 2013]. It becomes difficult for traditional models that rely on a single type of sensor data to capture the complicated traffic states in real time. Traffic information from only one data source is insufficient to meet the need of providing a real-time traffic congestion estimation in a large city. In this sense, it is an urgent research issue to carry on a deeper study on the hidden patterns of urban traffic congestions, and develop a new method which can effectively combine multisourced data for more accurately monitoring traffic conditions of an entire city in real time.

Traditional traffic monitoring methods rely on various road sensors such as loop detectors [Muñoz et al. 2003; Yuan et al. 2014], surveillance cameras [Ozkurt and Camci 2009], radar, etc. Yuan et al. proposed an extended Kalman filtering model to estimate network-wide traffic state by utilizing loop detector and floating car data [Yuan et al. 2014]. Oakurt and Camci proposed to apply the neural networks to calculate traffic density based on the video data collected from the video monitoring and surveillance systems [Ozkurt and Camci 2009]. A major issue of these methods is that the spatiotemporal coverage of the road sensors is usually limited due to the high cost of deploying and maintaining them [Zheng et al. 2014a]. For example, as deploying and maintaining loop sensors is very expensive in terms of money and human resources, they are usually employed for major roads rather than low-level streets [Zheng et al. 2014a]. Currently, GPS based probe vehicle data have been widely used to illuminate traffic conditions for applications including travel time estimation, map building, and congestion detection [Tao et al. 2012; Herring et al. 2010]. Many map services utilize probe vehicle data to estimate real-time congestions. Examples include Google Map, INRIX, Bing Map, NOKIA HERE Map, etc. [Higgins 2013]. However, as the GPS probe data are usually very sparse due to the low sampling frequency, they are usually not sufficient for fully estimating traffic conditions of a large arterial network [Wang et al. 2011]. Another issue of sensor-based methods is that they usually can only handle one type of sensor data, but cannot effectively incorporate other types of data as well as side information including road features, weather information, areas of the city, etc. Such sensor-based method is effective to detect the "recurring" congestions that are simply caused by more vehicles, but might be less effective to capture the "nonrecurring" congestions that are caused by various incidents such as accident and bad weather. How to combine different types of traffic data and traffic related side information for improving traffic congestion estimation is still not fully explored.

Currently, sharing real-time traffic information through social media platforms such as Twitter is becoming a common practice for both individual users and the official transportation departments [Endarnoto et al. 2011]. For example, *Roadnow Chicago* and *Total Traffic LA* are two Twitter accounts that focus on posting real-time traffic related tweets in Chicago and Los Angeles, respectively. Such tweets can be about road congestions, accidents, road constructions, and other traffic events. Besides the traffic related tweets posted by public transportation organizations and government, regular Twitter users can also post tweets to report traffic events during their traveling. By taking Twitter users as traffic sensors, the monitoring coverage of traffic conditions can be largely expanded as Twitter users including pedestrians, drivers, and passengers can spread over the entire city. In addition, traffic events like road construction and accident can be directly mentioned in the tweets, but are difficult to infer from GPS data. Such rich traffic event information reported by Twitter is essential to help us capture the reasons that cause "nonrecurring" congestions. It is reported that [Schrank et al. 2015] only half of the congestions experienced by Americans are "recurring" congestions, which are caused simply by more vehicles. The other half of congestions are "nonrecurring" congestions, and they are caused by temporary disruptions that take away part of the roadway from use. About 25% "nonrecurring" congestions are caused by incidents like traffic accidents and social events, 15% are caused by bad weather, and 10% are caused by road construction or closure. Twitter is a promising data source for us to obtain various traffic and social event information. Recently, there are increasing research interests to study how to utilize social media data to help understand traffic conditions [Daly et al. 2013; D'Andrea et al. 2015; Liu et al. 2014; Sílvio S. Ribeiro et al. 2012]. These works mainly focused on studying how to extract the traffic event information from tweets [D'Andrea et al. 2015; Liu et al. 2014; Schulz et al. 2013], how to locate the traffic events mentioned in the tweets [Sílvio S. Ribeiro et al. 2012], or how to monitor traffic congestions with real-time traffic event tweets [Chen et al. 2014; Wang et al. 2015]. These methods mostly ignored some auxiliary information including historical data, congestion correlation, road features, and weather, while this information is very helpful to reflect traffic conditions when the traffic related social media data are sparse [Wang et al. 2015].

The research focuses of this article are (1) studying the hidden traffic congestion correlation patterns among the road segments, and (2) building a hybrid model that can effectively combine the co-congestion patterns and different types of traffic related information including GPS probe data, traffic related tweets, social events, road features, Points of Interest (POIs), as well as weather information to more accurately estimate traffic congestions in urban areas. Specifically, we first widely collect and process traffic related tweets from both traffic authority accounts (explain later) and general user accounts. As illustrated in Figure 1, we then regard the traffic related tweets and GPS probe readings as two types of primary information. Each traffic related tweet can mention the location, traffic event, and time information, and each probe reading contains the vehicle speed and coordinate information. Based on the road segments mentioned in the tweets and the exact locations of the probe readings, we map them to the corresponding road segments. With a large number of historical data, we first investigate the traffic congestion correlations, namely, we discover which road segments geographically close to each other are very likely to co-occur congestion. We consider it as a spatiotemporal frequent pattern mining problem, and propose a search tree based method to efficiently discover all the frequent co-congestion patterns in downtown Chicago. The discovered co-congestion patterns could not only be helpful to the traffic congestion estimation task, but also can help us detect anomalies in the road networks. To more effectively capture the "nonrecurring" congestions, we also extensively collect other side information, including social events, road physical features,



CityMole Chicago @CMoleChicago · 2013年1月8日
IL-59 Neltnor Boulevard : Accident at IL-64 North [...] cmole.com/1Ctc #CMChicagoTraffic #Chicago

ChicagoTrafficTraker @ChiTrafTracker · 4小时
Heavy Traffic on SB Western: Kennedy Expy to Fullerton. 07:05 pm 09/11/2015 goo.gl/fb/ckUFUW

Total Traffic LA @TotalTrafficLA · 59分
Disabled vehicle, right lane blocked in #WestCovinaSanDimas on I-10 EB at Via Verde #LAtraffic bit.ly/10F395r

```
Fri Dec 12 19:53:13 UTC 2014,41.96803056,-87.74465194,382,0
Fri Dec 12 20:06:43 UTC 2014,41.97609111,-87.78584528,382,0
Sun Dec 07 15:24:48 UTC 2014,41.84599972,-87.74363556,0,11
Tue Dec 02 20:56:43 UTC 2014,41.84620917,-87.7435325,382,0
Fri Dec 05 20:52:55 UTC 2014,41.76479778,-87.7930775,103,22
Fri Dec 05 20:58:38 UTC 2014,41.76385667,-87.772915,3,0
Fri Dec 05 21:01:05 UTC 2014,41.76378139,-87.77299639,276,9
Thu Dec 04 14:26:03 UTC 2014,41.81607722,-87.62990194,358,45
Thu Dec 04 14:51:06 UTC 2014,41.86069361,-87.67870111,264,9
Thu Dec 04 15:19:53 UTC 2014,41.79219333,-87.80153972,178,9
Thu Dec 04 15:58:16 UTC 2014,41.74810556,-87.79888667,267,9
Thu Dec 11 14:22:03 UTC 2014,41.80568861,-87.63679417,382,0
Thu Dec 11 14:38:49 UTC 2014,41.79579528,-87.63783889,85,20
Thu Dec 11 15:03:29 UTC 2014,41.78994306,-87.63157111,171,16
Thu Dec 11 18:31:02 UTC 2014,41.75774333,-87.62539139,358,96
Thu Dec 11 18:41:39 UTC 2014,41.80654917,-87.63945528,282,0
Thu Dec 11 18:59:28 UTC 2014,41.8089425,-87.63819639,85,9
```

Fig. 1. Illustration of probe GPS data and social media data for traffic monitoring.

POI features, and weather information. To combine the preceding multisourced data, we finally propose a coupled matrix and tensor factorization scheme named TCE_R to collaboratively factorize the congestion matrix to low rank matrices with other matrices and tensors formed by the rich side information. The traffic states can then be estimated by completing the sparse traffic congestion matrix through the multiplication of the low rank matrices.

Compared to our previous work [Wang et al. 2016a], we make the following new contributions.

—We model the traffic congestion correlation mining task as a spatiotemporal frequent pattern mining problem from both GPS probe data and social media data. A search tree based pattern mining method is proposed to efficiently discover which road segments geographically close to each other are likely to co-occur traffic congestion.

—We utilize the discovered co-congestion patterns to (1) help better estimate urban traffic congestions, and (2) detect anomalies in the road network by time-sensitive pattern mining.

—To integrate GPS probe data, social media data, road features, POIs, weather information, and traffic correlations, we extend our previously proposed coupled matrix and tensor factorization scheme to more effectively estimate both "recurring" and "nonrecurring" traffic congestions.

—We conduct extensive evaluations on nearly 700 mile arterial roads of Chicago with 0.24 million traffic related tweets and more than 7 million GPS probe data from public passenger buses and regular vehicles. The results demonstrate the effectiveness of the proposed model in urban traffic congestions estimation by comparing it with our previous methods.

The remainder of the article is organized as follows. In Section 2, we give a formal definition of the studied problem and show the framework of our solution. Section 3 introduces how we collect and process data. Section 4 describes how to mine the traffic congestion correlations from historical data. The coupled matrix and tensor factorization schema is presented in Section 5. Evaluations are given in Section 6 followed by related work in Section 7. Finally, we conclude this article in Section 8.

2. PROBLEM DEFINITION AND FRAMEWORK

In this section, we first give some definitions to help us state the studied problem. Then we briefly describe the framework of our method.

Definition 2.1. **An arterial road R_i** [Wang et al. 2015]. An arterial road R_i can be represented as such a tuple $R_i = (name_i, dir_i, \mathbf{L}_i)$, where $name_i$ is the name of the arterial road, dir_i denotes the road direction, and $\mathbf{L}_i = (l_{i,1}, \ldots, l_{i,n})$ is the set of intersections on R_i. Each intersection $l_{i,j}$ contains the exact location information and can be represented as $l_{i,j} = (lat_{i,j}, lon_{i,j})$, where $lat_{i,j}$ and $lon_{i,j}$ represent the latitude and longitude, respectively.

Definition 2.2. **A road segment r_i** [Wang et al. 2015]. A segment r_i of the arterial road R_i is a continuous part of R_i. Formally, we define $r_i = (ID_i, name_i, \mathbf{l}_i)$, where ID_i is the road segment ID, $name_i$ is the name of the arterial road r_i belongs to, and $\mathbf{l}_i$ is a subset of $\mathbf{L}_i$.

Definition 2.3. **Road network $\mathcal{G}$.** A road network $\mathcal{G} = (E, V)$ is comprised of a set of road segments connected to each other in a graph format. $E = \{r_i\}$ is the set of the edges with each edge associated with a road segment, and $V = \{v_i\}$ is the set of the nodes with each node associated with an intersection.

Definition 2.4. **A traffic event tweet e_i** [Wang et al. 2015]. We represent a traffic event tweet e_i as such a tuple $e_i = (c_i, \mathbf{w}_i, t_i)$ where $c_i \in C$ is the traffic event category, $\mathbf{w}_i = (w_{i,1}, \ldots, w_{i,N_{e_i}})$ represents the words mentioning the locations of the event, and t_i denotes the event time.

Definition 2.5. **A GPS probe reading p_i.** We represent a GPS probe reading p_i as such a vector $p_i = \{s, lat, lon, head, t\}$, where s is the vehicle speed, lat is the latitude, lon is the longitude, $head$ is the heading of the probe, and t denotes the time of the probe reading.

Traffic Congestion. Traffic congestion refers to a condition on transport networks that occurs as use increases, and it is characterized by slower speeds, long trip times, and increasing vehicular queueing. As a fuzzy concept, the definition of traffic congestion can vary significantly from time to time and place to place. For example, the concept of congestion in highway can be very different from it on an arterial road in urban area. Therefore, congestion is difficult to define precisely in a mathematical sense [Schrank et al. 2012].

A straightforward definition of traffic congestion is that the average vehicle speed is lower than a predefined threshold. For example, Chicago Transit Authority (CTA for short) defines five-state traffic conditions in downtown Chicago by fully considering the traffic situations in urban area.[1] The five states are *heavy congestion*, *medium-heavy congestion*, *medium*, *light*, and *flow*. As shown in Table I, the corresponding traffic speeds of the five traffic states are 0–10, 10–15, 15–20, 20–25, and over 25 mph, respectively. Note that except for a vary few road segments, speed on arterials of

[1] https://data.cityofchicago.org/api/assets/88B2ABA5-BF4C-4A41-949C-2B11D725ADAB.

