Citywide Traffic Congestion Estimation with Social Media

Senzhang Wang*, Lifang He†, Leon Stenneth§, Philip S. Yu‡, Zhoujun Li*
* Beihang University, Beijing, 100191, China
† Shenzhen University, Shenzhen, 518060, China
§ Nokia's HERE Connected Driving, Chicago, IL ,USA
‡ University of Illinois at Chicago, Chicago, IL 60607, USA
{szwang, lizj}@buaa.edu.cn, lifanghescut@gmail.com
leon.stenneth@here.com, psyu@uic.edu

ABSTRACT

Conventional traffic congestion estimation approaches require the deployment of traffic sensors or large-scale probe vehicles. The high cost of deploying and maintaining these equipments largely limits their spatial-temporal coverage. This paper proposes an alternative solution with lower cost and wider spatial coverage by exploring traffic related information from Twitter. By regarding each Twitter user as a traffic monitoring sensor, various real-time traffic information can be collected freely from each corner of the city. However, there are two major challenges for this problem. Firstly, the congestion related information extracted directly from real-time tweets are very sparse due both to the low resolution of geographic location mentioned in the tweets and the inherent sparsity nature of Twitter data. Secondly, the traffic event information coming from Twitter can be multi-typed including congestion, accident, road construction, etc. It is non-trivial to model the potential impacts of diverse traffic events on traffic congestion. We propose to enrich the sparse real-time tweets from two directions: 1) mining the spatial and temporal correlations of the road segments in congestion from historical data, and 2) applying auxiliary information including social events and road features for help. We finally propose a coupled matrix and tensor factorization model to effectively integrate rich information for **C**itywide **T**raffic **C**ongestion **E**estimation (CTCE). Extensive evaluations on Twitter data and 500 million public passenger buses GPS data on nearly 700 mile roads of Chicago demonstrate the efficiency and effectiveness of the proposed approach.

Categories and Subject Descriptors

H.2 [**Database Management**]: Database Applications

SIGSPATIAL'15, November 03 - 06, 2015, Bellevue, WA, USA

ACM 978-1-4503-3967-4/15/11$15.00
DOI: http://dx.doi.org/10.1145/2820783.2820829

Keywords

social media, congestion estimation, tensor factorization

1. INTRODUCTION

As the number of vehicles steadily increases and the expansion of roadways is relatively slow, traffic congestion is becoming a central transportation issue in big cities [16, 18]. Traffic congestion not only leads to waste of time and money, but also increases air pollution due to extra gas emission. According to the study in [16], the yearly delayed time during congestion is about 5.5 billion hours in U.S. with more than 2.9 billion gallons fuel wasted, and the estimated cost is over 100 billion dollars. People living in urban areas, like Chicago and New York, are increasingly concerned about real-time traffic conditions, calling for data mining technologies that can instantly estimate citywide traffic congestions.

Traditional traffic congestion monitoring methods rely on various road sensor data collected from loop detectors [11], surveillance cameras [12], radars, etc. It is expensive to deploy and maintain these devices such that these methods are difficult to scale up to cover an entire city. 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 [24]. 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 [20]. The problem is that probe data are usually noisy. Existing methods matching GPS coordinates to a passable route may not be accurate due to noisy data [22].

The goal of this paper is to put forward an alternative traffic congestion estimation solution with lower cost and wider spatial coverage by utilizing one of the most popular social media sites Twitter as the data source. With the rising popularity of social media, Twitter has become an indispensable platform for online users to share real-time traffic information [7]. By regarding each Twitter user as a traffic sensor, traffic information can be freely obtained. Meanwhile, the data coverage is not limited to principle arterials as Twitter users including pedestrians, drivers, and passengers can spread over the entire city. In addition, traffic events like construction and accident can be directly obtained from tweets as they are summarized already, but may be difficult to infer from GPS data. Recently, some efforts have been devoted to utilizing social media data to help understand traffic conditions [4, 5, 10, 19]. These works mainly focus on studying either how to extract the traffic event in-

formation from tweets [5, 10, 17], or how to locate the traffic events mentioned in the tweets [19]. [3] has attempted to monitor traffic congestions with real-time traffic event tweets. However, auxiliary information including historical data, congestion correlation, and road features are not fully explored, thus it may not effectively estimate congestions of an entire city.

Although instant traffic information are increasingly available through social media, using Twitter as the primary data source to estimate traffic congestions of a city is very difficult due to the following two major challenges. 1) **Data sparsity.** Although Twitter data can cover a much larger area of the city, not all the traffic events can be captured from Twitter. Based on data analysis on our crawled tweets, only around 12% of congestions in Chicago can be captured from tweets in real-time. In addition, many traffic event tweets lack the exact location information. For example, a tweet may say "*Heavy traffic to O'Hare*". It is very hard to locate which road segments are in congestion currently. The low resolution of geographic location mentioned in the tweets makes the data sparser. For a citywide estimation task, relying on the sparse instant tweets only is far from enough. 2) **Heterogeneous data integration.** The traffic event information reported by Twitter can be multi-typed, including congestion, accident, road construction, disabled cars, road closure, etc. Besides traffic events, social events such as concerts and football matches may also remarkably affect traffic conditions nearby. Both traffic events and social events could be helpful for congestion estimation, especially for the non-recurrent traffic congestions. It is also challenging to quantitively model the latent impacts of the diverse traffic and social events on traffic congestion.

In this paper, we first widely collect and process traffic related tweets from both traffic authority accounts (explain later) and general user accounts. Then we utilize the data to investigate the spatial and temporal correlations among the road segments in congestion. A spatio-temporal frequent pattern mining method is given to effectively discover which road segments are likely to co-occure congestion. The congestion correlations could be used to facilitate instant traffic congestion estimation, but is largely ignored by previous studies [3, 14, 20]. We also extract social events within the city and use them as supplementary information to help our estimation task. Given a social event with time and location information, we propose using the Gaussian distribution to model the intensity of its impact on surrounding road segments based on their geographical distance. Motivated by the successful application of matrix completion technique to address sparse issue in many areas [9, 18], we next propose a coupled matrix and tensor factorization scheme to integrate the rich historical data, congestion correlation data as well as multi-typed event data. With the collaboratively factorized low rank matrices, we can effectively estimate the citywide traffic congestion by completing the missing values in the sparse road congestion matrix.

We summarize the contributions of this work as follows.

- We propose a traffic congestion estimation framework by using Twitter as the data source. Compared with traditional models, the proposed method can cover a much larger area of the city with lower cost.
- To alleviate data sparsity issue, we propose a spatio-temporal frequent pattern mining method to study the correlations among the road segments in congestion. We also extract social events and road features to facilitate congestion estimation.
- We propose a coupled matrix and tensor factorization scheme to collaboratively factorize the congestion matrix with the congestion correlation matrix, event tensor, and the road feature matrix. The congestion matrix of an entire city is then completed by multiplying the low rank latent factor matrices.
- We utilize the Chicago Transit Authority (CTA) public bus GPS data as the ground truth to evaluate our model. Besides the promising performance (0.126 MAE in rush hours, around 80% accuracy on top100 congested road segments) compared with the CTA probe data and various baselines, the results also demonstrate the scalability and robustness of the proposed approach.

The remainder of the paper is organized as follows. In Section 2, we give the framework of the model. Section 3 introduces data collection. Section 4 describes how to mine the road segment correlations in congestion. We elaborate the coupled matrix and tensor factorization schema in Section 5. Evaluations are given in Section 6 followed by related work in Section 7. We conclude this paper in Section 8.

