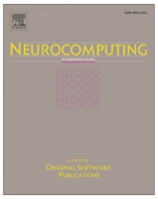
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Citywide road-network traffic monitoring using large-scale mobile signaling data

Qiuyang Huang[a], Yongjian Yang[a], Yuanbo Xu[a,*], Funing Yang[b], Zhilu Yuan[c,d], Yongxiong Sun[a]

[a] *College of Computer Science and Technology, Jilin University, Changchun 130012, China*
[b] *College of Applied Technology, Jilin University, Changchun, China*
[c] *Key Laboratory of Urban Land Resources Monitoring and Simulation, Ministry of Land and Resources of China, Shenzhen 518034, China*
[d] *Research Institute for Smart Cities, School of Architecture and Urban Planning, Shenzhen University, Shenzhen, China*

ARTICLE INFO

Article history:
Received 30 December 2019
Revised 22 May 2020
Accepted 24 July 2020
Available online 26 January 2021

Keywords:
Traffic monitoring
Signaling data
Road-network
Traffic condition

ABSTRACT

Road-network traffic monitoring on city-scale is critical for a wide range of applications, such as traffic forecasting, congestion identification, traffic safety, and urban planning, etc. Despite the fruitful research outcomes, however, most traffic monitoring models suffered from limited coverage, data sparsity, and data deviation, which leads to a biased and inaccurate result. With the widespread usage of mobile phones, mobile signaling data is of great value for various fields, especially for monitoring urban traffic. Thousands cell towers are distributed in the urban area, which can serve as ubiquitous sensors. Specifically, a mobile phone will passively generate a mobile signaling record that contains users' spatiotemporal information. When mobile phone users move with their phones, their phones will interact with cell towers and these towers can obtain their mobile signaling records. And these signaling records contain sufficient information for traffic monitoring. However, there also exists excessive noise in signaling records, which makes most monitoring models abandon these data. In this paper, we present the Urban-STM scheme, which utilizes large-scale anonymous and coarse-grained mobile signaling data to infer road-network traffic conditions. We apply our scheme to a real-world signaling dataset in Changchun city and present an extensive validation study based on 2000 taxicabs' GPS trajectories. Experiment results show that our scheme improves traffic monitoring performance in terms of coverage and accuracy.

1. Introduction

Road-network traffic monitoring plays a vital role in the Intelligent Transportation System (ITS), which can provide basic and necessary services for a wide range of applications, such as traffic forecasting, congestion identification, traffic safety, urban planning, etc. The main objective of road network traffic monitoring is to detect the average speed or traffic volume of each road in real-time. As an instance, a traffic monitoring system can set a threshold for a specific road, through the real-time monitoring of the road, the system can detect abnormal traffic conditions in time and issue an alarm, so as it can provide useful insights and guidelines for traffic congestion control in the current and future stages.

The conventional traffic monitoring method is to use the data collected by wireless sensing devices [1–4], which need to be installed on the crossings or segments, such as inductive loops, micro-loop probes, piezoelectric sensors, video detectors or radars. These methods can accurately perceive the traffic condition of the road. However, on the one hand, due to the high cost of equipment and installation, it cannot cover all roads, resulting in the problem of data sparsity; On the other hand, equipment failure and the poor signal will cause serious data missing. Other than that, most taxicabs and buses are equipped with GPS equipment, so, using vehicle GPS trajectories data to monitor road traffic conditions [5–8] has become the most common method. However, the disadvantage is the trajectories of taxicabs and buses have their own characteristics [9] and cannot represent other types of vehicles such as private cars so that it will cause data deviation. Many other researches are monitoring the road network by using different kinds of data [10–13]. Although these studies have achieved good results, most of them still suffered from limited coverage, data sparsity, or data deviation.

Nowadays, smart mobile phones provide a great convenience for people's daily life, and mobile phone penetration is in excess of 112.23 percent in China. Thousands of cell towers are dis-

[*] Corresponding author.
E-mail address: yuanbox15@mails.jlu.edu.cn (Y. Xu).

https://doi.org/10.1016/j.neucom.2020.07.150
0925-2312/© 2021 Elsevier B.V. All rights reserved.

tributed in the urban area, which are used for providing communication and internet services to mobile phone users. When mobile phone users make or receive calls, send or receive messages, surf the internet, even move across the coverage of different cell towers, the signaling records which contain users' spatiotemporal information will generate passively. So, we can obtain large scale mobile signaling data, which contains a great value for many fields, especially in urban traffic.