Table I. 5 Traffic States Defined by CTA in Downtown Chicago

Traffic states	heavy congestion	medium-heavy	medium	light	flow
Speed (mph)	0–10	10–15	15–20	20–25	>25
Value	1	0.8	0.6	0.4	0.2

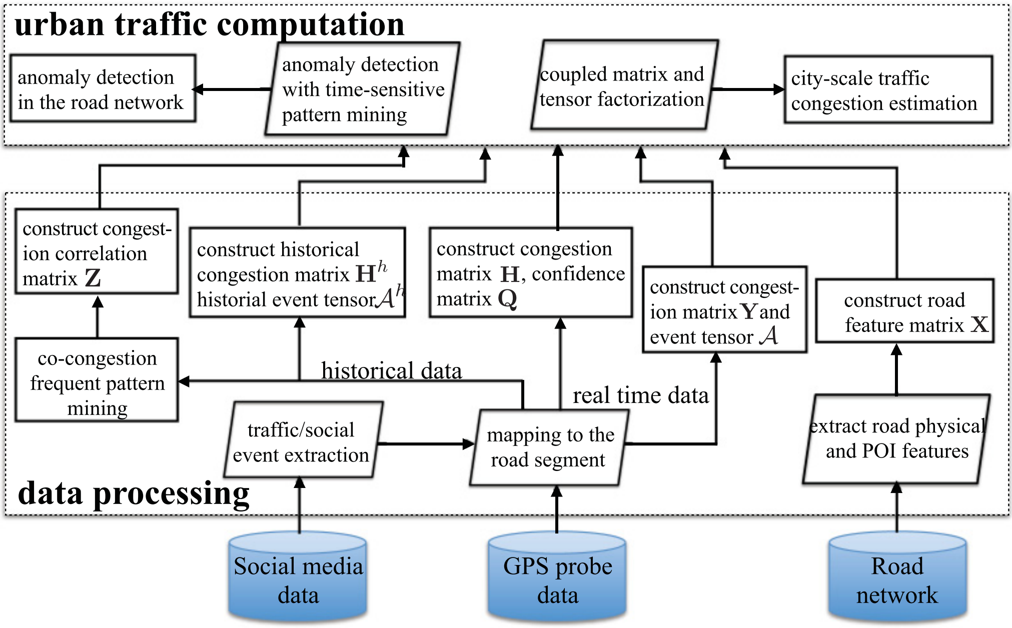

urban traffic computation
anomaly detection in the road network
anomaly detection with time-sensitive pattern mining
coupled matrix and tensor factorization
city-scale traffic congestion estimation
construct congest-ion correlation matrix **Z**
construct historical congestion matrix $\mathbf{H}^h$ historial event tensor $\mathcal{A}^h$
construct congestion matrix **H**, confidence matrix **Q**
construct congest-ion matrix **Y** and event tensor $\mathcal{A}$
construct road feature matrix **X**
co-congestion frequent pattern mining
historical data
real time data
traffic/social event extraction
mapping to the road segment
extract road physical and POI features
data processing
Social media data
GPS probe data
Road network

Fig. 2. Framework of our model.

downtown Chicago is limited to 30 mph by ordinance. To quantitatively distinguish the five states, we assign different values to different states as shown in the table. We assign larger values to worse traffic conditions and smaller values to better conditions. We first assign 1 to the traffic state *heavy congestion*, and set the value difference between two successive traffic states as 0.2. Thus, the values for the remaining four states *medium-heavy*, *medium*, *light*, and *flow* are 0.8, 0.6, 0.4, and 0.2, respectively. In the following parts of this article, we use the five traffic states defined by CTA as a measure of traffic conditions.

Figure 2 shows the framework of our model. One can see that there are mainly three types of data in the framework: social media data, GPS probe data, and the road network data. Based on the road network data, we extract road physical features and POI features on each road segment, and construct the road feature matrix $\mathbf{X}$. From social media data, we first identify the traffic related tweets, and then we extract the event category, location, and time information from each tweet. Based on this information, we construct the congestion matrix $\mathbf{Y}$, and the event tensor $\mathcal{A}$. The two dimensions of $\mathbf{Y}$ are road segment ID and time slot. $y_{ij} = 1$ means that the road segment r_i is in congestion in the jth time slot of a day. The three dimensions of $\mathcal{A}$ are road segment ID, event category, and time slot. $A_{ijk} = 1$ means an event with category k happens on the road segment r_i in the jth time slot of a day. For the probe GPS data, we first map each probe reading to the corresponding road segment based on their longitude and latitude information. Then we construct the congestion matrix $\mathbf{H}$ and the confidence matrix $\mathbf{Q}$. Each entry h_{ij} of $\mathbf{H}$ denotes the traffic state of the road segment r_i in the jth time slot estimated with the probe data. Each entry q_{ij} of $\mathbf{Q}$ denotes how reliable the estimated traffic state h_{ij} is. Note that we have two congestion matrices $\mathbf{Y}$ and $\mathbf{H}$ estimated by tweets and probe readings separately. We also construct the historical congestion probability matrices $\mathbf{Y}^h$ and $\mathbf{H}^h$, which could provide us with the prior knowledge of which road segments are more likely to be in congestion in some time intervals. With a large volume of historical traffic congestion information obtained from the two types of data sources, we also propose a spatial-temporal frequent pattern mining method to discover which road segments are very likely to co-occur congestion. Based on the discovered patterns, we further construct a congestion correlation matrix $\mathbf{Z}$. Each entry z_{ij} of $\mathbf{Z}$ denotes the probability of road segments r_i and r_j co-occurring congestion.

In this article, we focus on utilizing the preceding information to address the following two tasks: city-scale traffic congestion estimation and anomaly detection in the arterial network. As the congestion matrices $\mathbf{Y}$ and $\mathbf{H}$ are both very sparse and most entries are unknown, our goal is to perform matrix completion by utilizing coupled matrix and tensor factorization which makes full use of rich information. We also perform anomaly detection with a time-sensitive pattern mining algorithm to detect anomalies in a road network. We will elaborate each part of the framework in the following sections.

3. DATA COLLECTION AND PROCESSING

In this section, we describe how we collect and process traffic related data from multiple sources in downtown Chicago. We first show how we collect and process traffic related tweets and GPS probe data. We next briefly introduce how we extract road features, POIs, social events, and weather as auxiliary information.

3.1. Twitter Data Collection

We collect traffic event tweets from the following two types of accounts as in Wang et al. [2015, 2016a]: the Twitter accounts operated by official traffic departments and general user accounts.

Traffic Authority Account. Traffic authority Twitter accounts refer to the Twitter accounts that specialize in posting traffic related information. Such accounts are mostly operated by official transportation departments. Tweets posted by these accounts are formal and easy to process, and the exact location and time information are explicitly given. Taking the tweet "*Heavy Traffic on NB Western: Fullerton to Kennedy Expy. 06:15 pm 02/13/2015*" as an example, we can easily extract the road segment, traffic event category, and time information by key words matching. We identify 10 such Twitter accounts that report real-time traffic information of Chicago: *ChicagoDrives*, *ChiTraTracker*, *roadnowChicago*, *traffic_Chicago*, *IDOT_Illinois*, *WGNtraffic*, *TotalTrafficCHI*, *GeoTrafficChi*, *roadnowil*, and *rosalindrossi*.

General Sensor User. We also collect traffic related tweets from regular users. We selected 100,000 Twitter users registered in Chicago, and crawled more than 32.3 million tweets posted by them. Next, two major steps are conducted for data preprocessing. (1) *Traffic Event Tweets Identification.* We identify traffic event tweets from all the crawled tweets which match at least one term of the predefined vocabularies: "stuck," "congestion," "jam," "crowded," "pedestrian," "driver," "accident," "crash," "road blocked," "road construction," "slow traffic," "heavy traffic," "bad traffic," and "disabled vehicle." We first select the tweets that contain at least one of the keywords mentioned previously by keywords matching. Based on the keywords contained in the tweets, we can also identify the traffic event category. (2) *Tweet Geocoding.* We then geocode tweets to the road segments. For the tweets collected from traffic authority accounts, we can very easily locate the road segment where the traffic event occurs as such tweets usually have fixed formats. Taking the tweet "*Heavy Traffic on NB Western: Fullerton to*

Kennedy Expy. 06:15 pm 02/13/2015" as an example again, we can extract the road segment through the pattern {*on R1: R2 to R3*}, where $R1$, $R2$, and $R3$ are street names. For the tweets collected from general sensor users, it is harder to locate them. A small proportion (about 5%) of such tweets are geo-tagged. Thus, we can precisely map them to the corresponding road segment. For most tweets without geotags, we extract the name of the streets and landmarks information from the tweets. Taking the tweet "Bad traffic at Roosevelt this morning" as an example, we can extract the street name "Roosevelt" by matching it with all the Chicago street names. Thus, we can map the traffic event to the road segment "Roosevelt." If only one street name is mentioned in the tweet as in the example, we roughly consider the traffic event will influence the entire street. If two street names are mentioned and forms the pattern *{on R1...R2}*, we consider the traffic event happening on the road segment of street $R1$ that is close to the intersection between $R1$ and $R2$.

To examine how accurately we can extract the traffic event tweets, we randomly sample 1,000 processed tweets and manually label each of them to check whether it really reports a traffic event. The result shows that 38 tweets are falsely extracted, and the false alarm rate is 3.8% on the small sampled dataset. It shows that the previous two steps can very accurately identify traffic event relevant tweets. In all, we obtain 245,568 traffic event tweets from April 2014 to December 2014, around 80% of which are collected from traffic authority accounts. Each tweet reports a traffic event. 163,742 of them are related to slow traffic, 77,454 are related to accident, and the remaining 4,372 report other traffic events such as road construction and road closure.

3.2. Modeling Road Features, Social Events, and Weather Information

Road features are widely used for traffic estimation and prediction [Wang et al. 2015; Shang et al. 2014; Zheng et al. 2014b]. In this article, we use the following two types of road features: road physical features, and POI features. Social events [Wang et al. 2015] and weather can also affect traffic conditions, and thus we also extract them as important complementary information.

3.2.1. Road Physical Features and POIs. We extract the following physical features of a road segment: the road segment length $r.len$, the number of lanes $r.lane$, whether it is a one-way road $r.way$, the road heading $r.dir$, the number of intersections $r.inter$, the number of bus stops $r.stop$, and the area $r.area$ of the city that the road segment belongs to. Chicago Transit Authority divides the entire Chicago city into 29 traffic regions. Each region is comprised of two or three community areas with comparable traffic patterns. These features of a road segment r are modeled as a vector $f_r = (r.len, r.lane, r.way, r.dir, r.inter, r.stop, r.area)$.

A POI is a venue in a physical world that someone may find useful or interesting, like a shopping mall, a theater, or a hospital [Zheng et al. 2014b]. Each POI is associated with many attributes including the name, address, coordinates, and categories. In this article, we extract various types of POIs near each road segment and formulate them as a POI feature vector f_p^r. Each element $f_p^r(i)$ of f_p^r denotes the number of the POIs with type i.

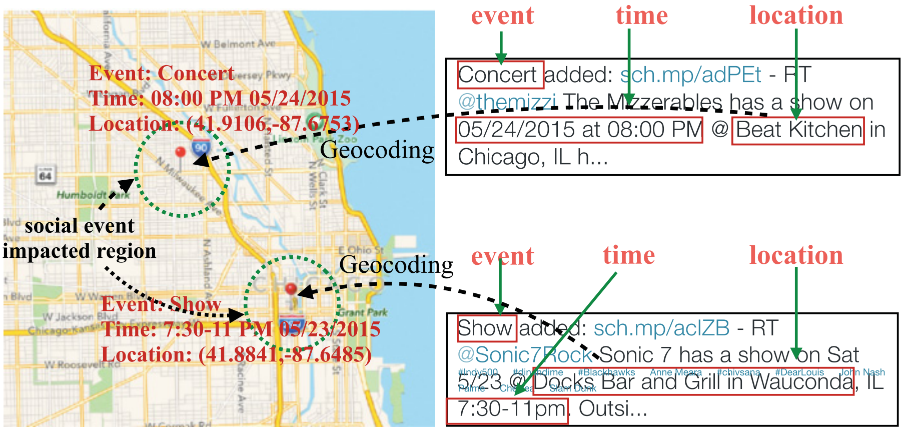
3.2.2. Social Events. Chicago Events is a Twitter account that focuses on posting various social events information in Chicago. Taking the tweet *"Concert added: sch.mp/adPEt - RT@themizzi The Mizzerables has a show on 05/24/2015 at 08:00 PM @ Beat Kitchen in Chicago IL."* as an example, we can easily extract the event type (*Concert*), time (*05/24/2015 at 08:00 PN*), and location (*Beat Kitchen in Chicago IL*) information from each posted tweet. Most such tweets have the fixed format as shown in the example. Thus, it is convenient for us to extract the event type, location, and

Table II. Description on Rich Features

Road physical features	*road segment length*, *number of lanes*, *one-way road*, *heading*, *number of intersections*, *number of bus stops*, *area of the city*
POI features	*Schools*, *Hospitals*, *Museums*, *Libraries*, *Parks*, *Police Stations*, *Parking zones*, *Market*, *Malls*, *Gas Station*, *Bus Station*, *Bars*, *Churches*, *Stadium*, *Restaurant*, *Cinema*

Types of social events	Parties: {*Party*, *Festival*, *Nightlife*, *Meeting*}
	Shows:{*Show*, *Live music*, *Exhibition*}
	Sports: {*Game*, *Concert*, *Performance*}

Weather information	*Storm*, *Freezing*, *Rain*, *Snow*, *Fog*, *Snow cover*, *High winds*


event
time
location
Event: Concert
Time: 08:00 PM 05/24/2015
Location: (41.9106,-87.6753)
Geocoding
social event
impacted region
Event: Show
Time: 7:30-11 PM 05/23/2015
Location: (41.8841,-87.6485)
Geocoding
event
time
location

Concert added: sch.mp/adPEt - RT @themizzi The Mizzerables has a show on 05/24/2015 at 08:00 PM @ Beat Kitchen in Chicago, IL h...

Show added: sch.mp/acIZB - RT @Sonic7Rock Sonic 7 has a show on Sat 5/23 @ Docks Bar and Grill in Wauconda, IL 7:30-11pm. Outsi...

Fig. 3. Social events extraction and geocoding.

time information. In all we crawled 5,196 social events of different types in Chicago. The types of these social events are shown in Table II. To alleviate the data sparsity issue, we group the social events into three types: *parties*, *shows*, and *sports*.