2. PRELIMINARY AND FRAMEWORK

This section provides a brief introduction to the framework of the proposed method. We start with some definitions to help us state the problem.

DEFINITION 1. ***An arterial road*** R_i. *An arterial road* R_i *can be represented as a tuple* $R_i = (name_i, dir_i, \boldsymbol{L}_i)$, *where* $name_i$ *is the name of the arterial road,* dir_i *denotes the road direction, and* $\boldsymbol{L}_i$ *is the set of intersections* $\boldsymbol{L}_i = (l_{i,1}, ..., l_{i,n})$. *The intersection* $l_{i,j}$ *contains the exact location information and can be represented as* $l_{i,j} = (lat_{i,j}, lon_{i,j})$.

DEFINITION 2. ***A road segment*** r_i. *A segment* r_i *of the arterial road* R_i *is a continuous part of* R_i. *Formally, we define* $r_i = (ID_i, name_i, \boldsymbol{l}_i)$, *where* ID_i *is the unique ID of the road segment,* $name_i$ *is the name of the arterial road* r_i *belongs to, and* $\boldsymbol{l}_i$ *is a subset of* $\boldsymbol{L}_i$.

DEFINITION 3. ***A*** k***-hop road segment*** $r_i(k)$. *Given a road segment* r_i, *we call it a* k*-hop road segment if it contains* $k+1$ *intersections.*

If we consider each Twitter user as a sensor monitoring the traffic conditions nearby, their traffic event tweets can be regarded as traffic event signals which can be related to congestion, car accident, blocked road segments, etc.

DEFINITION 4. ***A traffic event tweet*** e_i. *We represent a traffic event tweet* e_i *as such a tuple* $e_i = (c_i, \boldsymbol{w}_i, t_i)$ *where* $c_i \in C$ *is the traffic event category,* $\boldsymbol{w}_i = (w_{i,1}, ..., w_{i,N_{e_i}})$ *represents the words mentioning the locations of the event, and* t_i *denotes the event time.*

Figure 1 presents the framework of the proposed method. It is comprised of three major parts: 1) real-time Twitter data acquisition and traffic events extraction, 2) spatial and

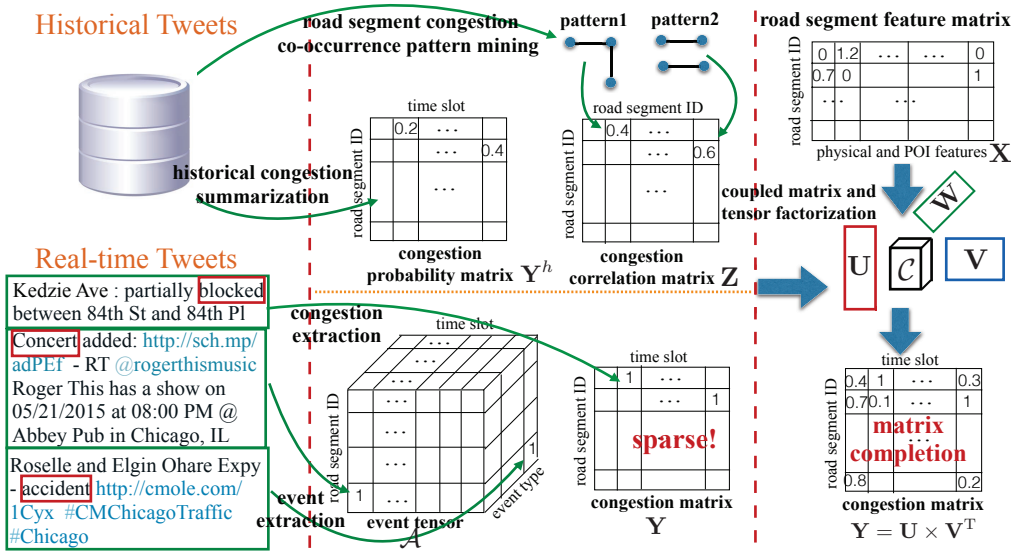


Figure 1: Framework of the proposed method

temporal correlation mining of traffic congestions from historical tweets, and 3) coupled matrix and tensor factorization by combining above information and road features for traffic congestion estimation. In the following sections, we will introduce the three parts in detail. Here we only give a brief description.

We instantly collect real-time traffic and social event tweets, and extract the event category, location, and time information from the tweets. Based on these information, we construct the real-time congestion matrix $\mathbf{Y}$ and 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 j_{th} hour 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 j_{th} hour of a day. Besides the traffic events like car accident, construction, we also extract social events such as concert and football match.

We also try to mine the spatial and temporal correlations among the road segments in congestions from a large volume of historical tweets. Specially, we conduct 1) congestion probability summarization of all the road segments, and 2) road segment congestion co-occurrence pattern mining. The former provides historical knowledge of which road segments are more likely to be in congestion in each time slot. The later shows us which groups of road segments are more likely to be in congestion simultaneously. Two matrixes are then constructed: historical congestion probability matrix $\mathbf{Y}^h$ and congestion correlation matrix $\mathbf{Z}$. h represents the historical data. The entry y^h_{ij} of $\mathbf{Y}^h$ denotes the probability of road segment r_i congested in the j_{th} hour of a day, and the entry z_{ij} of $\mathbf{Z}$ denotes the probability of road segments r_i and r_j co-occurring congestion.

The problem is that the congestion matrix $\mathbf{Y}$ is very sparse and most entries are empty. Predicting unfilled entries is known as the "matrix completion" problem and is increasingly used for sparse problems [18, 25]. Our task is to perform matrix completion by utilizing coupled matrix and tensor factorization which makes full use of rich information. Formally, the studied problem can be defined as follows.

Twitter based Citywide Traffic Congestion Estimation: *Given the traffic event tweets $\mathcal{E} = \{e_1, ..., e_m\}$, the road segments of a city $\mathcal{L} = \{r_1, ..., r_n\}$, the road features $\mathcal{F} = \{f_1, ..., f_n\}$, and the time slots $\mathcal{T} = \{t_1, ..., t_k\}$, the problem is how to estimate the traffic congestions on the road segments $\mathcal{L}$ in $\mathcal{T}$, namely how to complete the sparse congestion matrix $\boldsymbol{Y}$.*

3. DATA COLLECTION

In this paper we focus on studying the traffic congestions in Chicago. However, our method can be easily generalized to study traffic congestions of other cities. In this section, we first describe how we collect and process traffic event tweets. Next we introduce how to utilize and model other useful information including road features and social events.

3.1 Twitter Data Collection

We collect traffic event tweets from two types of data sources: particular Twitter accounts operated by official traffic departments and general user accounts.

Traffic Authority Account. Currently, it is common for official transportation departments of big cities to release real-time traffic information to the public through Twitter. For example, *ChiTraTracker* is a Twitter account operated by Chicago Transportation Authority (CTA) aiming at posting real-time traffic information. We call such accounts as traffic authority accounts. The advantages of the tweets from these accounts are that they are formal and easier to process, and the exact location and time information are usually explicitly given. For example, one such tweet can be like "*Heavy Traffic on NB Western: Fullerton to Kennedy Expy. 06:15 pm 02/13/2015*". We can easily extract the road segment, traffic event category, and time information from the tweet. The disadvantage is that the spatial coverage is limited. They mostly only focus on principle arterial roads. We identify 11 such Twitter accounts related to Chicago: *ChicagoDrives*, *ChiTraTracker*, *roadnowChicago*, *traffic_Chicago*, *CMoleChicago*, *IDOT_Illinois*, *WGNtraffic*, *TotalTrafficCHI*, *GeoTrafficChi*, *roadnowil*, and *rosalindrossi*. For each account, we crawl all the posted tweets from April 2014 to May 2015.