Signaling data has many advantages, such as huge-scale users, wide-coverage, and a large amount of data. However, there are still many challenges in the effective use of signaling data. 1) A lot of invalid data and inherent defects exist in signaling data, such as ping-pong effect and data drift; 2) There is no fixed sampling frequency, where data missing often occurs due to poor signal or other reasons, especially in vehicle travel mode; 3) The cell tower covers such a wide area that it cannot accurately know the specific location of each user. Table 1 highlights the main differences between different data sources using in traffic monitoring systems.

In this paper, an urban traffic monitoring scheme, which named Urban-STM, is established by using large-scale signaling data. We first build an effective data preprocessing module to complete data cleaning, correction, and extraction. Then we integrate the historical GPS trajectories of 2,000 taxicabs in the interpolation module to handle the data sparsity and data missing problem. Afterward, we present an improved Hidden Markov Model (HMM) map matching algorithm, which can infer user's real trajectory on the road-network according to the signaling data. Finally, the traffic speed estimation module is applied to count the average speed of each road. In summary, our contributions are as follows:

- We present the Urban-STM scheme, which can achieve citywide road-network traffic monitoring by using large scale mobile signaling data;
- We present practical preprocessing and interpolation module to make the large scale and coarse-grained signaling data more suitable for analysis;
- We present an improved HMM map matching algorithm, which take the spatiotemporal characteristics of historical trajectories data into account to improve the accuracy of the model;
- We apply the Urban-STM scheme in Changchun city as a case study and present an extensive validation study based on 2,000 taxicabs' GPS trajectories.

The remainder of this paper is organized as follows: We give a discussion of the related work in Section 2. Then present some definitions and provide an overview of the Urban-STM in Section 3. Section 4 introduces each module of the Urban-STM scheme in detail successively. Section 5 shows the experimental results. Finally, we give a conclusion in Section 6.

2. Related work

It has attracted lots of attention to achieve a traffic monitoring system which can provide real-time road traffic conditions information. On the one hand, it can help to plan routes for urban commuters to avoid congestion, and on the other hand, it can provide data support for effective management and planning for the city. The methods of traffic monitoring are mainly divided into the following categories.

The first category is traditional approaches which rely on the devices installed on the roads to gather information about road conditions and usage. For example, [1] measured the traffic conditions directly through inductive loop detectors embedded in road segments, [3] inferred the traffic flow and speed by cameras placed at road intersections. Due to the high cost of equipment and installation, such approaches can hardly obtain pervasive coverage. To handle this problem, [14] proposed a network tomography approach which can measure vehicle traveling times by exploiting just a limited number of cameras placed at road intersections. However, equipment failures and poor signals can lead to severe data missing and have a significant impact on traffic monitoring results.

The second, vehicle GPS data can be easily obtained because most taxicabs and buses are equipped with GPS equipment, so there are a lot of researches on traffic monitoring [6,10], traffic congestion identification [15] and traffic prediction [16–19] based on vehicle GPS data. There is a problem that these methods should match the GPS points to the road-network first. For high sampling frequency GPS data, it can be easily matched because the dense GPS trajectory points could clearly identify the matching road sections by distance. However, most vehicles have a low sampling frequency, the sampling interval maybe 30 s to 1 min even more, that makes map-matching for GPS data more difficult. About this, some incremental algorithm and global algorithm based on the range of GPS were proposed to solve this problem and have good performance [20–22]. Although the vehicle trajectory data can reflect the traffic conditions well, there is still a disadvantage that the trajectories of taxicabs and buses have their own characteristics and cannot represent other types of vehicles, so it will cause data deviation to some extent.

The other traffic monitoring methods mainly based on check-in data [23] or crowd sensing method [12]. These methods use different data to implement road traffic monitoring from a novel perspective, however, they still suffered limited coverage, data sparsity or data deviation.

In this paper, we achieved a traffic monitoring scheme named Urban-STM by using mobile phone signaling data. There are some other researches about mobile phone signaling data, such as [24–29], both of them were focus on map-matching for signaling data. Different with these methods, Urban-STM integrated historical GPS trajectories of 2,000 taxicabs to calculate the probability matrix of HMM, and the experimental results show that Urban-STM improves traffic monitoring performance in terms of coverage and accuracy.