We formulate a social event se as such a tuple $se = \{c_{se}, p_{se}, t_{se}\}$, where c_{se} is the event category, p_{se} is the event location, and t_{se} is the event time. For each social event tweet, we first extract the event type, time, and place information, and then identify the latitude and longitude information by geocoding. Figure 3 gives an example to show how we extract event information from tweets and geocode them. To model the impact of the social events on traffic, we propose to use a two-dimensional Gaussian model to measure the impact intensity of the event on the nearby road segments based on their Euclidean distance, that is,

$$\begin{aligned} I(r_i, se_j) &= f(loc_{r_i}; loc_{se_j}, \Sigma_{se_j}) \\ &= \frac{1}{2\pi |\Sigma_{se_j}|^{\frac{1}{2}}} e^{-\frac{1}{2}(loc_{r_i} - loc_{se_j})^T \Sigma_{se_j}^{-1} (loc_{r_i} - loc_{se_j})}, \end{aligned}$$

where r_i is the road segment, se_j is the social event, loc_{r_i} is the location of r_i, and loc_{se_j} is the location of se_j. Based on this model, the impact of the social event se_j on road segment r_i decreases with the increase of their distance loc_{r_i}-loc_{se_j}. The green dashed circles in Figure 3 represent the impacted regions of the social events. Given a road segment r_i, there might be several social events happening near it in the same time interval, and the overall impact of all the events on the traffic conditions of r_i can be

Table III. Notations and Their Meanings

Notation	Meaning
$minsup$	the threshold of support
$minconf$	the threshold of confidence
k	spatial constraint, number of hops from a road segment to another
η	temporal constraint, the time difference in hours of two traffic events
(k, η)	the spatiotemporal constraint
$N(r_i, k)$	k-hop neighborhood of the road segment r_i
G_S	the connectivity graph formed by the road segments set S
$Pr(r_i, r_j; k, \eta)$	confidence of congestion co-occurrence pattern $\{r_i, r_j\}$

calculated by

$$I(r_i) = \sum_j I(r_i, se_j).$$

3.2.3. Weather Information. Bad weather condition is an important factor causing "nonrecurring" congestions. About 15% "nonrecurring" traffic congestions in the United States are caused by bad weather conditions like heavy snow, storm, and fog. Although the effect of bad weather on traffic congestion is widely studied [Golob and Recker 2003], how to incorporate it for estimating traffic conditions is not well studied by previous works.

In this article, we consider the following weather conditions in a day: *Storm*, *Freezing*, *Rain*, *Snow*, *Fog*, *Snow cover*, *High winds*. We formulate the weather condition in a day as a vector f_{wea} with each entry $f_{sea}(i)$ denoting whether there exists an extreme weather as listed previously. Given two weather condition vectors f^i_{wea} and f^j_{wea}, we can calculate their cosine similarity: $Sim(f^i_{wea}, f^j_{wea}) = \frac{\sum_{k=1}^{n} f^i_{wea}(k) f^j_{wea}(k)}{\sqrt{\sum_{k=1}^{n}(f^i_{wea}(k))^2}\sqrt{\sum_{k=1}^{n}(f^j_{wea}(k))^2}}$.

Table II shows the extracted rich features including road physical features, POI features, social events, and weather information.

3.3. Probe Data Processing

We have more than 2.4 million GPS probe readings in the period of December 2014 in Chicago. These probes cover most arterial roads of downtown Chicago. Each probe reading contains the following key information: time, latitude, longitude, heading, and speed. As the road segment information cannot be directly obtained from the probe readings, we first need to map the probe readings to the corresponding road segments. To correctly map the probe location to the road segment, we calculate the distances from reported locations of the probe to the locations of the nearby road segments, and we select the road segment with the shortest distance to the reported probe reading as the one that the probe is on.

4. TRAFFIC CONGESTION CORRELATION MINING

Traffic congestions happening on road segments that are geographically close to each other can be correlated. A congested road segment could cause slow traffic on the nearby road segments. Mining the traffic congestion correlations could help us find the traffic bottleneck in a transportation network and guide us to improve traffic constructions. It can also help us better estimate traffic congestions when the available probe data is sparse. In this section, we model the traffic congestion correlation mining problem as a spatiotemporal frequent pattern mining task. We will first give some definitions to help us state the problem. Then we will introduce a search tree based method to efficiently discover the frequent co-congestion patterns. Before we describe the method, we first give some notations and show their meanings in Table III.

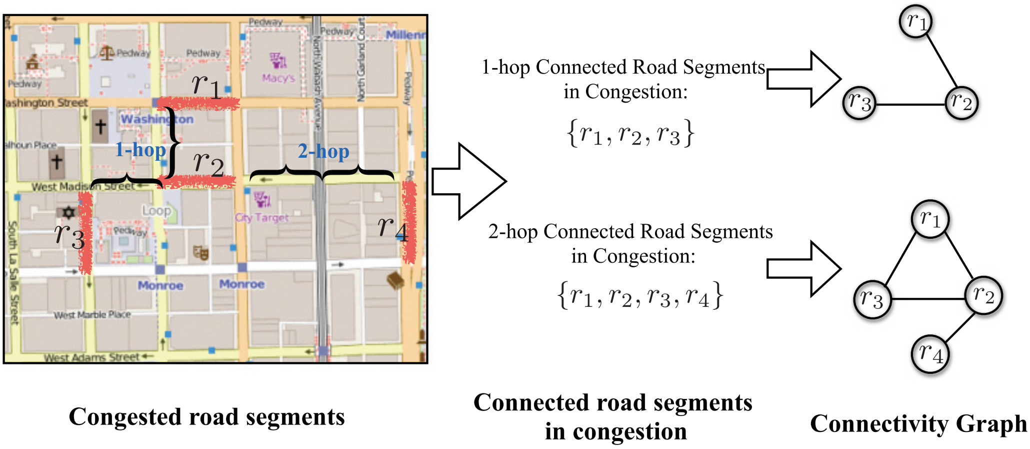


Fig. 4. Illustration of connected road segments in congestion and the corresponding connectivity graph.

4.1. Problem Formulation

Definition 4.1. **k-hop neighborhood $N(r_i, k)$ of a road segment r_i**. The k-hop neighborhood $N(r_i, k)$ of a road segment r_i is the set of road segments that can reach r_i within k hops. Given two road segments r_i and r_j, we say r_i can reach r_j in k hops if there exist two intersections p_i on r_i and p_j on r_j such that p_i reaches p_j in no more than k hops.

Definition 4.2. **Congestion co-occurrence of road segments r_i and r_j**. Given two road segments $\{r_i, r_j\}$, we call them co-occur in the congestion events $\{e_i, e_j\}$ if the following constraints are satisfied: (1) the road segment r_i of e_i is the k-hop neighborhood of the road segment r_j of e_j, that is, $r_i \in N(r_j, k)$; and (2) the difference of the event time t_i, t_j is less than η, that is, $|t_i - t_j| < \eta$, where k and η are the predefined spatiotemporal thresholds.

Definition 4.3. **k-hop Connected Road Segments in Congestion**. Given a k-hop distance threshold and a subset of road segments $S \subseteq R$ that co-occur congestion with the predefined spatiotemporal thresholds η, S is k-hop connected if $\forall r_i \in R, \exists r' \in R - r_i \ s.t. \ r_i \in N(r', k)$.

Definition 4.4. **k-hop Connectivity Graph**. Given a k-hop connected road segment S in congestion, the connectivity graph G_S is constructed as follows: (1) each vertex in G_S corresponds to a road segment r_i; and (2) there is an edge between two vertices r_i and r_j if $r_i \in N(r_j, k)$.

Figure 4 shows an example of the connected road segments in congestion and the corresponding connectivity graph. The very left figure is a partial map of downtown Chicago. As shown in the map, there are four road segments $\{r_1, r_2, r_3, r_4\}$ co-occur congestion in a particular time interval t. The distance between r_1 and r_2 is 1-hop, the distance between r_2 and r_3 is also 1-hop, and the distance between r_2 and r_4 is 2-hop. Based on the congested road segment set $\{r_1, r_2, r_3, r_4\}$ and their distances, we can extract the 1-hop connected road segments in congestion $\{r_1, r_2, r_3\}$ and the 2-hop connected road segments in congestion $\{r_1, r_2, r_3, r_4\}$. With the connected road segments, we can further construct the k-hop connectivity graph as shown in the very right part of the figure. Taking the 1-hop connected road segments in congestion as an example, as the distance between road segment r_1 and t_2 is 1-hop, there exists an edge between the two nodes. Similarly, there is also an edge between r_2 and r_3.

Definition 4.5. **Support of road segments $r = \{r_i, r_j\}$ in congestion**. Given a set of road segment congestion event E. The support of the road segments $r = \{r_i, r_j\}$ in congestion is defined as the possibility that a member of E whose road segment contains at least one road segment in r.

Definition 4.6. **Confidence of congestion co-occurrence of road segments $r = \{r_i, r_j\}$**. Given the spatiotemporal thresholds k and η. The confidence of congestion co-occurrence of road segments $r = \{r_i, r_j\}$ is the probability of the two road segments co-occurring congestion in the congestion event database E. It can be calculated by

$$Pr(\{r_i, r_j\}; k, \eta) = \frac{cardinality(\{r_i, r_j\}; k, \eta)}{cardinality(r_i \cup r_j)}.$$

Here $cardinality(\{r_i, r_j\}; k, \eta)$ denotes how many times that road segments r_i and r_j co-occur congestion in the congestion event database E, and $cardinality(r_i \cup r_j)$ denotes the times that at least one of the two road segments occur congestion.

Definition 4.7. **(k, η) neighborhood co-congestion patterns of road segments.** A (k, η) neighborhood co-congestion pattern of two road segments r_i and r_j is of the form: $\{r_i, r_j\}(minsup, minconf)$, where k and η are the spatiotemporal thresholds, and $minsup$ and $minconf$ are user-specified minimum support and minimum confidence.

The preceding definitions can be easily extended to the case that the pattern contains multiple road segments. Due to space limitation, we omit the descriptions on the patterns with more road segments and one can refer to Wang et al. [2016a] for more details.

4.2. Frequent Co-Congestion Patterns Generation

With the definitions in the last section, a traditional frequent pattern mining algorithm can be utilized to discover all the co-congestion patterns, such as the Apriori algorithm. However, Apriori method is computationally infeasible to address this problem for a large arterial road network. To fully explore the spatiotemporal constraints to efficiently prune the search space, we will introduce a search tree based method to more efficiently discover the co-congestion patterns.

To efficiently discover the spatial coevolving patterns in geo-sensory data, Zhang et al. proposed a search tree based method to prune the infrequent patterns by fully exploring the spatiotemporal constraints [Zhang et al. 2015]. Motivated by their work, we also construct a search tree which incorporates the spatial constraint to organize the search space. We search possible frequent co-congestion patterns on the constructed search tree with the temporal constraint and prune the branches that are infrequent in the very early stage. Specifically, our search tree based frequent pattern mining algorithm for the co-congestion road segments mainly contains two stages. In the first stage, we construct the connectivity graph based on a large number of traffic condition snapshots in each time interval. In the second stage, we generate a search tree based on the constructed connectivity graph, and conduct a top-down frequent pattern search on the tree. Next, we will introduce the two stages in detail.

4.2.1. Connectivity Graph Construction. We first introduce the first stage of the algorithm: connectivity graph construction. Figure 5 shows the process of generating a connectivity graph. Given the temporal constraint η and the spatial constraint k, we can extract an all k-hop connectivity graph for each traffic condition snapshot of the arterial network as shown in the left part of Figure 5. In the example shown in this figure, three snapshots of k-hop connectivity subgraphs are extracted in each time interval. Then we merge all the k-hop connectivity subgraphs to construct a large connectivity graph for

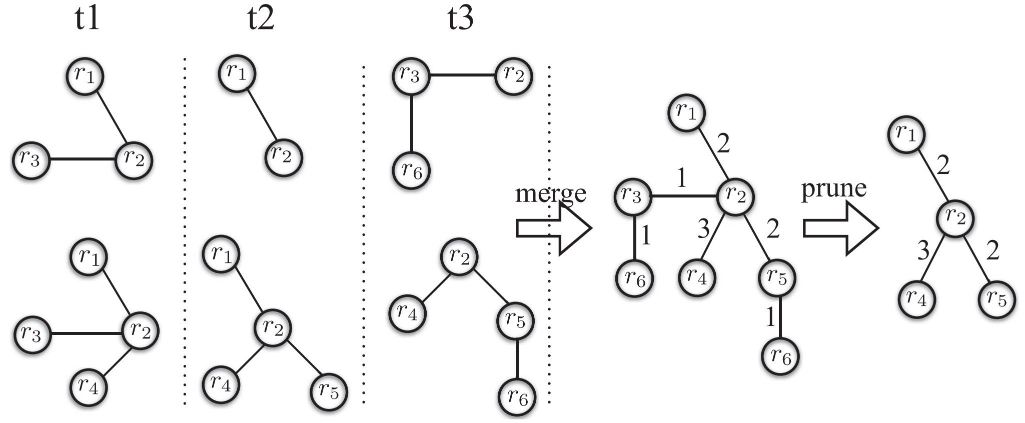


Fig. 5. Generating connectivity graph.

all the time intervals. Meanwhile, each edge of the graph is associated with a number denoting the times of the nodes co-occurring congestion. As shown in the middle of Figure 5, nodes r_1 and r_2 co-occur congestion for two times in the three time intervals, and thus number 2 is assigned to the edge connecting them. Finally, we further prune the large graph based on the support threshold of the road segment in congestion. For example, given the support threshold 2, we can prune all the edges with the weight 1. The very right figure of Figure 5 shows the pruned connected graph.

The advantages of the connectivity graph construction step are twofold. First, the final connectivity graph is assembled from many subgraphs in different time intervals. Thus, the spatiotemporal constraint is incorporated into the graph, since we only consider the k-hop co-congestions occurring in the same time interval. In addition, we further prune the edges whose weights are smaller than the co-congestion support threshold. Thus, the final connectivity graph can be considered as a preliminary step to prune the searching space. Second, the constructed connectivity graph is used to generate the search tree in the second step, on which all the frequent co-congestion patterns can be efficiently searched with a top-down searching algorithm.

4.2.2. Frequent Co-Congestion Patterns Mining with Search Tree. Before we introduce how to construct the search tree based on the connectivity graph, we first give some definitions following the work [Zhang et al. 2015].

Definition 4.8. **Parent**. Given a k-hop connectivity graph G, let Y be a size-k+1 connected component in G. Given a vertex order $\mathcal{O}$, the roll-up operation on Y removes the first possible node s in Y based on the $\mathcal{O}$ that makes the size-k set $X = Y - s$ still connected. We call X the parent of Y, and Y is the child of X.

As shown in Figure 6, assume the node order is $\mathcal{O} = (r_1, r_2, r_3, r_4, r_5, r_6)$. The $X = (r_2, r_3, r_4)$ is the parent of the connected component $Y = (r_1, r_2, r_3, r_4)$, because r_1 is the first node removed to which the component (r_2, r_3, r_4) is also connected according to the node order $\mathcal{O}$.