General Sensor User. We also select the Twitter users registered in Chicago based on their profiles, and crawl their tweets. In all we get more than 100,000 such users and we obtain more than 32.3 million corresponding tweets. Next, two major steps are conducted for data preprocessing. 1) *traffic event tweets Identification.* We select traffic event tweets from all the crawled tweets which match at least one term of the predefined vocabularies: "stuck", "congestion", "jam", "crowded", "pedestrain", "driver", "accident", "crash", "road blocked", "road construction", "slow traffic", "heavy traffic", and "disabled vehicle". Based on the keywords contained in the tweets, we also can identify the traffic event category. 2) *Tweet Geocoding.* We then geocode tweets to the road segments by matching their geo-tags and text content. Some tweets are geo-tagged. By combing the the geo-coordinates of tweets and the direction mentioned in the content, we can geocode the tweets to the road segments. For most tweets without geo-tags, we need to first identify the streets, landmarks, and direction information from the content by using gazetteer, and then geocode them to the road segments. Note that many tweets only mention a street name like "*Heavy traffic on State St*". It is hard to geocode it to an exact road segment, and we simply consider all the road segments of State Street are in congestion. For the tweets without any location information, we drop them.

In all we obtain 245,568 traffic event tweets. 163,742 of them are related to traffic congestion, 77,454 are related to accident, and 4,372 are reporting other traffic events such as road construction and road closure. We categorize these tweets into three types: congestion, accident and others. Other traffic events include road construction, road closure, etc. Figure 2(a) depicts the hourly number of congestion related tweets and accident related tweets. One can see that two peaks appear in rush hours of a day: 7 am to 8 am and 4 pm to 5 pm, which implies tweets could be a good sources for traffic information. Figure 2(b) shows the hourly sparsity of congestions reported by tweets. The sparsity is defined as $s_t = \frac{n_t}{N_t}$, where n_t is the number of congestions in time t reported by tweets and N_t is the total number of congestions in time t. The ground truth N_t comes from the CTA bus GPS data which will be described in detail later. One can see that the congestion related tweets in the night is sparser than that in the day time. On average, about 12% of congestions can be directly identified from tweets.

3.2 Road Features and Social Events

Road features are widely used for traffic estimation and predictition [18, 25]. Here we use two types of road features: road physical features, and point of interest (POI). We also extract social events from Twitter and study whether incorporating social events could help better estimate traffic congestion.

Physical features of a road segment. 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 $r.dir$, and the number of intersections $r.inter$. These features f of a road segment r is modeled as a vector $f_r = (r.len, r.lane, r.way, r.dir, r.inter)$.

POI features. A point of interest (POI) is a venue in a physical world that someone may find useful or interesting, like a shopping mall, a theater, or a hospital [25]. Each POI is associated with many attributes including the name, address, coordinates, and categories. In this paper, we extract

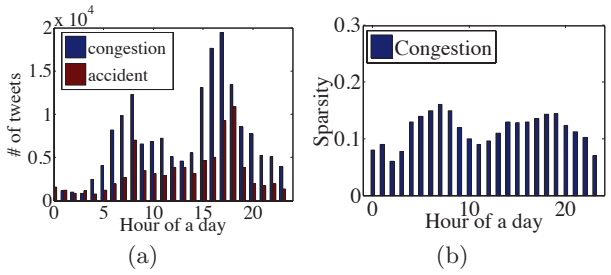


Figure 2: (a) Hourly # of tweets related to congestion and accident, and (b) hourly congestion sparsity of tweets.

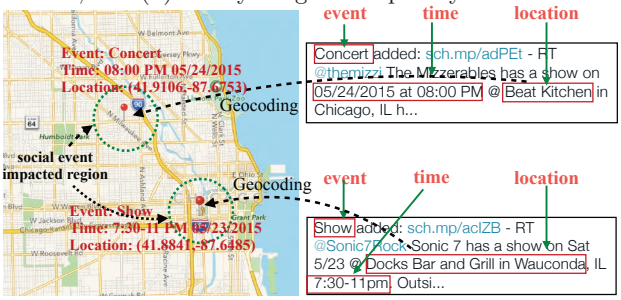


Figure 3: Social events extraction and geocoding

eight types of POIs: *Schools*, *Hospitals*, *Museums*, *Libraries*, *Parks*, *Police Stations*, *Parking zones*, and *Market & Malls*. For each road segment r, we identify all the POIs nearby and formulate them as a POI feature vector f_p^r. For instance, assuming there are 1 parking zone and 1 shopping mall nearby r, its POI feature thus is $f_p^r = (0, 0, 0, 0, 0, 0, 1, 1)$.

Social Events. There are a lot of geo-tagged tweets related to various social events. For example, *Chicago Events* is a Twitter account focusing on posting various social events information in Chicago. Taking the tweet *"Merger Party @ The Velvet Lounge 67 E Cermak Rd - 6pm-9pm"* as an example, we can easily extract the event type (*Party*), time (*6am-9pm*) and location (*67 E Cermak Rd*) information. 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. We group the social event related tweets into three types: *parties*, *music shows*, and *sports*.

For each such tweet, we first extract the event type, time, and place information, and then extract 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 using a two-dimensional Gaussian model to measure the impact intensity of the event on the nearby road segments based on their Euclidean distance, *i.e.*,

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

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.

4. CONGESTION CORRELATION MINING

Mining frequent spatio-temporal sequential patterns has been a focused theme in data mining research for decades [2, 21]. The main aim of this problem is to find sequences of events that occur frequently in spatio-temporal datasets. If we consider each traffic congestion event e_i with the time and location information as an item in the traffic congestion event database E, we can discover which road segments near to each other are more likely to be congested simultaneously by spatio-temporal frequent pattern mining algorithms. The spatial and temporal correlations might be potentially helpful for the congestion estimation task.

We first give some definitions as follows.

DEFINITION 5. ***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 reachs p_j in no more than k hops.*

DEFINITION 6. ***Support of road segment*** r_i ***in congestion***. *Given a set of road segment congestion event E. The support of the road segment r_i in congestion is defined as the possibility that a member of E whose road segment contains the road segment r_i.*

DEFINITION 7. ***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 constrains 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, i.e. $r_i \in N(r_j, k)$; and 2) the difference of the event time t_i, t_j is less than η, i.e. $|t_i - t_j| < \eta$, where k and η are the predefined spatio-temporal thresholds.*

DEFINITION 8. ***Confidence of congestion co-occurrence of road segments*** $\{r_i, r_j\}$. *Given the spatio-temporal thresholds k and η. The confidence of congestion co-occurrence of road segments $\{r_i, r_j\}$ is the probability of the two road segments co-occurring congestion in the congestion event database E. It can be calculated by*

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

Here $cardinality(r_i \xrightarrow{k,\eta} r_j)$ denotes how many times that road segments r_i and r_j co-occur congestion in the congestion event database E.