3. Definitions and overview

3.1. Definitions

Definition 1 (Road-network): Road-network is defined as a directed graph $G(V,E)$, where V is a set of nodes representing the

Table 1
Comparison between different data sources in traffic monitoring systems.

Date source	Sampling frequency	Localization error	Coverage	Data deviation
Sensors	High	Low	Low	High
Vehicle GPS	High	Low	Low	High
Cellphone app	High	Low	Low	High
Signaling data	Low	High	High	low

crossings of the road segments, and E is a set of edges representing road segments.

Definition 2 (Signaling Data): When the mobile phone communicates with a cell tower to obtain services, a mobile signaling record, which contains user ID, cell tower ID, the latitude and longitude of cell tower, and the timestamp, will generate passively. In this paper, $s_{i,j} = \{uid_j, cid_j, lat_j, lon_j, t_j\}$ represents the j-th record of user i, $S_i = \{s_{i,1}, s_{i,2}, \ldots, s_{i,j}\}$ represents the signaling record collection of user i, $S = \{S_1, S_2, \ldots, S_N\}$ represents signaling data of all users.

Definition 3 (Sparse Observation Trajectory): In the data preprocessing module, we filter and correct the original signaling data. Due to the sparsity of the original signaling data, we get the sparse signaling sequence, which we call the sparse observation trajectory. We use τs to represent the sparse observation trajectories of all users, τs_i represents the sparse trajectory of user i, $\tau s_{i,j}$ represents the j-th observation point of user i.

Definition 4 (Dense Observation Trajectory): In order to monitor the road network operation more accurately, we interpolate the sparse observation trajectories to get the dense trajectories. Similarly, τd represents the dense observation trajectories of all users, τd_i represents the sparse trajectory of user i, $\tau d_{i,j}$ represents the j-th observation point of user i.

Definition 5 (Hidden State Sequence): Each point in observation trajectory corresponds to a road segment, which is called hidden state. In the map-matching module training step, we convert the dense observation trajectories into hidden states sequences by the Viterbi decoding algorithm. We use τh to represent hidden states sequences of all users.

Definition 6 (Real Trajectory): Each real trajectory is the real driving track in the road network composed of continuous road sections and time. After completing the training of map-matching model, we finally obtain the real trajectories of all users represented by τr.

3.2. Overview

The real-time travel speed on a road segment is a basic index for monitoring road traffic status. In this study, we propose to utilize the mobile signaling data to capture the citywide traffic situation on the road network. The framework of our proposed Urban-STM is shown in Fig. 1. Using a large scale of mobile signaling data, the ultimate goal of this model is to estimate the traffic speed on each road segment on a citywide scale. Specifically, the model is comprised of four modules, including data preprocessing, interpolation, map-matching, and traffic speed estimation, each of which is described below.

In data preprocessing module, we filter and correct the original signaling data through the invalid data filtering sub-module and the data correction sub-module respectively. Then specially, we establish trajectory pattern recognition sub-module based on the decision tree method, which is used to extract vehicle trajectories from various travel modes. Through data preprocessing module, the original signaling data is converted into sparse observation trajectories.

Then in interpolation module, we integrate historical GPS trajectories of 2000 taxicabs to interpolate the sparse observation trajectories, so as to transform the sparse observation trajectories into dense observation trajectories.

Next, in map-matching module, we build an improved HMM, and use Viterbi algorithm to decode dense observation trajectories into hidden state sequences. We use a signaling data collection through which we can know the real trajectory of each user for adjusting the model parameters. After that, we can convert the sparse observation trajectories into real trajectories.

Finally, the traffic speed estimation module is used to calculate the average speed of each road section. Based on the introduction of above definitions and overview, our problem can be formulated as follows:

Given a road network G and a mobile signaling trajectories dataset S, then the real trajectory dataset τr in G which lead to the observation in S can be found. Finally, we estimate the road-network traffic average speed based on τr. Each module of Urban-STM is explained in detail in next section.

4. Methods

4.1. Data preprocessing

In reality, the original signaling data often have some defects due to equipment failure, system failure, bad signal and inherent defects. To make the data more suitable for our analysis, we designed the preprocessing steps as described in the following subsections.