A good property of search tree constructed from the connectivity graph is that if there are no frequent co-congestion patterns over a connected component X of G, we can safely prune X and all its descendants without missing any patterns [Zhang et al. 2015]. As an effective method to organize the entire search space in a hierarchical way, we can efficiently discover all the frequent co-congestion patterns and prune the search space by scanning the entire search tree.

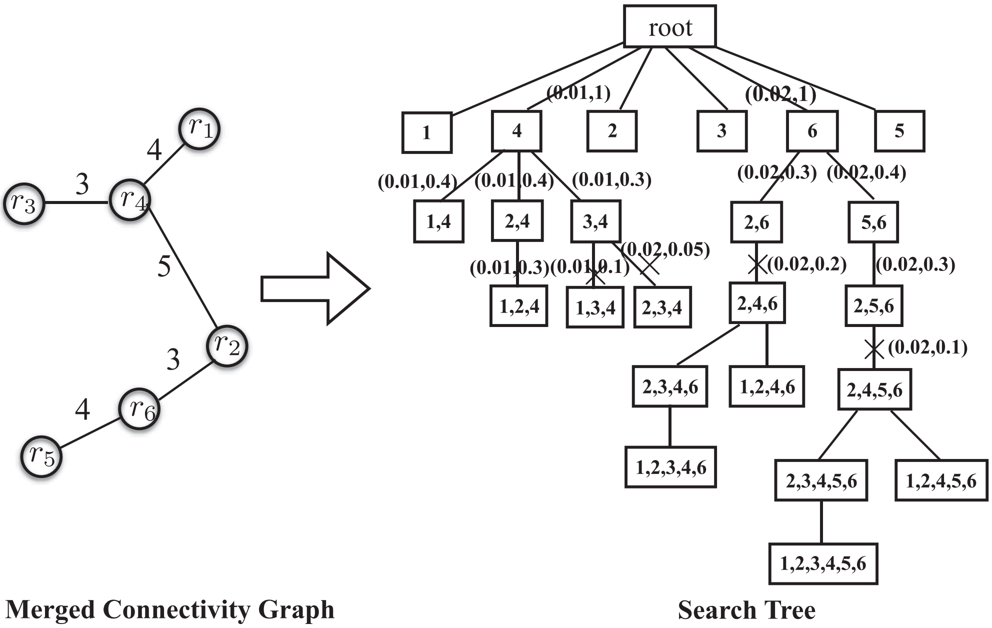


Fig. 6. Illustration of extracting the search tree from the connectivity graph.

Figure 6 shows a search tree (right) constructed from a connectivity graph (left). The root node of the tree is empty set $\emptyset$. The tree can be constructed in both a depth-first and width-first way. However, we will show that the search tree organizes the connected components conceptually. We do not need to really construct the tree beforehand. Instead, we can perform depth-first construction from the root node, and only visit the nodes that are frequent co-congestion patterns. As shown in the right part of Figure 6, suppose the $(minsup, minconf)$ is set to $(0.01, 0.3)$. The support and confidence of pattern $\{4\}$ is 0.01 and 1, which is larger than the threshold. We further search all the child nodes $\{1, 4\}$, $\{2, 4\}$, and $\{3, 4\}$. We find that the three larger patterns are also frequent, and we continue the depth-first search. The node $\{1, 4\}$ has no child node, and thus the search on this branch stops. The child node of node $\{2, 4\}$ is $\{1, 2, 4\}$. Although the pattern $\{1, 2, 4\}$ is also frequent with the support and confidence $(0.01, 0.3)$, it does not have child node, and thus the search on this branch also stops. The child nodes of node $\{3, 4\}$ are $\{1, 3, 4\}$ and $\{2, 3, 4\}$. As there are not frequent patterns, we prune the two branches, and the search on this branch also stops.

4.2.3. Constructing the Traffic Congestion Correlation Matrix. Based on the discovered frequent co-congestion patterns, we construct the traffic congestion correlation matrix $\mathbf{Z}$. For each pattern $\{r_i, \ldots, r_m\}(minsup, minconf)$, we use their confidence to represent the correlation between each pair of road segments in the pattern. For example, given the pattern $\{r_i, r_j, r_k\}(minsup, minconf)$ and the confidence value $conf$, we set $z_{ij} = conf$, $z_{ik} = conf$, and $z_{jk} = conf$.

5. TCE_R: COUPLED TENSOR AND MATRIX FACTORIZATION TO INTEGRATE RICH INFORMATION

In this section, we will introduce a coupled matrix and tensor factorization schema to integrate the previously described rich information for estimating traffic congestions. We first will give a review of some notations and tensor operations used in the article. Then we will describe how we construct the congestion matrix and the confidence matrix based on the sparse GPS probe data. Next, we will propose the coupled matrix and tensor factorization method. Finally, we will introduce how to incorporate the historical data into the model to further alleviate the data sparsity issue.

5.1. Notations and Preliminaries

As our model uses tensor factorization techniques to facilitate matrix factorization. We first give a quick review of some tensor notations and operations, and for more details one can refer to Kolda and Bader [2009].

Tensors are higher-order arrays that generalize the notions of vectors and matrices. The order of a tensor is the number of dimensions, also known as ways or modes. In this article, we use the third-order tensor. Scalars are denoted by lowercase letters, for example, a. Vectors are denoted by boldface lowercase letters, for example, $\mathbf{a}$. Matrices are denoted by boldface capital letters, for example, $\mathbf{X}$. Tensors are denoted by calligraphic letters, for example, $\mathcal{A}$. The ith entry of a vector $\mathbf{a}$ is denoted by a_i, element (i, j) of a matrix $\mathbf{X}$ is denoted by x_{ij}, and element (i, j, k) of a third-order tensor $\mathcal{A}$ is denoted by a_{ijk}. The ith row and the jth column of a matrix $\mathbf{X}$ are denoted by $\mathbf{x}_{i:}$ and $\mathbf{x}_{:j}$, respectively.

The *norm* of a tensor $\mathcal{A} \in \mathbb{R}^{N\times M\times L}$ is defined as

$$\|\mathcal{A}\| = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{L} a_{ijk}^2}.$$

This is analogous to the matrix *Frobenius norm*, which is denoted $\|\mathbf{X}\|$ for a matrix $\mathbf{X}$.

The *outer product* of two vectors $\mathbf{a} \in \mathbb{R}^N$ and $\mathbf{b} \in \mathbb{R}^M$, denoted by $\mathbf{a} \circ \mathbf{b}$, is a matrix of size $N \times M$ with the elements $\left(\mathbf{a} \circ \mathbf{b}\right)_{ij} = a_i b_j$.

The *n-mode product* of a tensor $\mathcal{C} \in \mathbb{R}^{I_1\times I_2\cdots\times I_N}$ with a matrix $\mathbf{U} \in \mathbf{R}^{I_n\times J}$, denoted by $\mathcal{C} \times_n \mathbf{U}$, is a tensor of size $I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$ with the elements $(\mathcal{C} \times_n \mathbf{U})_{i_1\cdots i_{n-1} j i_{n+1}\cdots i_N} = \sum_{i_n=1}^{I_n} a_{i_1 i_2\cdots i_N} u_{i_n j}$.

The *Tucker factorization* of a tensor $\mathcal{A} \in \mathbb{R}^{N\times M\times L}$ is defined as

$$\mathcal{A} = \mathcal{C} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W} = \sum_{r=1}^{R}\sum_{s=1}^{S}\sum_{t=1}^{T} c_{rst}\mathbf{u}_n \circ \mathbf{v}_m \circ \mathbf{w}_l,$$

where $\mathbf{U} \in \mathbb{R}^{N\times R}, \mathbf{V} \in \mathbb{R}^{M\times S}$, and $\mathbf{W} \in \mathbb{R}^{L\times T}$ are the factor matrices. The tensor $\mathcal{C} \in \mathbb{R}^{R\times S\times T}$ is the core tensor and its entries show the level of interaction between the different components.

5.2. Modeling Sparse GPS Probe Data

In this section, we will introduce how to model the sparse GPS probe data by constructing the confidence matrix $\mathbf{Q}$ and the probe congestion matrix $\mathbf{H}$. Given a road segment r_i and the time interval t, assume the set of GPS probe readings is $\{s_1, s_2 \ldots, s_n\}$. With the probe readings, we estimate the traffic conditions of r_i in t as follows. We first average the probe speed $s_{average} = \frac{1}{n}\sum_{i=1}^{n} s_i$, and then estimate the traffic conditions based on the estimated average probe speed and five-state traffic conditions defined by CTA shown in Table I. We then fill the probe congestion matrix $\mathbf{H}$ with the traffic state values. Each entry $h_{i,j}$ of $\mathbf{H}$ denotes the congestion state of the road segment r_i in time interval j. For example, $h_{i,j} = 1.0$ means that the probe speed on the road segment r_i in j is less than 10 miles per hour, and thus it is in heavy congestion state. For the road segment without any probe readings in some time intervals, the corresponding entries in matrix $\mathbf{H}$ are empty and need to be estimated.

As the probe readings are very sparse, a large proportion of entries of $\mathbf{H}$ is estimated by only one or two probe readings. Thus, the estimated traffic states are mostly unreliable. We consider that the reliability of the estimated traffic states is related to the sparsity of probe readings. More readings imply a more reliable estimation; otherwise the estimation is unreliable. To quantitatively measure the reliability of the traffic states estimated by probes, we also construct a confidence matrix $\mathbf{Q}$. Each entry q_{ij} of $\mathbf{Q}$ is calculated by such a sigmoid function

$$q_{ij}(n) = \frac{1}{1 + e^{n - Cardinality(p_{ij})}}, \tag{1}$$

where n is a predefined threshold of the probe reading size. In this article, we set n to 3. One can see that more probe readings can result in a larger q_{ij}, which means the estimated traffic state is more reliable.

5.3. Coupled Matrix and Tensor Factorization Model

The insight of using matrix and tensor factorization to estimate urban traffic congestion is as follows: given the very sparse road congestion matrices $\mathbf{Y}$ and $\mathbf{H}$ estimated by tweets and probe readings, respectively, try to complete the two matrices by factorizing each into two low rank latent matrices $\mathbf{U}$ and $\mathbf{V}$. Before we introduce our method, we first describe some symbols as follows. $\mathcal{A} \in \mathbb{R}^{N \times M \times L}$ represents the event tensor, $\mathbf{X} \in \mathbb{R}^{N \times K}$ represents the road feature matrix, $\mathbf{Y} \in \mathbb{R}^{N \times M}$ is the congestion matrix, $\mathbf{Q} \in \mathbb{R}^{N \times M}$ is the confidence matrix, $\mathbf{H} \in \mathbb{R}^{N \times M}$ is the probe congestion matrix, and $\mathbf{Z} \in \mathbb{R}^{N \times N}$ is the congestion correlation matrix. Here N is the number of road segments, M is the number of time slots (hour) per day, K is the number of road features, and L is the number of event categories.

As the congestion matrices $\mathbf{Y}$ and $\mathbf{H}$ are both very sparse, factorizing them directly usually cannot achieve promising results. To fully utilize other traffic related information including weather, social events, and road features, we utilize the context-aware matrix factorization schema, which can effectively integrate multiple related matrices to improve the performance of matrix factorization [Karatzoglou et al. 2010; Zheng et al. 2014b]. Motivated by this method, we factorize the congestion matrices $\mathbf{H}$ and $\mathbf{H}$ collaboratively with the road feature matrix $\mathbf{X}$, congestion correlation matrix $\mathbf{Z}$, and event tensor $\mathcal{A}$. The road feature matrix $\mathbf{X}$ can be factorized into the multiplication of two matrices, $\mathbf{X} = \mathbf{U} \times \mathbf{F}$, where $\mathbf{U} \in \mathbb{R}^{N \times R}$ and $\mathbf{F} \in \mathbb{R}^{R \times K}$ are low rank latent factors for road segments and geographical features, respectively. We factorize the road feature matrix $\mathbf{X}$ based on the idea that road segments with similar road features are more likely to present similar traffic conditions. Likewise, the congestion matrices $\mathbf{Y}$ and $\mathbf{H}$ can both be factorized into the multiplication of two matrices, $\mathbf{Y} = \mathbf{U} \times \mathbf{V}^{\mathrm{T}}$ and $\mathbf{H} = \mathbf{U} \times \mathbf{V}^{\mathrm{T}}$, where $\mathbf{V} \in \mathbb{R}^{M \times R}$ is a low rank latent factor matrix for time slots. We assume the congestion matrix $\mathbf{Y}$ and the probe congestion matrix $\mathbf{H}$ share the same low rank latent matrices $\mathbf{U}$ and $\mathbf{V}$, because the final congestion matrix $\mathbf{Y}$ should be similar to the probe congestion matrix $\mathbf{H}$ as much as possible. The event tensor can be factorized as $\mathcal{A} = \mathcal{C} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$, where $\mathbf{W} \in \mathbb{R}^{L \times T}$ is a low rank latent factor matrix for event categories. The idea is that road segments with similar traffic events occurring in the same time interval are more likely to present similar traffic conditions.