DEFINITION 9. (k, η) ***neighborhood road co-occurrence pattern in congestion.*** *A (k, η) neighborhood road co-occurrence pattern in congestion is of the form: $r_i \xrightarrow{k,\eta} r_j$ $(minsup, minconf)$, where k and η are the spatio-temporal thresholds, $minsup$ and $minconf$ are the user-specified minimum support and minimum confidence.*

We have discussed two road segments co-occurrence patterns in congestion. Above 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 only give a brief introduction on how to compute the probability of multiple road segments co-occurrence patterns in congestions.

DEFINITION 10. ***The probability*** $Pr(\{r_i, ..., r_m\} \xrightarrow{k,\eta} r_j)$ ***of the pattern*** $\{r_i, ..., r_m\} \xrightarrow{k,\eta} r_j$ *is the probability of finding r_j in the (k, η) neighborhood of the congestion road segment set $\{r_i, ..., r_m\}$.*

We propose to use the sequence rule to add r_j to the congestion road segment set $\{r_i, ..., r_m\}$ and merge them into the same pattern. In the sequence rule, a new congestion road segment belongs to an existing road congestion pattern if it co-occurs with at least one road segment. That is, $\exists r_u \in \{r_i, ..., r_m\}$, $r_u \xrightarrow{k,\eta} r_j(minsup, minconf)$. The probability of the pattern can be computed by

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

With above defined pattern rules, we use an Apriori [1] like algorithm to mine all the frequent patterns. Due to space limitation, we omit the details of the algorithm. Based on the discovered patterns, we construct the road segment correlation matrix $\mathbf{Z}$ as shown in the upper part of Figure 1. Each entry z_{ij} of $\mathbf{Z}$ is the probability of congestion pattern of road segments r_i and r_j, *i.e.*, $z_{ij} = Pr(r_i \xrightarrow{k,\eta} r_j)$.

5. CTCE MODEL

As our model use tensor factorization techniques to facilitate matrix factorization, we first give a quick review of some tensor notations and operations. 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 paper, we use the third-order tensor. Scalars are denoted by lowercase letters, *e.g.*, a. Vectors are denoted by boldface lowercase letters, *e.g.*, $\mathbf{a}$. Matrices are denoted by boldface capital letters, *e.g.*, $\mathbf{X}$. Tensors are denoted by calligraphic letters, *e.g.*, $\mathcal{A}$. 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. Alternatively, the ith row of a matrix, $\mathbf{a}_{i:}$, may be denoted more compactly as $\mathbf{a}_i$.

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 *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:

$$\mathcal{A} = \mathcal{C} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$$

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.

5.1 Coupled Matrix and Tensor Factorization

The main idea of using matrix factorization to estimate citywide traffic congestion is: given a very sparse road congestion matrix $\mathbf{Y}$, try to complete $\mathbf{Y}$ by factorizing it into two low rank latent matrices. As shown in Figure 4, $\mathcal{A} \in \mathbb{R}^{N\times M\times L}$ 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, 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.

To achieve a higher accuracy of filling in the missing entries of $\mathbf{Y}$, we factorize it collaboratively with the matrices $\mathbf{X}$, $\mathbf{Z}$ and tensor $\mathcal{A}$. As illustrated in Figure 4, 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. Likewise, the congestion matrix $\mathbf{Y}$ can be factorized into the multiplication of two matrices, $\mathbf{Y} = \mathbf{U} \times \mathbf{V}^{\mathrm{T}}$, where $\mathbf{V} \in \mathbb{R}^{M\times R}$ is a low rank latent factor matrix for time slots. The event tensor can be factorized as $\mathcal{C} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$, where $\mathbf{W} \in \mathbb{R}^{L\times T}$ is a low rank latent factor matrix for event categories.

The objective function is defined as follows,

$$\mathcal{L}(\mathbf{U},\mathbf{V},\mathbf{W},\mathcal{C},\mathbf{F}) = \frac{1}{2}||\mathbf{Y}-\mathbf{U}\mathbf{V}^{\mathrm{T}}||^2 + \frac{\lambda_1}{2}||\mathcal{A}-\mathcal{C}\times_1\mathbf{U}\times_2\mathbf{V}\times_3\mathbf{W}||^2 + \frac{\lambda_2}{2}||\mathbf{X}-\mathbf{U}\mathbf{F}||^2 + \frac{\lambda_3}{2}\mathrm{tr}(\mathbf{U}^{\mathrm{T}}\mathbf{L}_z\mathbf{U}) + \frac{\lambda_4}{2}(||\mathbf{U}||^2 + ||\mathbf{V}||^2 + ||\mathbf{W}||^2 + ||\mathcal{C}||^2 + ||\mathbf{F}||^2)$$

where $\mathrm{tr}(\cdot)$ denotes the matrix traces, $||\mathbf{Y}-\mathbf{U}\mathbf{V}^{\mathrm{T}}||^2$ is to control the error of factorization of $\mathbf{Y}$, $||\mathcal{A}-\mathcal{C}\times_1\mathbf{U}\times_2\mathbf{V}\times_3\mathbf{W}||^2$ is to control the error of factorizing tensor $\mathcal{A}$, $||\mathbf{X}-\mathbf{U}\mathbf{F}||^2$ is to control the error of factorization 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 controlling the contribution of each part during the collaborative factorization. $\mathbf{L}_z = \mathbf{D} - \mathbf{Z}$ 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 obtained through the following deduction, which guarantees 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 $\mathbf{U}$.

$$\begin{aligned}
&\frac{1}{2}\sum_{i,j}||\mathbf{u}_i-\mathbf{u}_j||_2^2 z_{ij} = \sum_{i,j}\mathbf{u}_i z_{ij}\mathbf{u}_i^{\mathrm{T}} - \sum_{i,j}\mathbf{u}_i z_{ij}\mathbf{u}_j^{\mathrm{T}} \\
&= \sum_i \mathbf{u}_i d_{ii}\mathbf{u}_i^{\mathrm{T}} - \sum_{i,j}\mathbf{u}_i z_{ij}\mathbf{u}_j^{\mathrm{T}} \\
&= \mathrm{tr}(\mathbf{U}^T(\mathbf{D}-\mathbf{Z})\mathbf{U}) = \mathrm{tr}(\mathbf{U}^{\mathrm{T}}\mathbf{L}_z\mathbf{U})
\end{aligned}$$

5.2 Coupled Matrix and Tensor Factorization with Historical Data

As the road congestion matrix $\mathbf{Y}$ and the road event tensor $\mathcal{A}$ may be sparse in a day, we also use historical Twitter data for help. We utilize the historical congestion probability matrix $\mathbf{Y}^h$ and the historical road event probability tensor $\mathcal{A}^h$. Each entry y_{ij}^h of $\mathbf{Y}^h$ is the probability that the road segment r_i is in congestion in the time slot t_j of a day; and each entry a_{ijk}^h of $\mathcal{A}^h$ denotes the possibility that an event e_k occurs on (traffic event) or near (social event) road