4.1.1. Data filtering

In this sub-module, we first delete the invalid data such as fields missing data and duplicate data, then preliminary screen the data according to the following indicators:

$$n = |S_i| \tag{1}$$

$$d_{se} = \mathrm{Dis}(s_{i,0}, s_{i,n}) \tag{2}$$

$$\bar{d} = \frac{1}{n-1}\sum_{j=0}^{n-1}\mathrm{Dis}(s_{i,j}, s_{i,j+1}) \tag{3}$$

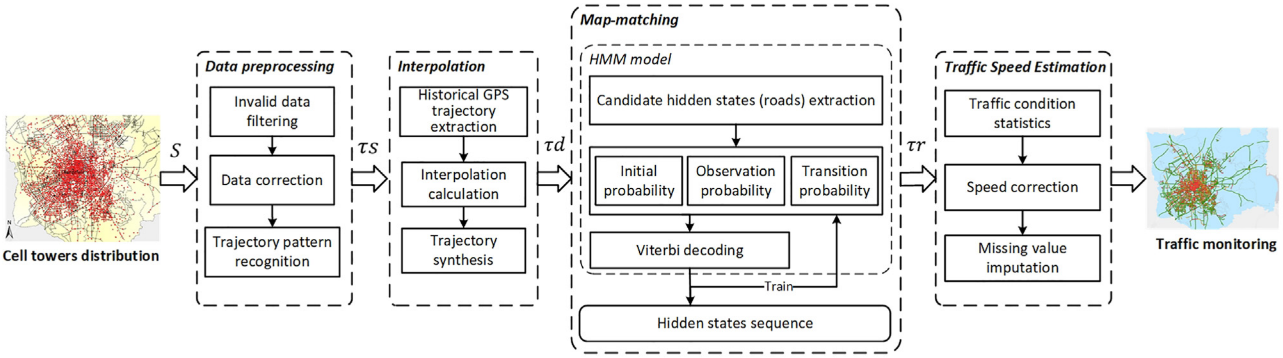


Fig. 1. The Urban-STM scheme architecture.

where $|S_i|$ is the number of signaling records of user i, $\mathrm{Dis}(a, b)$ is the straight-line distance function between a and b, so d_{se} is the distance between the starting record and the ending record of S_i , $\overline{d}$ is the average distance between two consecutive trajectory points of S_i. Here, we set corresponding thresholds (n > 10, d_{se}>100 m, $\overline{d}$<500 m) for these three indicators for preliminary screening.

4.1.2. Data correction

There are two common phenomena in the signaling data which have serious impact on our results, that is, ping-pong effect and data drift.

In the mobile communication system, if the signal strength of two cell towers changes dramatically in a certain area or the coverage area overlaps, the connection will switch back and forth between the two cell towers, producing the so-called ping-pong effect.

For an observation trajectory of user i, if the latitude and longitude of $s_{i,j}$ are equal to those of $s_{i,j+m}$, and $t_{j+m} - t_j < \Delta t$, that will be treated as a ping-pong switching, then $s_{i,j+1}$ to $s_{i,j+m}$ will be deleted.

Data drift denotes the scenarios when the observed location is far away from the actual location, perhaps due to the signal reflection or refraction. For an observed signaling record $s_{i,j}$, if $\mathrm{Dis}(s_{i,j-1}, s_{i,j}) > V * (t_j - t_{j-1})$ and $\mathrm{Dis}(s_{i,j}, s_{i,j+1})$ < $V^*(t_{j+1} - t_j)$, where V is the speed threshold. then $s_{i,j}$ will be treated as a data drift record and will be deleted.

4.1.3. Trajectory pattern recognition

In the city, people can choose a variety of travel modes, such as walking, riding, bus, vehicle (taxicabs or private cars), subway, etc. Among various travel modes, vehicle travel mode can directly reflect the operation of the road network, so we try to extract vehicle trajectories from signaling data. In this paper, the decision tree is used for trajectory pattern classification. Except the three indicators mentioned in data preprocessing sub-module, we extract the following additional features:

$$l = \sum_{j=0}^{n-1} \mathrm{Dis}(s_{i,j}, s_{i,j+1}) \tag{4}$$

$$\overline{v}_{se} = \frac{l}{t_n - t_0} \tag{5}$$

$$\overline{v} = \frac{\mathrm{Dis}(s_{i,j}, s_{i,j+1})}{t_{j+1} - t_j} \tag{6}$$