The objective function of the context-aware matrix and tensor factorization schema is as follows:

$$\begin{aligned} &\mathcal{L}(\mathbf{U}, \mathbf{V}, \mathbf{W}, \mathcal{C}, \mathbf{F}) \\ &= \frac{1}{2}(||\mathbf{P} \odot (\mathbf{Y} - \mathbf{U}\mathbf{V}^{\mathrm{T}})||^2 + ||\mathbf{Q} \odot (\mathbf{H} - \mathbf{U}\mathbf{V}^{\mathrm{T}})||^2) + \frac{\lambda_1}{2}||\mathbf{X} - \mathbf{U}\mathbf{F}||^2 + \frac{\lambda_2}{2}\mathrm{tr}(\mathbf{U}^{\mathrm{T}}\mathbf{L}_z\mathbf{U}) \\ &+ \frac{\lambda_3}{2}||\mathcal{R} \odot (\mathcal{A} - \mathcal{C} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W})||^2 + \frac{\lambda_4}{2}(||\mathbf{U}||^2 + ||\mathbf{V}||^2 + ||\mathbf{W}||^2 + ||\mathcal{C}||^2 + ||\mathbf{F}||^2), \end{aligned} \tag{2}$$

where $\odot$ represents the Hadamard product of two matrices, and $\mathrm{tr}(\cdot)$ denotes the matrix trace. $\mathbf{P}$ is an indication matrix for all the nonzero entries in $\mathbf{Y}$, that is, $p_{ij} = 1$ if and only if $y_{ij} > 0$, namely, there is at least one tweet that reports traffic congestion on road segment r_i in time interval t_j. Similarly, $\mathbf{Q}$ is an indication matrix for the nonzero entries in $\mathbf{H}$, and $\mathcal{R}$ is an indication tensor for the nonzero entries in $\mathcal{A}$, that is, $r_{ijk} = 1$ if and only if $a_{ijk} > 0$, namely, there is at least one tweet that reports a traffic or social event. $||\mathbf{P} \odot (\mathbf{Y} - \mathbf{U}\mathbf{V}^{\mathrm{T}})||^2$ is to control the factorization error of $\mathbf{Y}$. $||\mathbf{Q} \odot (\mathbf{H} - \mathbf{U}\mathbf{V})||^2$ is to control the factorization error of $\mathbf{H}$. $||\mathcal{A} - \mathcal{C} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}||^2$ is to control the factorization error of tensor $\mathcal{A}$. $||\mathbf{X} - \mathbf{U}\mathbf{F}||^2$ is to control the factorization error of $\mathbf{X}$. $||\mathbf{U}||^2 + ||\mathbf{V}||^2 + ||\mathbf{W}||^2 + ||\mathcal{C}||^2 + ||\mathbf{F}||^2$ is a regularization penalty to avoid overfitting; λ_1, λ_2, λ_3, and λ_4 are parameters to control the contribution of each part. $\mathbf{L}_z = \mathbf{D} - \mathbf{Z}$ is the Laplacian matrix of the road segment congestion correlation graph in which $\mathbf{D}$ is a diagonal matrix with diagonal entries $d_{ii} = \sum_i z_{ij}$. $\mathrm{tr}(\mathbf{U}^{\mathrm{T}}\mathbf{L}_z\mathbf{U})$ is used to guarantee two road segments r_i and r_j with a higher congestion correlation (i.e., z_{ij} is big) should also have a closer distance between the vector $\mathbf{u}_i$ and $\mathbf{u}_j$ in the matrix $\mathbf{U}$.

5.4. Incorporating Historical Data

As we mentioned previously, the real-time probe data and traffic related tweets are both sparse. Relying on real-time data only is far from enough to fully estimate the traffic conditions of an arterial network. To address this issue, previous works [Chen et al. 2014; Wang et al. 2015] utilized a large volume of historical data as important reference information to facilitate the real-time estimation task.

In this article, we explore the following prior knowledge from historical data and incorporate them into our model: the historical congestion probability and the historical event occurrence probability for each road segment in each hour of a day. To model the prior knowledge, we construct the historical congestion probability matrix $\mathbf{Y}^h$ and the historical event occurrence probability tensor $\mathcal{A}^h$. Each entry y_{ij}^h denotes the empirical probability of the road segment r_i being in congestion state in t_j based on statistical analysis on historical tweets. Each entry a_{ijk}^h of $\mathcal{A}^h$ denotes the possibility that a traffic or social event e_k occurs on or near road segment r_i in hour t_j of a day. To integrate the prior knowledge mined from historical data to further alleviate the sparsity issue of real-time data, we incorporate the historical congestion probability matrix $\mathbf{Y}^h$ and the historical road event probability tensor $\mathcal{A}^h$ into our model.

Note that different from our previous model [Wang et al. 2016a], we take the weather information into consideration to guide us in selecting more relevant historical data. Given a day d_i on which we aim to estimate traffic conditions in each hour, we first calculate the weather similarity $Sim(f_{wea}^i, f_{wea}^j)$ between d_i and each historical day d_j by the method introduced in Section 3.2.3. Then we select the top-k similar days in the past year. We only use the traffic data in top-k similar days to construct the matrix $\mathbf{Y}^h$. A small k may lead to data sparsity, while a large k may result in including some days with less similar weather conditions. For different cities, k can be different due to different climatic features. In this article, we set k as 30 by fully considering the weather features including raining day and snowing day of Chicago. Based on this idea, the historical congestion probability $\mathbf{Y}^h$ can be constructed by

$$\mathbf{Y}^h = \frac{1}{k}\sum_{i=1}^{k} Sim\left(f_{wea}^{d_i}, f_{wea}^{d_j}\right)\mathbf{Y}_{d_i}. \tag{3}$$

One can see that each entry of the matrix $\mathbf{Y}^h$ is the weighted average entry on all the top-k historical days. The day with a more similar weather condition is assigned with a larger weight. By incorporating the historical prior knowledge, we finally have the following objective function:

$$\begin{aligned}&\mathcal{L}(\mathbf{U},\mathbf{V},\mathbf{W},\mathcal{C},\mathbf{F})\\&=\frac{1}{2}(||\mathbf{P}\odot(\mathbf{Y}-\mathbf{U}\mathbf{V}^{\mathrm{T}})||^2+||\mathbf{Q}\odot(\mathbf{H}-\mathbf{U}\mathbf{V}^{\mathrm{T}})||^2)+\frac{\lambda_1}{2}||\mathbf{Y}^h-\mathbf{U}\mathbf{V}^{\mathrm{T}}||^2\\&\quad+\frac{\lambda_2}{2}||\mathcal{R}\odot(\mathcal{A}-\mathcal{C}\times_1\mathbf{U}\times_2\mathbf{V}\times_3\mathbf{W})||^2+\frac{\lambda_3}{2}||\mathbf{X}-\mathbf{U}\mathbf{F}||^2+\frac{\lambda_4}{2}\mathrm{tr}(\mathbf{U}^{\mathrm{T}}\mathbf{L}_z\mathbf{U})\\&\quad+\frac{\lambda_5}{2}||\mathcal{A}^h-\mathcal{C}\times_1\mathbf{U}\times_2\mathbf{V}\times_3\mathbf{W}||^2+\frac{\lambda_6}{2}(||\mathbf{U}||^2+||\mathbf{V}||^2+||\mathbf{W}||^2+||\mathcal{C}||^2+||\mathbf{F}||^2),\end{aligned}\tag{4}$$

where $||\mathbf{Y}^h-\mathbf{U}\mathbf{V}^{\mathrm{T}}||^2$ is to control the error of factorizing the historical congestion probability matrix $\mathbf{Y}^h$, and $||\mathcal{A}^h-\mathcal{C}\times_1\mathbf{U}\times_2\mathbf{V}\times_3\mathbf{W}||^2$ is to control the error of factorizing the historical event tensor $\mathcal{A}^h$. The insight is that the congestion states of the road segments should be similar to their historical congestion states. Therefore, we assume that the congestion matrix $\mathbf{Y}$ should be similar to $\mathbf{Y}^h$, and the two matrices should share the same low rank factor matrices $\mathbf{U}$ and $\mathbf{V}$. Likewise, we also assume that the event tensor $\mathcal{A}$ and the historical event tensor $\mathcal{A}^h$ share the same factor matrices $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$, and the core tensor $\mathcal{C}$.

The objective function is not jointly convex to all the variables $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$, $\mathcal{C}$, and $\mathbf{F}$. Thus, it is very hard to get closed-form solutions to minimize the objective function. We use an elementwise optimization algorithm to iteratively update each entry in the matrices and tensor independently by gradient descent [Wang et al. 2015; Shang et al. 2014; Zheng et al. 2014b; Karatzoglou et al. 2010]. We omit the algorithm detail here, and one can refer to the works Shang et al. [2014] and Wang et al. [2015] for more details for solving this problem. Here we only list the gradient for each variable as follows:

$$\begin{aligned}&\nabla_{\mathbf{u}_{i:}}\mathcal{L}=[\mathbf{p}_{i:}\odot(\mathbf{u}_{i:}\mathbf{V}^{\mathrm{T}}-\mathbf{y}_{i:})]diag(\mathbf{p}_{i:})\mathbf{V}+[\mathbf{q}_{i:}\odot(\mathbf{u}_{i:}\mathbf{V}^{\mathrm{T}}-\mathbf{h}_{i:})]diag(\mathbf{q}_{i:})\mathbf{V}+\lambda_1(\mathbf{u}_{i:}\mathbf{V}^{\mathrm{T}}-\mathbf{y}^h_{i:})\mathbf{V}\\&+\lambda_2[r_{ijk}(\mathcal{C}\times_1\mathbf{u}_{i:}\times_2\mathbf{v}_{j:}\times_3\mathbf{w}_{k:}-a_{ijk})\mathcal{C}\times_2\mathbf{v}_{j:}\times_3\mathbf{w}_{k:}+\lambda_3(\mathbf{u}_{i:}\mathbf{F}-\mathbf{x}_{i:})\mathbf{F}^{\mathrm{T}}+\lambda_4(\mathbf{L}_z\mathbf{U})_{i:}\\&+\lambda_5(\mathcal{C}\times_1\mathbf{u}_{i:}\times_2\mathbf{v}_{j:}\times_4\mathbf{w}_{k:}-a^h_{ijk})\mathcal{C}\times_2\mathbf{v}_{j:}\times_3\mathbf{w}_{k:}+\lambda_6\mathbf{u}_{i:},\end{aligned}\tag{5}$$

$$\begin{aligned}\nabla_{\mathbf{v}_{j:}}\mathcal{L}&=[\mathbf{p}^{\mathrm{T}}_{:j}\odot(\mathbf{v}_{j:}\mathbf{U}^{\mathrm{T}}-\mathbf{y}_{:j}{}^{\mathrm{T}})]diag(\mathbf{p}_{:j})\mathbf{U}+[\mathbf{q}^{\mathrm{T}}_{:j}\odot(\mathbf{v}_{j:}\mathbf{U}^{\mathrm{T}}-\mathbf{h}_{:j}{}^{\mathrm{T}}]diag(\mathbf{q}_{:j})\mathbf{U}\\&+\lambda_1(\mathbf{v}_{j:}\mathbf{U}^{\mathrm{T}}-\mathbf{y}^{h}{}_{:j}{}^{\mathrm{T}})\mathbf{U}+\lambda_2[r_{ijk}(\mathcal{C}\times_1\mathbf{u}_{i:}\times_2\mathbf{v}_{j:}\times_3\mathbf{w}_{k:}-a_{ijk})\mathcal{C}\times_1\mathbf{u}_{i:}\times_3\mathbf{w}_{k:}]\\&+\lambda_5(\mathcal{C}\times_1\mathbf{u}_{i:}\times_2\mathbf{v}_{j:}\times_3\mathbf{w}_{k:}-a^h_{ijk})\mathcal{C}\times_1\mathbf{u}_{i:}\times_3\mathbf{w}_{k:}+\lambda_6\mathbf{v}_{j:},\end{aligned}\tag{6}$$

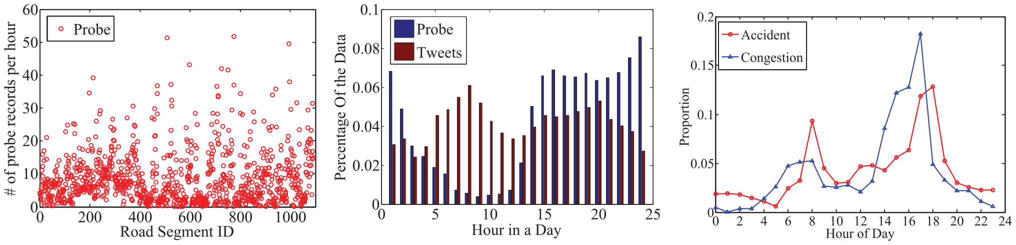
$$\begin{aligned}&\nabla_{\mathbf{w}_{k:}}\mathcal{L}=\lambda_2[r_{ijk}(\mathcal{C}\times_1\mathbf{u}_{i:}\times_2\mathbf{v}_{j:}\times_3\mathbf{w}_{k:}-a_{ijk})\mathcal{C}\times_1\mathbf{u}_{i:}\times_2\mathbf{v}_{j:}]+\lambda_3(\mathcal{C}\times_1\mathbf{u}_{i:}\times_2\mathbf{v}_{j:}\times_3\mathbf{w}_{k:}\\&-a^h_{ijk})\mathcal{C}\times_1\mathbf{u}_{i:}\times_2\mathbf{v}_{j:}+\lambda_6\mathrm{w}_{k:},\end{aligned}\tag{7}$$

$$\begin{aligned}&\nabla_{\mathcal{C}}\mathcal{L}=\lambda_2[r_{ijk}(\mathcal{C}\times_1\mathbf{u}_{i:}\times_2\mathbf{v}_{j:}\times_3\mathbf{w}_{k:}-a_{ijk})\mathbf{u}_{i:}\circ\mathbf{v}_{j:}\circ\mathbf{w}_{k:}]+\lambda_3(\mathcal{C}\times_1\mathbf{u}_{i:}\times_2\mathbf{v}_{j:}\times_3\mathbf{w}_{k:}\\&-a^h_{ijk})\mathbf{u}_{i:}\circ\mathbf{v}_{j:}\circ\mathbf{w}_{k:}+\lambda_6\mathcal{C},\end{aligned}\tag{8}$$

$$\nabla_{\mathrm{F}}\mathcal{L}=\lambda_4\mathbf{u}_{i:}{}^{\mathrm{T}}(\mathbf{u}_{i:}F-\mathbf{x}_{i:})+\lambda_6\mathrm{F}.\tag{9}$$

6. EXPERIMENT

In this section, we will evaluate the performance of the TCE_R model on estimating traffic congestions of downtown Chicago. We first will describe the experiment setup, including the dataset, ground truth, and evaluation metrics. Then we will use the proposed co-congestion pattern mining algorithm to perform anomaly detection. We will



(a) # of probes

(b) Hourly distributions

(c) Traffic event correlation.