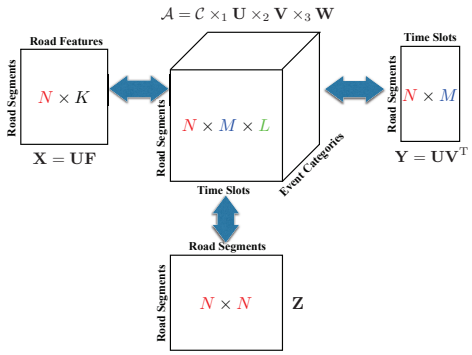


Figure 4: Coupled matrix and tensor factorization

segment r_i in the time slot t_j of a day. By combining the historical data, we have the following objective function

$$\mathcal{L}(\mathbf{U},\mathbf{V},\mathbf{W},\mathcal{C},\mathbf{F}) = \frac{1}{2}||\mathbf{Y}-\mathbf{U}\mathbf{V}^{\mathrm{T}}||^2 + \frac{\lambda_1}{2}||\mathbf{Y}^h-\mathbf{U}\mathbf{V}^{\mathrm{T}}||^2 + \frac{\lambda_2}{2}||\mathcal{A}-\mathcal{C}\times_1\mathbf{U}\times_2\mathbf{V}\times_3\mathbf{W}||^2 + \frac{\lambda_3}{2}||\mathcal{A}^h-\mathcal{C}\times_1\mathbf{U}\times_2\mathbf{V}\times_3\mathbf{W}||^2 + \frac{\lambda_4}{2}||\mathbf{X}-\mathbf{U}\mathbf{F}||^2 + \frac{\lambda_5}{2}\mathrm{tr}(\mathbf{U}^{\mathrm{T}}\mathbf{L}_z\mathbf{U}) + \frac{\lambda_6}{2}(||\mathbf{U}||^2 + ||\mathbf{V}||^2 + ||\mathbf{W}||^2 + ||\mathcal{C}||^2 + ||\mathbf{F}||^2)$$

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 idea here is that the congestion states of the roads should be similar to their historical congestion states. A road segment r_i is more likely to be in congestion than the road segment r_j if r_i's congestion probability is much higher than r_j historically. Therefore, we assume that the congestion matrix $\mathbf{Y}$ in a day 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 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 element-wise optimization algorithm to iteratively update each entry in the matrices and tensor independently by gradient descent [9, 18, 25]. More specifically, we have the gradient for each variable as follows:

$$\begin{aligned}
\nabla_{\mathbf{u}_{i:}}\mathcal{L} &= (\mathbf{u}_{i:}\mathbf{V}^{\mathrm{T}} - \mathbf{y}_{i:})\mathbf{V} + \lambda_1(\mathbf{u}_{i:}\mathbf{V}^{\mathrm{T}} - \mathbf{y}_{i:}^h)\mathbf{V} + \\
&\lambda_2(\mathcal{C}\times_1\mathbf{u}_{i:}\times_2\mathbf{v}_{j:}\times_3\mathbf{w}_{k:} - a_{ijk})\mathcal{C}\times_2\mathbf{v}_{j:}\times_3\mathbf{w}_{k:} + \\
&\lambda_3(\mathcal{C}\times_1\mathbf{u}_{i:}\times_2\mathbf{v}_{j:}\times_3\mathbf{w}_{k:} - a_{ijk}^h)\mathcal{C}\times_2\mathbf{v}_{j:}\times_3\mathbf{w}_{k:} + \\
&\lambda_4(\mathbf{u}_{i:}\mathbf{F} - \mathbf{x}_{i:})\mathbf{F}^{\mathrm{T}} + \lambda_5(\mathbf{L}_z\mathbf{U})_{i:} + \lambda_6\mathbf{u}_{i:}
\end{aligned}$$

$$\begin{aligned}
\nabla_{\mathbf{v}_{j:}}\mathcal{L} &= (\mathbf{v}_{j:}\mathbf{U}^{\mathrm{T}} - \mathbf{y}_{:j}{}^{\mathrm{T}})\mathbf{U} + \lambda_1(\mathbf{v}_{j:}\mathbf{U}^{\mathrm{T}} - \mathbf{y}^h{}_{:j}{}^{\mathrm{T}})\mathbf{U} + \\
&\lambda_2(\mathcal{C}\times_1\mathbf{u}_{i:}\times_2\mathbf{v}_{j:}\times_3\mathbf{w}_{k:} - a_{ijk})\mathcal{C}\times_1\mathbf{u}_{i:}\times_3\mathbf{w}_{k:} + \\
&\lambda_3(\mathcal{C}\times_1\mathbf{u}_{i:}\times_2\mathbf{v}_{j:}\times_3\mathbf{w}_{k:} - a_{ijk}^h)\mathcal{C}\times_1\mathbf{u}_{i:}\times_3\mathbf{w}_{k:} + \lambda_6\mathbf{v}_{j:}
\end{aligned}$$

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

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

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

Details of the algorithm is given in Algorithm 1.

Algorithm 1 Coupled Matrix and Tensor Factorization

Input: Tensors $\mathcal{A}$, $\mathcal{A}^h$, matrices $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Y}^h$, $\mathbf{Z}$, an error threshold ε, and the max iteration times $IterMax$
Output: Low rank matrices $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$, $\mathbf{F}$, core tensor $\mathcal{C}$

Initialize $\mathbf{U} \in \mathbb{R}^{N\times R}$, $\mathbf{V} \in \mathbb{R}^{M\times R}$, $\mathbf{W} \in \mathbb{R}^{L\times T}$, $\mathbf{F} \in \mathbb{R}^{R\times K}$, and $\mathcal{C} \in \mathbb{R}^{R\times R\times T}$ with small random values
Set η as step size
$d_{ii} = \sum_i z_{ij}$
$\mathbf{L}_z = \mathbf{D} - \mathbf{Z}$
while ($t < IterMax$ and $Loss^t - Loss^{t+1} > \varepsilon$) **do**
  for each $y_{ij} \neq 0$ **do**
    for each a_{ijk} **do**
      Get $\nabla_{\mathbf{u}_{i:}}\mathcal{L}$, $\nabla_{\mathbf{v}_{j:}}\mathcal{L}$, $\nabla_{\mathbf{w}_{k:}}\mathcal{L}$, $\nabla_{\mathcal{C}}\mathcal{L}$, $\nabla_{\mathbf{F}}\mathcal{L}$
      $\mathbf{u}_{i:}{}^{t+1} = \mathbf{u}_{i:}{}^{t} - \eta\nabla_{\mathbf{u}_{i:}{}^{t}}\mathcal{L}$
      $\mathbf{v}_{j:}{}^{t+1} = \mathbf{v}_{j:}{}^{t} - \eta\nabla_{\mathbf{v}_{j:}{}^{t}}\mathcal{L}$
      $\mathbf{w}_{k:}{}^{t+1} = \mathbf{w}_{k:}{}^{t} - \eta\nabla_{\mathbf{w}_{k:}{}^{t}}\mathcal{L}$
      $\mathcal{C}^{t+1} = \mathcal{C}^{t} - \eta\nabla_{\mathcal{C}^t}\mathcal{L}$
      $\mathbf{F}^{t+1} = \mathbf{F}^{t} - \eta\nabla_{\mathbf{F}^t}\mathcal{L}$
    end for
  end for
end while
return $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$, $\mathcal{C}$, and $\mathbf{F}$

6. EVALUATION

6.1 Dataset

We use the GPS traces data of Chicago Transit Authority (CTA) public passenger buses as our 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. Congestion estimates will be produced every ten minutes to measure the current estimated speed for 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 which contains more than 500 million records[1]. 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 5 probe records; otherwise we use a weighted average between the historical and real time speed. We first need to map the speed of buses to several discrete congestion states. Note that except for a very few segments, speed on city arterials is limited to 30 mph by ordinance. We use the 5-state traffic conditions in Chicago defined by CTA: heavy congestion, medium-heavy congestion, medium, light, and flow conditions. The speeds are corresponding to 0-10, 10-15, 15-20, 20-25, and over 25 miles. We assign the 5 congestion states with values 1.0, 0.8, 0.6, 0.4, and 0.2, respectively. Higher value means heavier congestion. Roughly we consider a road segment is in congestion if the speed is lower than 15 mph.