$$s_v^2 = \frac{1}{n-1} \sum_{j=0}^{n-1} \left(\frac{\mathrm{Dis}(s_{i,j}, s_{i,j+1})}{t_{j+1} - t_j} - \overline{v} \right)^2 \tag{7}$$

where l is the sum of the Euclidean distance between every two adjacent records of $S_i, \overline{v}_{se}$ is the average speed of S_i from starting record to the ending record, $\overline{v}$ is the average value of the speed between every two adjacent records in S_i, s_v^2 is the variance of speed between adjacent records. In addition, we use time (0–23), date (0 for weekdays and 1 for weekends) and weather as extra features. We take the information gain ratio as the criterion of feature selection to construct the decision tree model.

Entropy is defined as follows:

$$\mathrm{H}(S) = -\sum_{k=1}^{M} p_k \log_2 p_k \tag{8}$$

where p_k is the probability of selecting the classification k, and there are M classes in total. We can calculate the information gain ratio by the following formula:

$$\mathrm{g_R}(S, a) = \frac{\mathrm{H}(S) - \mathrm{H}(S|a)}{\mathrm{H_X}(S)} \tag{9}$$

$$\mathrm{H}(S|a) = -\sum_{j=1}^{F} \left(\frac{|S^j|}{|S|} \mathrm{H}\left(S^j\right) \right) \tag{10}$$

$$\mathrm{H_X}(S) = -\sum_{j=1}^{F} \left(\frac{|S^j|}{|S|} \log_2 \left(\frac{|S^j|}{|S|} \right) \right) \tag{11}$$

For the feature a, suppose it can divide the dataset S into F subsets: $S^1, S^2, \ldots\ldots, S^F$, $\mathrm{g_R}(S, a)$ is the information gain ratio obtained by feature a. Finally, we choose the feature recursively according to the information gain ratio to build the trajectory pattern recognition tree model:

$$\vec{f_T} = \left(n, d_{se}, \overline{d}, l, \overline{v}_{se}, \overline{v}, s_v^2, time, date, weather\right) \tag{12}$$

4.2. Interpolation

Mobile signaling data does not have the fixed sampling frequency like the trajectories data collected by the vehicle GPS equipment. Theoretically, it will generate a record when a mobile phone user pass through the coverage of each cell tower, but in practice, the data is often sparse due to poor signal and other reasons, especially the vehicle trajectories because of the fast speed. There are two consequences caused by the sparse signaling data. On the one hand, it leads to the uncertainty of trajectory, that is, there will be multiple paths matching a sparse observation trajectory. On the other hand, it does not satisfy the requirement of using Hidden Markov Model for map matching. The simple linear or nonlinear interpolation cannot solve the first problem effectively, to overcome this, we integrate historical GPS trajectories of 2000 taxicabs to interpolate the sparse observation trajectories.

4.2.1. Historical GPS trajectories extraction

For a sparse observation trajectory τs_i, we calculate the distance between each two adjacent records, if $\mathrm{Dis}(\tau s_{i,j}, \tau s_{i,j+1})$ > λ_{insert}, interpolation is needed, and the number of interpolation points is $\lfloor \frac{\mathrm{Dis}(\tau s_{i,j}, \tau s_{i,j+1})}{\lambda_{insert}} \rfloor$, where $\lfloor * \rfloor$ denotes the integer part. Then we need to extract historical taxicabs GPS trajectories which are related to $\tau s_{i,j}$ and $\tau s_{i,j+1}$. In this paper, the related GPS trajectories is defined as the following two cases:

1. For a taxicab GPS trajectory p, if p passes the coverage of the two cell towers sequentially which corresponding to $\tau s_{i,j}$ and $\tau s_{i,j+1}$, then p will be treated as a related GPS trajectory of $\tau s_{i,j}$ and $\tau s_{i,j+1}$.
2. For two taxicab GPS trajectories p_1 and p_2, if p_1 passes the coverage of the cell tower which corresponding to $\tau s_{i,j}, p_2$ passes the coverage of the cell tower which corresponding to $\tau s_{i,j+1}$, and there are overlapping road sections in p_1 and p_2, then p_1 and p_2 will be treated as related GPS trajectories of $\tau s_{i,j}$ and $\tau s_{i,j+1}$.