Fig. 7. Data Statistics: (a) Average # of probe readings for each road link in each hour. (b) Hourly distributions of probe readings and tweets on each road segment. (c) Hourly occurrence correlations between traffic accidents and congestions reported by tweets.

evaluate the effectiveness of the proposed algorithm through a case study. Next we will perform quantitative evaluations. Specifically, we will study the effect of parameters on the model performance and test the model performance in rush hours of a day. Finally, we will evaluate the scalability and robustness of TCE_R.

6.1. Dataset Analysis and Experiment Setup

Datasets and Analysis. The Twitter data are described in Section 3. From each tweet, we extract the location, time, and traffic event information. We categorize these tweets into three types: congestion, accident, and others through keywords matching. We also have more than 2.4 million GPS probe readings on 1,257 arterial road links of downtown Chicago in December 2014. The total length of the road links is nearly 700 miles.

Some statistics of the two datasets are given in Figure 7. Figure 7(a) shows the hourly average number of probe readings for each road segment. One can see that first the probe data are sparse on the whole, and on average only around three probe readings are available for each road segment in our dataset. Second, the probe data are unevenly distributed on the arterial network. Probes frequently appear on a very small number of road segments, while for most road segments there are only several probe readings or even no readings. Figure 7(b) shows the distributions of the two types of date in each hour of the day. One can see that most probe data are distributed in the time interval from 14:00pm to 0:00am. Most traffic related tweets are posted in two time intervals from 5:00am to 10:00am and from 15:00pm to 22:00pm. The hourly distributions of the two datasets are not perfectly consistent, which implies tweets could be a good complementary data to the probe GPS data. Figure 7(c) shows the occurrence correlation between two major traffic events reported by tweets: traffic accident and congestion. One can see that the two occurrence curves show very similar increasing and decreasing trends, indicating a strong occurrence correlation.

Ground Truth. Obtaining the ground truth itself is a challenging problem. The manually annotated ground truth is very expensive, and thus is not feasible for a large transportation network. Previous studies show that the bus probe data in urban areas can provide a good approximation of the real traffic conditions [Wang et al. 2011; Raffaele et al. 2015]. Thus, we use the traffic conditions reported by CTA as the ground truth. This dataset includes the traffic speed data of Chicago's arterial streets in real time by continuously monitoring and analyzing GPS traces received from more than 2,000 CTA public passenger buses. The GPS probes of the CTA buses report their current locations and speed information every 10 minutes. The entire bus routes contain 1,257 road segments covering nearly 700 miles of arterial roads. We use the publicly available historical traffic data collected from 11/25/2014 to 12/30/2014,

Table IV. Top-10 Co-Congestion Patterns in Chicago

Co-congestion Patterns
(Michigan&Roosevelt, Michigan&Congress), (State&Roosevelt, State&Congress), (Wacker&Madison, Wacker&Van Buren)
(Lake Shore Dr&Division, Lake Shore Dr&Oak St), (Lake Shore Dr&Belmont, Lake Shore Dr&Diversey)
(Michigan&Roosevelt, Michigan&Congress), (Roosevelt&Clark, Roosevelt&Lake Shore Dr)
(Halsted&59th, Halsted&63rd), (Halsted&79th, Halsted&83rd), (Garfield&Halsted, Garfield&Racine), (Garfield&Ashland, Garfield&Damen)
(Ashland&Roosevelt, Ashland&Harrison), (Roosevelt&Ashland, Roosevelt&Clark), (Congress&Clark, Congress&I-294 Expy)
(Cermak&Racine, Cermak&Ashland), (Cermak&Damen, Cermak&Western), (Ashland&Roosevelt, Ashland&Cermark)
(Western&Fullerton, Western&Kennedy Expy), (Western&Devision, Western&North), (Western&Roosevelt, Western&Cermak)
(Milwaukee&Fullerton, Milwaukee&Diversey), (Milwaukee&Addison, Milwaukee&Irving Park)
(Pulaski&31st, Pulaski&I-55 Expy), (Pulaski&43rd Pulaski&I-55 Expy)
(31st&Dr Martin L King Jr, 31st&State), (Dr Martin L King Jr&26th, Dr Martin L King Jr&31th), (Dr Martin L King Jr&35th, Dr Martin L King Jr&31th)

which contains more than 5 million records.[2] Each record contains time, bus ID, road segment ID, the number of buses on the road segment, and the average speed. For each road segment in each hour, we use the real-time average speed as the ground truth speed if there are more than five probe readings; otherwise, we use a weighted average between the historical and real-time speeds. We use the method described in Section 5.1 to assign congestion values denoting the traffic conditions based on the bus speed. Roughly, we consider a road segment is in congestion if the speed is lower than 15 mph. Note that the GPS data of CTA buses used in ground truth is different from the probe data used in estimation. The probes used in our model are installed on general vehicles rather than buses.

Evaluation Metrics. We use the evaluation metric *precision@k* to conduct a coarse-grained estimation with only two traffic states: congestion and flow. We consider the road segment is congested if the average vehicle speed is lower than 15 mph. We obtain the congested road segments as follows. We first complete the road congestion matrix $\mathbf{H}$ by multiplying the low rank matrices $\mathbf{U}$ and $\mathbf{V}^{\mathrm{T}}$, and then rank the values of all the entries in $\mathbf{H}$. As in a particular time slot usually only a small portion of road segments are in congestion and most others are not, we consider the road segments with top-k entry values are in congestion.

To conduct a fine-grained evaluation, we also evaluate the TCE_R model in estimating traffic conditions with five traffic states in rush hours. Hence, we also use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) as evaluation metrics by normalizing the entry values in the range from 0.2 to 1 as shown in Table I.

$$MAE = \frac{\Sigma_i |y_i - \hat{y}_i|}{N} \text{ and } RMSE = \sqrt{\frac{\Sigma_i (y_i - \hat{y}_i)^2}{N}},$$

where y_i is the ground truth traffic state value and $\hat{y}_i$ is the estimated value.

6.2. Co-Congestion Patterns Mining

Table IV shows the top-10 co-congestion patterns discovered by the proposed search tree based frequent co-congestion patterns in the arterial network of downtown

[2] https://data.cityofchicago.org/Transportation/Chicago-Traffic-Tracker-Historical-Congestion-Esti/77hq-huss.

Chicago. The spatiotemporal constraint is set to 2-hops and 1 hour, and the $(minsup, minconf)$ are set to $(0.05, 0.3)$. One can see that the length of patterns are mostly less than 4. The first discovered pattern contains three road segments: (*Michigan&Roosevelt, Michigan&Congress*), (*State&Roosevelt, State&Congress*), and (*Wacker&Madison, Wacker&Van Buren*). This is not surprising as the three road segments all cross downtown Chicago. There is more traffic in downtown Chicago, thus the road segments in the downtown area are more likely to be in congestion. In addition, the three road segments are parallel to each other. They are more likely to be congested simultaneously as they are close to each other and head in the same direction. For the road segments of other patterns, they are mostly either connected to each other or belong to the same streets.

6.3. Anomaly Detection with Time-Sensitive Patterns Mining

The co-congestion patterns can evolve over time with the change of road conditions. For example, some new roads are built and added to the road network or some roads are closed for construction. Adding new roads may lead to the disappearance of some existing patterns; while closed road segments may make the road segments nearby more likely to be congested and some new patterns may appear. Identifying the evolving patterns may help us better plan the road construction and detect anomaly [Pan et al. 2013].

We identify the following three evolution patterns of the road segment co-occurrence in traffic congestion.

(1) *Formation*. Given two successive time intervals t_1 and t_2, a pattern $\{r_i, r_j\}; (\eta, k)$ appears in the time interval t_2 but not in the time interval t_1. That is to say, pattern $\{r_i, t_j\}; (\eta, k)$ is a new pattern for the time interval t_2.
(2) *Dissipation*. Given two successive time intervals t_1 and t_2, a pattern $\{r_i, r_j\}; (\eta, k)$ appears in the time interval t_1 but not in the time interval t_2. That is to say, pattern $\{r_i, r_j\}; (\eta, k)$ disappears from the time interval t_1 to the time interval t_2.
(3) *Remarkable change*. Given two successive time intervals t_1 and t_2, a pattern $\{r_i, r_j\}; (\eta, k)$ exists in both time intervals. However, the difference of the probabilities of the pattern in the two time intervals exceeds a predefined threshold μ, that is, $|Pr^{t_1}(\{r_i, r_j\}; k, \eta) - Pr^{t_2}(\{r_i, r_j\}; k, \eta)| > \mu$.

In this article, we use month as the time interval granularity. For each month m_i, we use the proposed search tree based co-congestion pattern mining algorithm to discover all the patterns P_i. Thus, we can have such a time series of patterns $(P_1, P_2, \ldots, P_n)$. Then we identify the preceding three evolution patterns. Table V gives some examples of discovered anomaly patterns of the three types.

The co-occurrence pattern (*Cermak&Halsted, Cermak&State*), (*18th&Wentworth, 18th&State*) appeared in December of 2014. This is a formation pattern as it did not appear in previous months. This occurrence of pattern is caused by the street closure of the 18th street bridge over the Chicago river for construction from 12/06/2014 to 03/01/2015. Likewise, a new co-congestion pattern (*31st&Halsted, 31st&Canal*), (*31st&Halsted, 35th&Halsted*), (*Halsted&26th, Halsted&31th*) appeared in November and December of 2014 due to the road segment closure of 31st street for construction. From October of 2014 on, the co-congestion pattern (*Division&Halsted, Division&Larrabee*), (*Division&Damen, Division&Ashland*) disappeared. This is because a new interim bridge at Division Street over the Chicago River opened to traffic in September of 2014, which greatly eased the traffic jams nearby. For the *Remarkable change* pattern type, we give two examples. The first one is the pattern (*Roosevelt&Halsted, Roosevelt&State*), (*Michigan&Roosevelt, Michigan&Congress*). The probability of this pattern increased significantly in the month February of

Table V. Three Types of Anomalies Detected by Time-Sensitive Pattern Mining

Anomalies	Co-congestion patterns	Causes
Formation	Time of formation: Dec. 2014 (Cermak&Halsted, Cermak&State), (18th&Wentworth, 18th&State)	Closure of the 18th Street bridge over the Chicago river, between Canal Street and Wentworth Avenue from 12/06/2014 to 03/01/2015
	Time of formation: Nov., Dec. 2014 (31st&Halsted, 31st&Canal), (31st&Halsted, 35th&Halsted), (Halsted&26th, Halsted&31th)	Closure of NB Halsted at 31st Street from 11/13/2014 to 01/31/2015
Dissipation	Time of dissipation: Oct. 2014 (Division&Halsted, Division&Larrabee), (Division&Damen, Division&Ashland)	A new interim bridge at Division Street over the Chicago River opened to traffic in Sep. 2014
Remarkable change	Time of remarkable change: Feb. 2015 (Roosevelt&Halsted, Roosevelt&State), (Michigan&Roosevelt, Michigan&Congress)	Heavy snow storm in this month
	Time of remarkable change: Feb. 2014 (State&47th, State&Garfield), (Dr Martin L King Jr&47th, Dr Martin L King Jr&51st)	The Red Line subway construction of CTA from Cermak-Chinatown through 95th/Dan Ryan

2015 due to a heavy snow storm of this month. The second example is the pattern (*State&47th, State&Garfield*), (*Dr Martin L King Jr&47th, Dr Martin L King Jr&51st*). In February of 2014, the Red Line CTA subway construction from Cermak-Chinatown through 95th/Dan Ryan was finished and began operation. Therefore, the nearby traffic congestion was effectively reduced.

6.4. Parameter Study

There are six parameters in our model, and different parameter settings could largely affect the algorithm performance [Bin et al. 2015]. To study the effect of the parameters and examine the importance of the multisourced data, in this subsection we conduct a parameter sensitivity study by tuning the parameter values from 0.01 to 1,000. We first find a relatively reasonable parameter setting by grid search for all the parameters. Then, to examine the effect of each parameter on the model performance, we fix the values of all the other five parameters and study how the *precision@k* changes with the value of the remaining one parameter.

The left figure in Figure 8 shows the *precision@k* curves of the parameters λ_1, λ_2, and λ_4, and the right figure shows that of the parameters λ_3, λ_5, and λ_6. One can see that the performance curves change greatly with the change of the parameters λ_1, λ_2, and λ_4. It implies that the three parts of information, namely, historical traffic congestion information, traffic/social events, and co-congestion patterns are very useful to the model. Ignoring them (very small parameter values) could significantly decrease the performance of the model. For example, the *precision@k* with $\lambda_1 = 10$ is about 0.8, while it decreases to only 0.68 if we ignore the historical traffic congestion information by setting $\lambda_1 = 0.01$. Compared to λ_1, λ_2, and λ_4, the effect of the parameters λ_3, λ_5, and λ_6 on the model performance is less significant. One can see that *precision@k* does not change much with the change of the parameters λ_5 and λ_6, which implies that road physical features and historical traffic/social event information does not help much. λ_6 can also affect the performance, which means that the regularization term does help avoid overfitting. One can also see that too large a value of the parameters will hurt

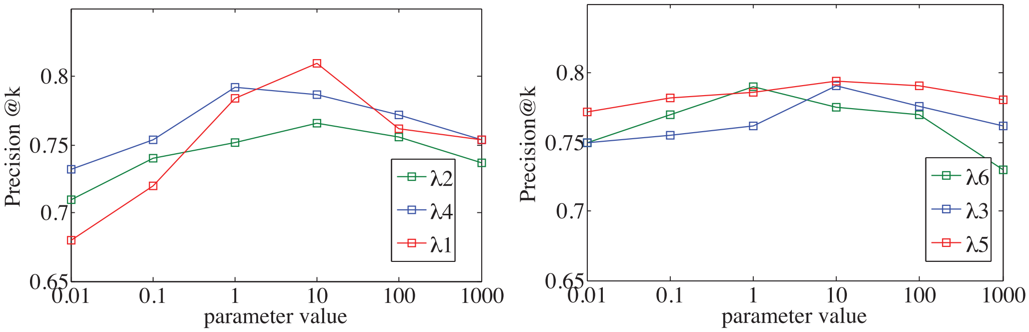


Fig. 8. Performance of the model with different parameter values.

the performance, as a parameter with a very large value will make the corresponding part of information dominate the importance of the other information.