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

Table 1: Precision@k of different methods

Precision@k on Weekday					
	top-10	top-20	top-30	top-100	top-200
CF	0.500	0.437	0.414	0.412	0.325
TSE	0.821	0.786	0.774	0.676	0.577
CTCE-$\mathcal{A}-\mathcal{A}^h$	0.844	0.836	0.823	0.775	0.715
CTCE-Y^h	0.842	0.810	0.822	0.785	**0.722**
CTCE	**0.873**	**0.866**	**0.853**	**0.824**	0.718
Precision@k on Weekend					
	top-10	top-20	top-30	top-100	top-200
CF	0.485	0.436	0.472	0.415	0.346
TSE	0.812	0.821	0.785	0.735	0.580
CTCE-$\mathcal{A}-\mathcal{A}^h$	0.847	**0.842**	0.821	0.742	0.644
CTCE-Y^h	0.825	0.812	0.786	0.737	0.654
CTCE	**0.854**	0.834	**0.822**	**0.754**	**0.678**

We use all the crawled tweets to construct historical congestion probability matrix $\mathbf{Y}^h$, the correlation matrix $\mathbf{Z}$, and historical event tensor $\mathcal{A}^h$. We then use the tweets and CTA GPS data from 11/25/2014 to 12/15/2014 as the training data to obtain the best parameter settings, and the data from 12/16/2014 to 12/30/2014 as the testing data.

6.2 Baselines and Evaluation Metrics

We use the following methods as baselines.

- TSE. TSE is a Traffic Speed Estimation model based on a context aware matrix factorization approach [18]. As the main idea of TSE is similar to our model, we use it as a baseline to address the studied problem.
- CF. Collaborative filtering (CF) is widely used in recommendation [15]. We can consider the congestion estimation task as a CF problem by simply factorizing the road congestion matrix only.
- CTCE-Y^h. To study whether the historical road congestion information could help, we use the CTCE model without the historical road congestion matrix as a baseline. In the baseline, we set $\lambda_1 = 0$.
- CTCE-$\mathcal{A}$-$\mathcal{A}^h$. To study whether event information can improve the performance, we also use the CTCE model without the road-event-time tensors $\mathcal{A}$ and $\mathcal{A}^h$ as a baseline. In this baseline, we set $\lambda_2 = 0$ and $\lambda_3 = 0$.

We use *precision*, *recall*, *F1-score*, and *precision*@k as the evaluation metrics. We first complete the road congestion matrix $\mathbf{Y}$ by multiplying the low rank matrices $\mathbf{U}$ and $\mathbf{V}^{\mathrm{T}}$, and then rank the values of all the entries in $\mathbf{Y}$. We consider the road segments with top-k entry values are in congestion.

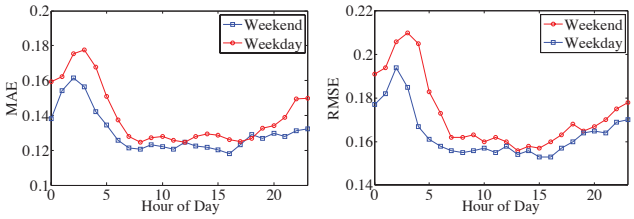
Although tweets usually roughly report whether a road segment is in congestion or not, but ignore the congestion states, CTCE model can estimate different states based on the values in the completed matrix $\mathbf{Y}$. Hence we also use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) as evaluation metrics by normalizing the entry values in the range from 0.2 to 1.

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

where y_i is the ground truth and $\hat{y_i}$ is the estimated value.

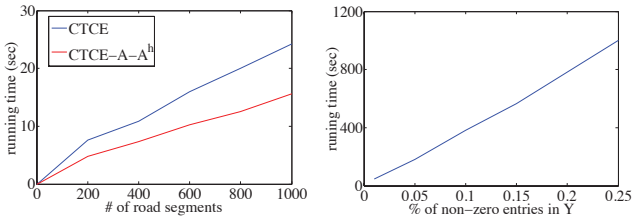
6.3 Quantitive Evaluation

Evaluation with precision@k. We first evaluate whether the proposed CTCE model can accurately estimate whether



(a) MAE (b) RMSE

Figure 5: MAE and RMSE in each hour



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

Figure 6: Scalability study of CTCE

Table 3: Robustness testing by adding noisy data

Method	% noise	Precision	Recall	F1-score
CTCE	0%	0.475	0.485	0.480
	2%	0.524	0.427	0.470
	5%	0.664	0.336	0.446
	10%	0.739	0.287	0.413
CF	0%	0.176	0.995	0.299
	2%	0.166	0.823	0.277
	5%	0.166	0.767	0.273
	10%	0.162	0.634	0.248

a road segment is in congestion or not. Given the estimated road congestion matrix $\mathbf{Y}$ and the k, we first rank all the entry values of $\mathbf{Y}$. Then we consider the top-k entries in $\mathbf{Y}$ are in congestion.

Table 1 shows the results of different methods over various k. As the traffic conditions on weekdays and weekends may be quiet different, we present the results by weekday and weekend separately. The best results are highlighted by bold. One can see that in most cases the proposed CTCE model achieves the best performance with only two exceptions on the precision@200 on weekday and precision @20 on weekend. The performance of CF model is significantly lower than all the other methods. This is not surprising as only the instant congestion information are explored by CF. TSE model and CTCE-Y^h model are both inferior to CTCE model, which implies historical information and road segment correlation in congestion are useful for the instant estimation task. CTCE model is also better than CTCE-$\mathcal{A}$-$\mathcal{A}^h$ model, thus we can infer that traffic and social events information do help better estimate traffic congestions.

Performance analysis in each hour. The traffic conditions on different hours of a day may also be distinct significantly. In rush hours, we have more available traffic event tweets and the estimation results may be more reliable, *e.g.* from 6-9 am. In the deep night, on the contrary, the performance may be bad due to sparser data. To investigate whether the estimation performance varies in different hours of a day, we show the hourly MAE and RMSE of the proposed CTCE model in Figure 5. One can observe that in the time interval 0:00-5:00 am, the estimation performance is undesirable. The estimations on rush hours of a day are much more stable with lower MAE (around 0.13) and RMAE (around 0.16). This is mainly because very few people travel in deep night and the available traffic event tweets is sparser than that in the day time. One can also observe that the estimation performance on weekdays is consistently better than that on weekends. This also makes sense as people's travel patterns are more regular in weekdays than in weekends. Another interesting observation is that the weekend figures are pushed to the right. This is probably because on weekends people start their activities on the road later to spend more time at home.

Performance evaluation in rush hours. Based on the fact that people may concern more on the traffic conditions in rush hours, we report the experimental results of different methods in the time intervals 6:00-10:00 am and 15:00-19:00 pm in Table 2. The figures in bold are the best results. One can see that the proposed CTCE model outperforms all the baselines in most cases. Similar to our previous result, it also shows the traffic conditions in weekday are easier to predict than that in weekend. For example, the MAE in the hour 6:00-7:00 am estimated by CTCE is 0.1259 for weekday, and the number is 0.1375 for weekend. Similar result can also be found for RMSE. CF is significantly inferior to other methods. TSE is better than CF, but significantly inferior to other methods. CTCE performs consistently better than CTCE-Y^h and CTCE-$\mathcal{A}$-$\mathcal{A}^h$, which demonstrates the the effectiveness of the proposed matrix and tensor factorization model in capturing and integrating the historical congestion and event information in rush hours.

6.4 Scalability Analysis

Figure 6(a) shows the running time trends of CTCE and CTCE-$\mathcal{A}$-$\mathcal{A}^h$ on a PC with mac OSX 10.9.5 system, 4 GB memory and dual-core i5 processor. One can see that the running time of both models linearly increase with the increase of road segment size. Although the the running time of CTCE is consistently slightly longer that CTCE-$\mathcal{A}$-$\mathcal{A}^h$ due to tensor factorization, the figure shows that CTCE is not time consuming. For the congestion matrix $\mathbf{Y}$ with 1000 road segments, the running time of CTCE is less than 30 seconds. This is mainly because the matrices are very sparse and only a small proportion of entires have non-zero values. To further study the effect of congestion matrix $\mathbf{Y}$ sparsity on the running time, we plot the running times of CTCE under various proportions of non-zero entires in $\mathbf{Y}$. One can see that a denser $\mathbf{Y}$ gives rise to much longer running time of CTCE: more than 1000 seconds are needed for a $\mathbf{Y}$ with 20% non-zero entries. Fortunately, in real case the congestion matrix $\mathbf{Y}$ is usually very sparse with less 5% non-zero entries. Thus, CTCE model can easily scale to large data with thousands of road segments in a city.

6.5 Robustness Analysis with Noisy Data

As mentioned in the data collection part, the tweets may be noisy and the locations of the traffic events may not be precisely identified. To examine the sensitively of the proposed CTCE model on noisy data, we study the robustness of CTCE by randomly adding some noise to the congestion matrix $\mathbf{Y}$. For a randomly selected entry y_{ij}, if $y_{ij} = 1$, we set $y_{ij} = 0$; otherwise we set $y_{ij} = 1$. Table 3 shows the

Table 2: MAE and RMSE of different methods in rush hours

	6:00-7:00	7:00-8:00	8:00-9:00	9:00-10:00	15:00-16:00	16:00-17:00	17:00-18:00	18:00-19:00
MAE on Weekday								
CF	0.2233	0.2343	0.2682	0.2737	0.2250	0.2284	0.2109	0.2566
TSE	0.2178	0.2047	0.2016	0.2260	0.1653	0.2016	0.1627	0.1699
CTCE-$\mathcal{A} - \mathcal{A}^h$	0.1376	0.1316	0.1289	0.1345	0.1345	0.1270	0.1308	**0.1256**
CTCE-Y^h	0.1742	0.1776	0.1657	0.1726	0.1725	0.1628	0.1764	0.1772
CTCE	**0.1259**	**0.1215**	**0.1208**	**0.1232**	**0.1201**	**0.1180**	**0.1233**	0.1267
MAE on Weekend								
CF	0.2542	0.2913	0.2252	0.2313	0.2136	0.2223	0.2008	0.2187
TSE	0.2335	0.2123	0.2059	0.1980	0.1674	0.1378	0.1866	0.1968
CTCE-$\mathcal{A} - \mathcal{A}^h$	0.1424	0.1356	**0.1328**	0.1455	0.1432	0.1320	0.1275	0.1283
CTCE-Y^h	0.1875	0.1746	0.1833	0.1725	0.1722	0.1683	0.1794	0.1675
CTCE	**0.1375**	**0.1281**	0.1345	**0.1271**	**0.1287**	**0.1260**	**0.1250**	**0.1268**
RMSE on Weekday								
CF	0.2568	0.2715	0.2953	0.3108	0.2659	0.2650	0.3162	0.3167
TSE	0.2348	0.2163	0.2495	0.2695	0.1977	0.2284	0.1865	0.1932
CTCE-$\mathcal{A} - \mathcal{A}^h$	0.1624	0.1770	**0.1435**	0.1658	0.1820	0.1766	0.1675	**0.1564**
CTCE-Y^h	0.1982	0.1974	0.1856	0.2015	0.1847	0.2004	0.1973	0.1988
CTCE	**0.1582**	**0.1563**	0.1556	**0.1562**	**0.1537**	**0.1543**	**0.1567**	0.1643
RMSE on Weekend								
CF	0.2967	0.3387	0.3452	0.3114	0.2664	0.2649	0.2565	0.3127
TSE	0.2690	0.2701	0.2484	0.2331	0.2105	0.1842	0.2361	0.2360
CTCE-$\mathcal{A} - \mathcal{A}^h$	0.1927	0.1866	**0.1635**	0.1840	0.2105	0.1826	0.1725	0.1745
CTCE-Y^h	0.2013	0.1988	0.2132	0.1976	0.2055	0.1965	0.2125	0.1876
CTCE	**0.1732**	**0.1654**	0.1637	**0.1604**	**0.1572**	**0.1620**	**0.1634**	**0.1681**

results. One can see the F1-scores achieved on the data with 2% and 5% noise are comparable to that achieved on clean data, which shows the robustness of the model. Even with 10% noisy data, the performance is acceptable with 0.413 F1-score. This is mainly because both the historical data and congestion correlation provide additional information for the real-time congestion estimation. As a comparison, we also give the results of CF which purely factorize the congestion matrix $\mathbf{Y}$. One can see that: 1) the F1-score is consistently much lower than CTCE, and 2) with the increase of noisy data, the performance drops significantly.

6.6 Case Study

Figure 7 presents the estimated traffic states and the ground truth (GT) on three road segments that are highly likely to be congested: *SB Halsted from Chicago to Grand*, *NB State from Congress to Wacker*, and *NB Western from Fullerton to Kennedy Expy* on the day 12/10/2014. Red lines are the ground truth with three traffic states: congestion, normal, and unknown with the values 1, 0, and -1, respectively. In some time intervals lacking ground truth due to data sparsity, we consider the traffic state is unknown and set the corresponding value to -1. Blue dashed lines represent the estimated traffic states with value 0.8 denoting congestion and -0.2 denoting normal. One can see that the three road segments are all likely to be congested, especially the road segment in *NB State*. One can also see that CTCE can estimate the traffic congestion state with an accuracy around 80%. For each road segment, there is a traffic event happening on that day. The road segment *SB Halsted from Jackson to Harison* was closed that day and accidents happened at 4:15 pm and 5:38 am on *NB State* and *NB Western*, respectively. The three road segments are all congested in the time intervals of the traffic events, and CTCE model captures the event information and gives accurate estimations.