6.5. Estimation Accuracy with Two Traffic States

We first use the evaluation metric *precision@k* to conduct a coarse-grained estimation with only two traffic states: congestion and flow. We consider the road segment is congested if the average vehicle speed is lower than 15 mph. We complete the following methods as baselines.

—**Collaborative filtering (CF).** CF is widely used in recommendation [Sarwar et al. 2001]. We can consider the congestion estimation task as a CF problem by factorizing the road congestion matrix only. In CF, only the congestion matrix **H** is used and factorized.

—**Coupled Hidden Markov Model (CHMM).** CHMM is a classical sparse GPS probe-based traffic congestion estimation model [Herring et al. 2010]. The main idea of this model is to utilize a Markov process to model the evolving traffic states on the road segments and the traffic correlations among the neighbor road segments.

—**TSE.** TSE is a Traffic Speed Estimation model based on a context-aware matrix factorization approach [Shang et al. 2014]. We apply the TSE model to cofactorize the following matrices: the road feature matrix **X**, the probe traffic congestion matrix **H**, the probe historical congestion matrix $\mathbf{H}^h$, and the congestion correlation matrix **Z**. TSE only utilizes the GPS probe data, but ignores the social media data.

—**CTCE.** CTCE model uses Twitter as the major data source to estimate urban traffic congestions [Wang et al. 2015]. We conduct a comparison with this model to investigate whether incorporating probe data can further improve the performance.

—**CEMD.** CEMD is our previous model that combines both probe GPS data and social media data to estimate urban traffic congestions [Wang et al. 2016a]. However, the co-congestion patterns are not fully studied and explored in this model. In addition, the CEMD model ignores the historical probe data information and the effect of different weather conditions on traffic congestions.

Table VI shows the average *precision@k* of the six methods over various *k*. As the traffic conditions on weekdays and weekends are different, we present the results by weekday and weekend separately. Different from our previous experiment setting which conducts parameter searching on the entire dataset [Wang et al. 2016a], we take an alternative approach on experiment parameters set up as follows. We use the data of the first 2 weeks as the training data to conduct matrix and tensor factorization for obtaining the best parameters. Then we use the learned parameters to estimate traffic conditions in the following 2 weeks. As the best parameters of the training data may

Table VI. Average *Precision@k* of Different Methods

Average *Precision@k* on Weekday								
	top-10	top-20	top-30	top-50	top-100	top-150	top-200	top-300
CF	0.500	0.437	0.414	0.400	0.412	0.387	0.375	0.352
TSE	0.821	0.786	0.774	0.720	0.676	0.656	0.643	0.617
CHMM	0.850	0.810	0.756	0.732	0.712	0.692	0.673	0.662
CTCE	0.870	0.840	0.823	0.820	0.784	0.745	0.723	0.712
CEMD	**0.880**	0.850	0.838	0.822	0.767	0.752	0.742	0.724
TCE_R	**0.880**	**0.860**	**0.842**	**0.824**	**0.810**	**0.785**	**0.750**	**0.734**
Average *Precision@k* on Weekend								
	top-10	top-20	top-30	top-50	top-100	top-150	top-200	top-300
CF	0.485	0.436	0.472	0.440	0.415	0.375	0.366	0.347
TSE	0.812	0.821	0.785	0.800	0.735	0.713	0.678	0.657
CHMM	0.840	0.825	0.813	0.772	0.756	0.722	0.682	0.675
CTCE	0.854	0.834	0.822	0.820	0.754	0.725	0.715	0.694
CEMD	0.872	0.847	**0.838**	0.822	0.767	0.745	0.723	0.715
TCE_R	**0.880**	**0.850**	0.835	**0.826**	**0.774**	**0.752**	**0.746**	**0.722**

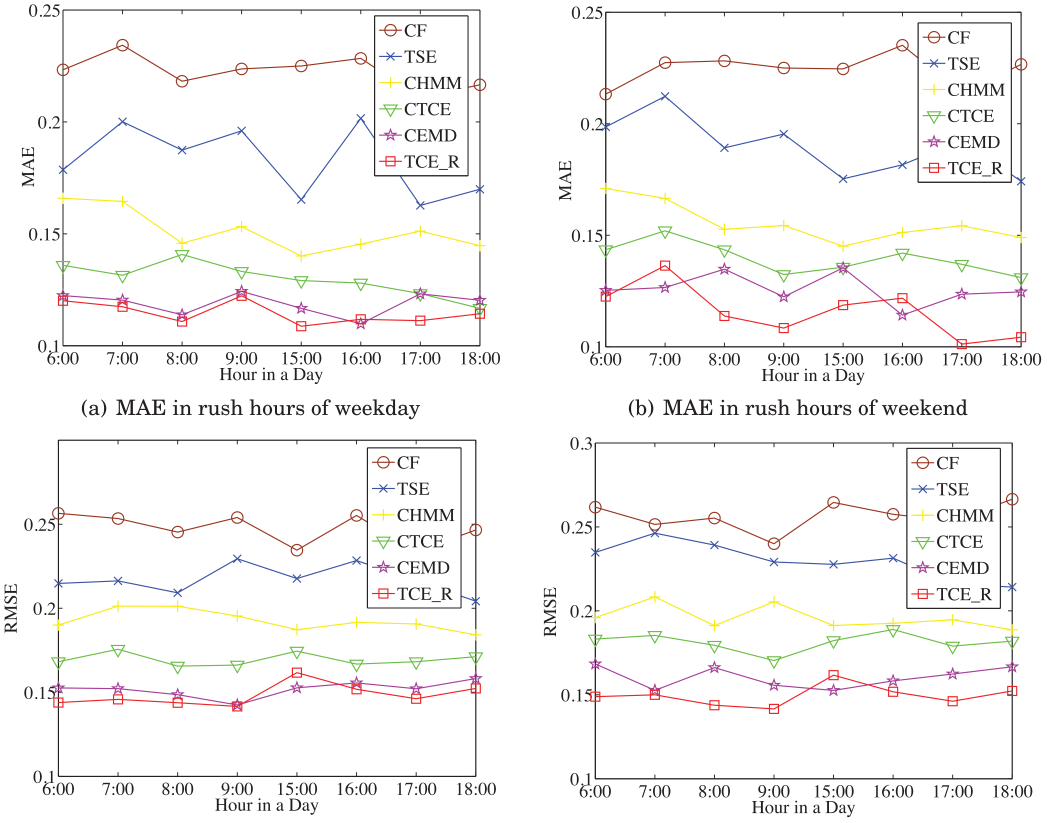
not perfectly fit the testing data, the performance of the CTCE and CEMD models is inferior to that we reported in our previous work. The best results are highlighted by bold. One can see that the proposed TCE_R model achieves the best performance in most cases with only one exception. One can see that the performance of the CF model is significantly inferior to the other methods. This is not surprising as the CF model only factorizes the congestion matrix and ignores other information. The performance is poor when the congestion matrix is very sparse. The CHMM model is better than the CF and TSE models, but inferior to the other three methods. This is because although CHMM is effective to capture the spatiotemporal correlations among the road segments to estimate traffic conditions with probe data, it cannot effectively handle multiple types of information. The CEMD model outperforms the CTCE model, which demonstrates the combination of probe data and social media data can further improve the estimation performance. However, CEMD is inferior to the TCE_R model in most cases, which means that incorporating the traffic correlation information and other information including weather data and historical probe speed data can further improve the performance.

One can also observe that the estimation performance on weekdays is slightly better than that on weekends. This implies that peoples' travel patterns are more regular on weekdays than on weekends, and thus the estimation on weekdays is easier than the estimation on weekends.

6.6. Quantitive Evaluation in Rush Hours

To conduct a fine-grained evaluation, we further evaluate the proposed TCE_R model in estimating traffic conditions with five traffic states in rush hours. Here we use MAE and RMSE as evaluation metrics to evaluate the performance of the methods in estimating traffic congestion state values.

Based on the fact that people may be more concerned with the traffic conditions in rush hours, we report the MAE and RMSE of different methods in the time intervals 6:00-10:00 am and 15:00-19:00 pm in Figure 9. Similar to our previous experiment, we also show the results on weekdays and weekends separately due to their different traffic conditions. One can see that in most cases the TCE_R achieves the lowest MAE and RMSE on both weekdays and weekends. Of all the methods, the performance of CF is the worst. The TSE model is only better than the CF model but significantly inferior to other methods. The CTCE model is consistently better than CHMM. It implies that using the GPS probe data only is less effective in traffic congestion estimation task



(a) MAE in rush hours of weekday

(b) MAE in rush hours of weekend

(c) RMSE in rush hours of weekday

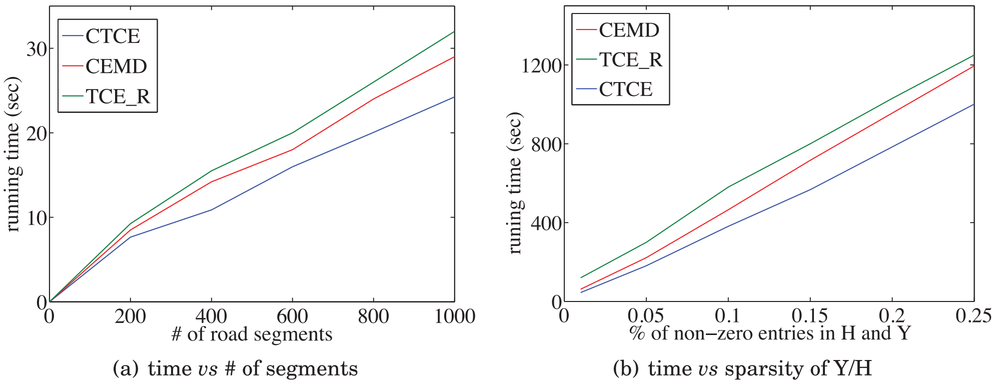
(d) RMSE in rush hours of weekend

Fig. 9. MAE and RMSE of different methods in rush hours of weekday and weekend.

compared to the method that combines multisourced data. The performance of TCE_R is better than CEMD in most cases. However, the improvement is not very significant on weekdays. This is probably because the traffic conditions on weekdays are easier to estimate than on weekends.

6.7. Scalability Analysis

To study the scalability of the proposed TCE_R, we compare the running time of TCE_R with CEMD and CTCE. The experiment results are shown in Figure 10. Figure 10(a) shows the increase trend of the running time for the three methods with the rising of road segment number. One can see that the running time of the three models linearly increases with the increase of road segment size. The running time of TCE_R is slightly longer than CTCE and CEMD. This is firstly because TCE_R combines the probe data and Twitter data. Secondly, TCE_R considers more side information, thus it needs to factorize more matrices. However, the figure shows that TCE_R is not time-consuming. For an arterial road network with 1,000 road segments, the running time of TCE_R is only around 30 seconds due to the very sparse matrices. To further study the effect of congestion matrices $\mathbf{Y}$ and $\mathbf{H}$ sparsity on the running time, we plot the running times of the three methods under various proportions of nonzero entries in $\mathbf{Y}$ and $\mathbf{H}$. One can see that a denser congestion matrix leads to a much longer running time of the three methods. The running time of TCE_R is always slightly longer than the other



(a) time *vs* # of segments (b) time *vs* sparsity of Y/H

Fig. 10. Scalability study on different methods.

Table VII. F1-score of Difference Methods with Sparse Probe Data

#% of probe data	40%	50%	60%	70%	80%	90%	100%
CF	0.216	0.227	0.242	0.257	0.275	0.284	0.299
TSE	0.326	0.337	0.342	0.375	0.452	0.465	0.476
CEMD	0.415	0.426	0.432	0.455	0.468	0.486	0.492
TCE_R	**0.422**	**0.435**	**0.446**	**0.461**	**0.476**	**0.498**	**0.502**

two methods. In the real case the congestion matrices **Y** and **H** are both very sparse due to the very spare data. Thus, the proposed TCE_R model can easily scale to large arterial road networks with thousands of road segments.

6.8. Robustness Study with Sparse GPS Probe Data

To further study the robustness of the proposed TCE_R model when handling very spare GPS probe data, we also conduct experiments to test the performance of TCE_R with different percentages of GPS probe readings. Table VII shows the F1-score of TCE_R and the other three models when 40%, 50%, 60%, 70%, 80%, 90%, and 100% real-time GPS probe readings are available for training the model, respectively. One can see that with the increase of the available GPS probe readings, the F1-score of all the methods increases. More data means better performance for all the methods. Among all the methods, TCE_R achieves the best performance while CF is the worst, demonstrating the power of the multisourced data and the effectiveness of TCE_R in fusing them. One can also see that the performance of TCE_R does not decrease much when sparer GPS probe readings are available. This is mainly because TCE_R takes both historical GPS probe data and historical traffic event tweets into consideration as prior knowledge, and thus it can still give a reasonable estimation even though the real-time GPS probe data are sparse. It demonstrates the robustness of TCE_R in traffic congestion estimation with sparse GPS probe readings.