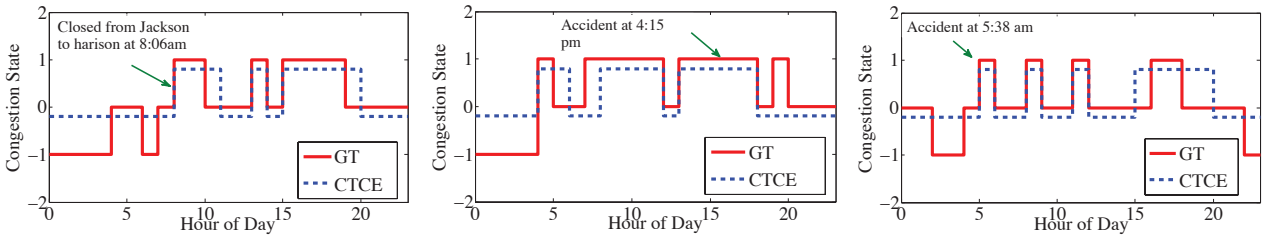
7. RELATED WORK

Traffic estimation has been extensively studied in the area of traffic engineering [6, 13, 14]. Conventional methods rely on GPS equipment on vehicles, loop detectors, cameras, and other instructions to obtain real-time traffic data, and use the data to estimate the traffic speed, density, and volume. Previous researches can be roughly categorized into traffic modeling on individual roads [8, 11, 14] and on a road network [13, 18, 23]. Models of traffic estimation on an individual road usually employ a Fundamental Diagram [8] to learn the relations among vehicle speed, traffic density, and volume for a particular road by using sensor data. To estimate the traffic density at unmonitored locations along a highway, [11] proposed a macroscopic traffic flow model SMM by using the loop detector data. [14] proposed a Gaussian Mixture Hidden Markov Models (GM-HMM) to detect traffic condition with the MPEG video data. Recently, many research attentions have also been devoted to investigating how to estimate or model traffic conditions on a road network [6, 13, 18, 23]. Such models mainly utilized the Floating Car Data (FCD) [23] generated by the GPS sensors equipped in vehicles. [6] studied the problem of using real-time FCD data based on traces of GPS positions to predict the traffic on Italian motorway network. [23] investigated how to use the trajectories of taxis collected by GPS to efficiently find driving directions for drivers.

With the prevalence of social media, more and more works have tried to utilize Twitter as a new data source for detecting or monitoring traffic events [17, 5, 7, 19, 4, 3, 10]. Most of these works focused on studying either how to extract and visualize the traffic event information from tweets [5, 17, 10, 7] or how to locate the traffic events mentioned in the tweets [4, 19]. Classification and event detection techniques are usually used to detect and categorize traffic related tweets with various event types [5, 17]. Although some geo-tagged tweets have location information, for most tweets the location information should be extracted from the content. To this aim, some text mining approaches combined with geocoding are employed [4, 19].

8. CONCLUSION AND FUTURE WORK

This paper proposes a novel citywide traffic congestion es-



(a) SB Halsted(Chicago to Grand) (b) NB State(Congress to Wacker) (c) NB Western(Fullerton to Kennedy)

Figure 7: Case study of CTCE and the ground truth on three road segments on 12/10/2014

timation schema by exploring Twitter data. Our research is an important complementary to conventional methods for improving many real applications. The studied problem is very challenging, however, considering the sparsity and diversity of the data from Twitter. To address the challenges, we analyze and utilize the traffic congestion correlations among road segments and integrate multi-typed traffic and social events to enrich the sparse real-time tweets. We also propose a coupled matrix and tensor factorization algorithm to seamlessly combine the rich information for a better estimation model. Evaluations on real data in Chicago shows the effectiveness of the proposed method.

Potential directions of future work include extending the model to other urban computing applications, such as traffic prediction and route planning. It would also be interesting to investigate the combination of social media data and GPS probe data to enhance traffic monitoring, which could be a promising solution to improve the estimation accuracy while reduce cost on deploying sensors.

9. ACKNOWLEDGEMENTS

This work is supported in part by the National Natural Science Foundation of China (Grant Nos. 61170189, 61370126, 61202239, 61303017, 61503253), National High Technology Research and Development Program of China under grant (No. 2015AA016004), the Fund of the State Key Laboratory of Software Development Environment (No. SKLSDE-2015ZX-16), Microsoft Research Asia Fund (No. FY14-RES-OPP-105), US NSF through grants (CNS-1115234), Google Research Award, and the Pinnacle Lab at Singapore Management University.

10. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, 1994.

[2] H. Cao, N. Mamoulis, and D. W. Cheung. Mining frequent spatio-temporal sequential patterns. In *ICDM*, 2005.

[3] P.-T. Chen, F. Chen, and Z. Qian. Road traffic congestion monitoring in social media with hinge-loss markov random fields. In *ICDM*, 2014.

[4] E. M. Daly, F. Lecue, and V. Bicer. Westland row why so slow?: fusing social media and linked data sources for understanding real-time traffic conditions. In *IUI*, 2013.

[5] E. D'Andrea, P. Ducange, B. Lazzerini, and F. Marcelloni. Real-time detection of traffic from twitter stream analysis. *Intelligent Transportation Systems*, PP(99):1–15, 2015.

[6] C. de Fabritiis, R. Ragona, and G. Valenti. Traffic estimation and prediction based on real time floating car data. In *ITCS*, 2008.

[7] S. K. Endarnoto, S. Pradipta, A. S. Nugroho, and J. Purnama. Traffic condition information extraction and visualization from social media twitter for android mobile application. In *ICEEI*, 2011.

[8] D. Helbing. Traffic and related self-driven many-particle systems. *Reviews of modern physics*, 2001.

[9] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context aware collaborative filtering. In *RecSys*, 2010.

[10] M. Liu, K. Fu, C.-T. Lu, G. Chen, and H. Wang. A search and summary application for traffic events detection based on twitter data. In *SIGSPATIAL GIS*, 2014.

[11] L. Muñoz, X. Sun, R. Horowitz, and L. Alvarez. Traffic density estimation with the cell transmission model. In *the 2003 American Control Conference*, 2003.

[12] C. Ozkurt and F. Camci. Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks. *Mathematical and Computational Application*, 14(3):187–196, 2009.

[13] W. Pattara-Atikom, P. Pongpaibool, and S. Thajchayapong. Estimating road traffic congestion using vehicle velocity. In *ITS Telecommunications*, 2006.

[14] F. Porikli and X. Li. Traffic congestion estimation using hmm models without vehicle tracking. In *Intelligent Vehicles Symposium*, 2004.

[15] B. Sarwar, G. karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, 2001.

[16] D. Schrank, B. Eisele, and T. Lomax. *2012 URBAN MOBILITY REPORT Powered by INRIX Traffic Data*. 2012.

[17] A. Schulz, P. Ristoski, and H. Paulheim. I see a car crash: Real-time detection of small scale incidents in microblogs. In *ESWC*, 2013.

[18] J. Shang, Y. Zheng, W. Tong, E. Chang, and Y. Yu. Inferring gas consumption and pollution emission of vehicles throughout a city. In *KDD*, 2014.

[19] J. Sílvio S. Ribeiro, J. Clodoveu A. Davis, D. R. R. Oliveira, J. Wagner Meira, T. S. Gonçalves, and G. L. Pappa. Traffic observatory: a system to detect and locate traffic events and conditions using twitter. In *SIGSPATIAL LBSN*, 2012.

[20] S. Tao, V. Manolopoulos, S. Rodriguez, and A. Rusu. Real-time urban traffic state estimation with a-gps mobile phones as probes. *Journal of Transportation Technologies*, 2(1):22–31, 2012.

[21] I. Tsoukatos and D. Gunopulos. Efficient mining of spatiotemporal patterns. 2121:425–442, 2001.

[22] Y. Wang, Y. Zhu, and Z. He. Challenges and opportunities in exploiting large-scale gps probe data. In *Technical Report, HPL-2011-109*, 2011.

[23] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-drive: Driving directions based on taxi trajectories. In *SIGSPATIAL GIS*, 2010.

[24] Y. Zheng, L. Capra, O. Wolfson, and H. Yang. Urban computing: Concepts, methodologies, and applications. *ACM Trans. On Intelligent Systems and Technology*, 5(3), 2014.

[25] Y. Zheng, T. Liu, Y. Wang, Y. Zhu, Y. Liu, and E. Chang. Diagnosing new york city's noises with ubiquitous data. In *UbiComp*, 2014.