7. RELATED WORK

Traffic congestion estimation on both highways and arterial transportation networks has been extensively studied in the community of traffic engineering [Porikli and Li 2004; Pattara-Atikom et al. 2006; de Fabritiis et al. 2008]. Due to the lack of ubiquity and a uniform penetration rate of the probe data, traffic monitoring on an arterial transportation network in the urban area is more challenging compared to traffic monitoring on highways [Park and Lee 2004; van Zuylen et al. 2010]. Existing works on traffic monitoring and traffic congestion estimation on an arterial transportation network can be roughly categorized into traffic modeling on individual roads [Helbing 2001; Muñoz et al. 2003; Porikli and Li 2004; de Fabritiis et al. 2008] and on a road network [Pattara-Atikom et al. 2006; Yuan et al. 2010; Shang et al. 2014]. These works mostly rely on various road sensor data to conduct traffic monitoring. For example, loop detector data can be widely utilized to estimate the traffic density at unmonitored locations along a highway [Muñoz et al. 2003]. Another commonly used road sensor data is the video data generated by road cameras. Porikli and Li proposed a Gaussian Mixture Hidden Markov Models (GM-HMM) to detect traffic conditions with the road camera video data [Porikli and Li 2004]. Monitoring traffic conditions on a road network is much more challenging than that on an individual road due to the data sparsity issue. To address this issue, it is necessary to capture and model the spatiotemporal correlations of the traffic conditions among the road segments connected to each other [Pattara-Atikom et al. 2006; Yuan et al. 2010; de Fabritiis et al. 2008; Herring et al. 2010; Shang et al. 2014]. Monitoring traffic conditions of a road network usually uses different types of sensor data such as the Floating Car Data (FCD) [Yuan et al. 2010] or GPS probe data [Herring et al. 2010]. As a representative work, Herring et al. proposed a coupled Hidden Markov Model which can effectively capture the spatiotemporal correlations among the road segments to more effectively estimate traffic congestions of an arterial network with GPS probe data [Herring et al. 2010]. Fabritiis et al. studied the problem of using real-time FCD data based on traces of GPS positions to predict the traffic on a motorway network in Italian [de Fabritiis et al. 2008]. Zheng provided a comprehensive survey [Zheng 2015b] on algorithms and applications of mining trajectory data, especially the vehicle trajectory data. Many urban traffic analysis models with the taxi vehicle trajectory data are discussed and summarized in the survey. Yuan investigated how to use the trajectories of taxis collected by GPS to efficiently find driving directions for drivers [Yuan et al. 2010]. With the availability of other rich information like POIs and road features, some recent research tried to explore these features to help estimate traffic conditions. Shang et al. proposed a context-aware matrix factorization algorithm to estimate the traffic speed of the road network in Beijing by integrating POIs and road geographic features [Shang et al. 2014].

Currently, social media data is widely used to improve many online applications [Ma et al. 2015]. There are also increasing research interests to study how to utilize social media data to help understand traffic conditions [Endarnoto et al. 2011; D'Andrea et al. 2015; Liu et al. 2014; Schulz et al. 2013; Chen et al. 2014; Wang et al. 2015, 2016a, 2016b]. Most previous works focused on investigating either how to extract and visualize the traffic event information from tweets [Endarnoto et al. 2011; D'Andrea et al. 2015; Liu et al. 2014; Schulz et al. 2013] or how to locate the traffic events mentioned in the tweets [Daly et al. 2013; Sílvio S. Ribeiro et al. 2012]. Chen et al. proposed a Hinge-Loss Markov Random Fields to effectively monitor traffic conditions of an arterial network with social media data [Chen et al. 2014]. A major limitation of this work is that this model can only handle social media data but cannot effectively incorporate other types of traffic information. As the social media data can be very sparse, relying only on them usually cannot achieve desirable performance. Wang et al. further incorporated other information such as social events and road features with social media data to more effectively estimate citywide traffic congestions [Wang et al. 2015]. However, this work used social media as the major data source, and cannot effectively combine sensor data. As the first attempt, Wang et al. further studied how to combine social media data and GPS probe data to better understand urban traffic congestions [Wang et al. 2016a]. As an extension of this work, in this article we combine social media data and GPS probe data as well as other rich side information including road features, social events, traffic correlations, and explore a framework that can effectively combine the rich information for better computing urban traffic

Table VIII. Features of Related Works

	Sensor	Social media	Road network	Side information	Traffic correlation
Muñoz et al. [2003]	√				
Porikli and Li [2004]	√				
Herring et al. [2010]	√		√		√
van Zuylen et al. [2010]	√		√		
de Fabritiis et al. [2008]	√		√		
Daly et al. [2013]	√	√			
Shang et al. [2014]	√		√	√	
Chen et al. [2014]		√	√		
Wang et al. [2015]		√	√	√	√
Wang et al. [2016b]	√	√	√		√
Wang et al. [2016a]	√	√	√	√	√
The proposed TCE_R	√	√	√	√	√

conditions. Basically, this type of methods explore richer information and aim to fuse the data coming from multiple sources to better perform an estimation task. For more details of the methodologies for multisource data fusing, one can refer to the survey [Zheng 2015a]. To clearly show the features of different traffic monitoring or congestion estimation models, we list related works and corresponding features in Table VIII. We summarize five different features of these models: whether using sensor data, whether using social media data, whether monitoring or estimating traffic conditions on a road network, whether using side information (physical features, POIs, weather), and whether exploring traffic correlations. Our models contain all five features while other models only contain parts of these features. Thus, our model is more powerful to incorporate rich information into a unified framework to more accurately estimate traffic conditions of an arterial road network.

8. CONCLUSION

This article proposes a novel framework to effectively integrate GPS probe data and social media data for more accurately computing urban traffic congestions. Due to the sparsity characteristic, GPS probe data are usually not sufficient to precisely estimate traffic conditions of a large arterial network. To address this issue, we extensively collect traffic related tweets that report various traffic event information from Twitter. We also exploit other side information that might affect traffic conditions including road features, social events, as well as weather information. With the rich traffic related data, we first study the novel co-congestion frequent pattern mining problem by proposing a search tree based method. The discovered traffic co-congestion patterns are then used to detect anomaly in the arterial network and help us better estimate traffic conditions of a large arterial network. To effectively integrate the rich information, we finally propose a coupled matrix and tensor factorization model. This model can complete the congestion matrix with the help of other matrices and tensors formed by the multisourced data. We finally conduct extensive evaluations on the arterial network of downtown Chicago, and the results verify the effectiveness, and robustness of the proposed method.

REFERENCES

Gu Bin, Victor S. Sheng, and Shuo Li. 2015. Bi-parameter space partition for cost-sensitive SVM. In *Proceedings of the 24th International Conference on Artificial Intelligence*. 3532–3539.

Po-Ta Chen, Feng Chen, and Zhen Qian. 2014. Road traffic congestion monitoring in social media with hinge-loss Markov random fields. In *Proceedings of the IEEE International Conference on Data Mining*. 80–89.

Elizabeth M. Daly, Freddy Lecue, and Veli Bicer. 2013. Westland row why so slow?: Fusing social media and linked data sources for understanding real-time traffic conditions. In *Proceedings of the International Conference on Intelligent User Interfaces*. 203–212.

Eleonora D'Andrea, Pietro Ducange, Beatrice Lazzerini, and Francesco Marcelloni. 2015. Real-time detection of traffic from Twitter stream analysis. *IEEE Transactions on Intelligent Transportation Systems* 16, 4 (Aug. 2015), 2269–2283.

Corrado de Fabritiis, Roberto Ragona, and Gaetano Valenti. 2008. Traffic estimation and prediction based on real time floating car data. In *Proceedings of the International Conference on Information Technology and Computer Science*. 197–203.

Sri Krisna Endarnoto, Sonny Pradipta, Anto Satriyo Nugroho, and James Purnama. 2011. Traffic condition information extraction and visualization from social media Twitter for android mobile application. In *Proceedings of the International Conference on Electronics Engineering and Informatics*. 1–4.

Thomas F. Golob and Wilfred W. Recker. 2003. Relationships among urban freeway accidents, traffic flow, weather, and lighting conditions. *Journal of Transportation Engineering* 129, 4 (2003), 342–353.

Dirk Helbing. 2001. Traffic and related self-driven many-particle systems. *Reviews of Modern Physics* 73, 4 (Dec. 2001), 1067–1141.

Ryan Herring, Aude Hofleitner, Pieter Abbeel, and Alexandre Bayen. 2010. Estimating arterial traffic conditions using sparse probe data. In *Proceedings of the International IEEE Conference on Intelligent Transportation Systems*. 929–936.

Sean Higgins. 2013. Nokia's HERE adds real-time traffic info to Esri mapping platform: HERE processes 20 billion real-time GPS probe points a month. (2013). http://www.spar3d.com/news/related-new-technologies/nokias-here-adds-real-time-traffic-info-to-esri-mapping-platform/.

Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. 2010. Multiverse recommendation: N-dimensional tensor factorization for context aware collaborative filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems*. 79–86.

Jeffrey R. Kenworthy. 2006. The eco-city: Ten key transport and planning dimensions for sustainable city development. *Environment and Urbanization* 18, 1 (2006), 67–85.

Tamara G. Kolda and Brett W. Bader. 2009. Tensor decompositions and applications. *SIAM Review* 51, 3 (2009), 455–500.

Meiling Liu, Kaiqun Fu, Chang-Tien Lu, Guangsheng Chen, and Huiqiang Wang. 2014. A search and summary application for traffic events detection based on Twitter data. In *Proceedings of the ACM SIGSPATIAL International Conferences on Advances in Geographic Information Systems*. 549–552.

Tinghuai Ma, Jinjuan Zhou, Meili Tang, Yuan Tian, Abdullah Al-Dhelaan, Mznah Al-Rodhaan, and Sungyoung Lee. 2015. Social network and tag sources based augmenting collaborative recommender system. *IEICE Transactions on Information and Systems* E98-D, 4 (2015), 902–910.

Laura Muñoz, Xiaotian Sun, Roberto Horowitz, and Luis Alvarez. 2003. Traffic density estimation with the cell transmission model. In *Proceedings of the 2003 American Control Conference*. 3750–3755.

Celil Ozkurt and Fatih Camci. 2009. Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks. *Mathematical and Computational Application* 14, 3 (2009), 187–196.

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*. 344–353.

Taehyung Park and Sangkeon Lee. 2004. A Bayesian approach for estimating link travel time on urban arterial road network. In *Proceedings of the International Conference on Computational Science and Its Applications*. 1017–1025.

W. Pattara-Atikom, P. Pongpaibool, and S. Thajchayapong. 2006. Estimating road traffic congestion using vehicle velocity. In *Proceedings of the 6th International Conference on ITS Telecommunications*. 1001–1004.

Fatih Porikli and Xiaokun Li. 2004. Traffic congestion estimation using HMM models without vehicle tracking. In *Proceedings of the Intelligent Vehicles Symposium*. 188–193.

Carli Raffaele, Mariagrazia Dotoli, Nicola Epicoco, Biagio Angelico, and Antonio Vinciullo. 2015. Automated evaluation of urban traffic congestion using bus as a probe. In *Proceedings of the 2015 IEEE International Conference on Automation Science and Engineering*. 967–972.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the International World Wide Web Conference*. 285–295.

David Schrank, Bill Eisele, and Tim Lomax. 2012. *2012 URBAN MOBILITY REPORT Powered by INRIX Traffic Data*.

David Schrank, Bill Eisele, Tim Lomax, and Jim Bak. 2015. 2015 Mobility Scorecard. Technical Report jointly published by The Texas A&M Transportation Institute, The Texas A&M University System and INRIX, Inc.

Axel Schulz, Petar Ristoski, and Heiko Paulheim. 2013. *I See a Car Crash: Real-time Detection of Small Scale Incidents in Microblogs*. Springer, 22–33.

Jingbo Shang, Yu Zheng, Wenzhu Tong, Eric Chang, and Yong Yu. 2014. Inferring gas consumption and pollution emission of vehicles throughout a city. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, 1027–1036.

Sílvio S. Ribeiro, Jr., Clodoveu A. Davis, Jr., Diogo Rennó R. Oliveira, Wagner Meira, Jr., Tatiana S. Gonçalves, and Gisele L. Pappa. 2012. Traffic observatory: A system to detect and locate traffic events and conditions using Twitter. In *Proceedings of the ACM SIGSPATIAL International Workshop on Location-Based Social Networks (LBSN)*. 5–11.

Sha Tao, Vasileios Manolopoulos, Saul Rodriguez, and Ana Rusu. 2012. Real-time urban traffic state estimation with A-GPS mobile phones as probes. *Journal of Transportation Technologies* 2, 1 (2012), 22–31.

Theodore Tsekeris and Nikolas Geroliminis. 2013. City size, network structure and traffic congestion. *Journal of Urban Economics* 76 (2013), 1–14.

Henk J. van Zuylen, Fangfang Zheng, and Yusen Chen. 2010. Using probe vehicle data for traffic state estimation in signalized urban networks. In *Traffic Data Collection and its Standardization*. Vol. 144. Springer, New York.

Senzhang Wang, Lifang He, Leon Stenneth, Philip S. Yu, and Zhoujun Li. 2015. Citywide traffic congestion estimation with social media. In *Proceedings of the ACM SIGSPATIAL International Conferences on Advances in Geographic Information Systems*.

Senzhang Wang, Lifang He, Leon Stenneth, Philip S. Yu, Zhoujun Li, and Zhiqiu Huang. 2016a. Estimating urban traffic congestions with multi-sourced data. In *Proceedings of the 2016 IEEE International Conference on Mobile Data Management*. 82–91.

Senzhang Wang, Fengxiang Li, Leon Stenneth, and Philip S. Yu. 2016b. Enhancing traffic congestion estimation with social media by coupled hidden Markov model. In *Proceedings of the 2016 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. 247–264.

Yin Wang, Yanmin Zhu, and Zhaocheng He. 2011. *Challenges and Opportunities in Exploiting Large-Scale GPS Probe Data*. Technical Report HPL-2011-109. HP Laboratories.

Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. 2010. T-drive: Driving directions based on taxi trajectories. In *Proceedings of the ACM SIGSPATIAL International Conferences on Advances in Geographic Information Systems*. 99–108.

Yufei Yuan, Hans Van Lint, Femke Van Wageningen-Kessels, and Serge Hoogendoorn. 2014. Network-wide traffic state estimation using loop detector and floating car data. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations* 18, 1 (2014), 41–50.

Chao Zhang, Yu Zheng, Xiuli Ma, and Jiawei Han. 2015. Assembler: Efficient discovery of spatial co-evolving patterns in massive geo-sensory data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1415–1424.

Yu Zheng. 2015a. Methodologies for cross-domain data fusion: An overview. *IEEE Transactions on Big Data* 1, 1 (2015), 16–34.

Yu Zheng. 2015b. Trajectory data mining: An overview. *ACM Transactions on Intelligent Systems and Technology* 6, 3, Article 29 (May 2015).

Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014a. Urban computing: Concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology* 5, 3 (2014), Article 38.

Yu Zheng, Tong Liu, Yilun Wang, Yanmin Zhu, Yanchi Liu, and Eric Chang. 2014b. Diagnosing New York City's noises with ubiquitous data. In *Proceedings of the ACM Joint International Conference on Pervasive and Ubiquitous Computing*. 715–725.

Received June 2016; revised December 2016; accepted December 2016