4.2.2. Interpolation calculation

After the related GPS trajectories are extracted, we start to calculate the coordinates of the interpolation points. The rectangle rt with $\tau s_{i,j}$ and $\tau s_{i,j+1}$ as diagonal is the interpolation region, the length and width are ls and ss respectively, the number of interpolation points $n_{insert} = \lfloor \frac{ls}{\lambda_{insert}} \rfloor$.

Then we divide rt into n_{insert} sub-rectangles along ls, as shown in the solid rectangle in Fig. 2, and each sub-rectangle is represented by srt. Next, continue to divide srt into a grid of $\lfloor \frac{ls}{n_{insert} \cdot w} \rfloor$ rows and

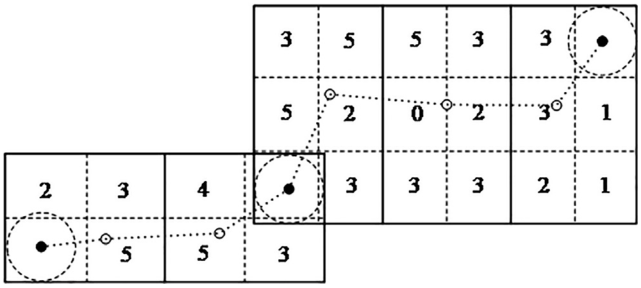


Fig. 2. Interpolation calculation. The black points are sparse observation points, the hollow points are interpolation points.

$\lfloor \frac{ss}{w} \rfloor$ columns as shown in the dotted rectangle in Fig. 2, where w is the expected side length of each sub-rectangles. Finally, the interpolation is calculated as follows:

$$ne_k = nf_k + \delta \cdot nt_k \tag{13}$$

$$point_{insert} = \frac{1}{\sum_{k=1}^{m} ne_k} \sum_{k=1}^{m} (ne_k \cdot point_k) \tag{14}$$

where ne_k is the number of related GPS trajectories in the k-th sub-rectangle of srt, nt_k and nf_k represent the number of related GPS trajectories in the same time slice and different time slices respectively. Considering that the historical related GPS trajectories of the same time slice of the day has greater reference, so nt_k is weighted by a factor δ, here we set δ = 1.5. Finally, the interpolation coordinate $point_{insert}$ is calculated as Eq. 14, where $point_k$ is the latitude and longitude of grid center.

4.3. Map-matching

Through preprocessing and interpolation module, we obtain a large-scale dense observation trajectories dataset τd. In order to monitor the road-network traffic condition, we need to accurately infer the real trajectories according to τd. The map-matching problem can be modeled as HMM, in which τd and $G(V, E)$ are regarded as input and the real trajectories dataset as output. Compared with the standard HMM algorithm, we combine the related taxicabs GPS trajectories (described in Interpolation section) to calculate the observation probability and transfer probability. The detail of map-matching is described in this section.

4.3.1. Candidate hidden states extraction

Hidden Markov Model (HMM) considers the problem as a Markov process, which can be used to infer hidden states sequence from observation states sequence. For our map matching problem, τd can be seen as observation states sequences and E in $G(V, E)$ can be seen as hidden states collection. However, it will cause huge amount of calculation if we use E as hidden states collection directly, because there are too many roads in E. So, we have to reduce the number of hidden state candidates. We extract the roads intersect with the coverage area for each cell tower, and build the index so that we can get hidden states collection quickly.

4.3.2. Build HMM

In order to infer the true trajectory sequence from the candidate set through HMM, we must obtain three key elements, they are observation probability matrix B, state transition probability matrix A and initial probability matrix π.

The observation probability matrix $B = \{b_{j,k}\}$, where $b_{j,k} = p(\tau d_{i,j} | h_k)$, and h_k is the kth hidden state in the candidate roads, $p(\tau d_{i,j} | h_k)$ is the probability of observing $\tau d_{i,j}$ when user i driving on road k. So, B is the probability matrix between the candidate hidden states and the observed states. The formula of each element in B is as follows:

$$b_{j,k} = \begin{cases} \frac{ne_k}{ne_j} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left(d_{j,k}-\mu\right)^2}{2\sigma^2}}, & \text{if} \quad d_{i,j} < \lambda_d, \\ 0, & \text{otherwise}. \end{cases} \tag{15}$$

where $d_{j,k}$ is the shortest distance between road h_k and $\tau d_{i,j}$, λ_d is the distance threshold here we set λ_d = 300 m as the initial value, ne_k here is the number of related taxicabs GPS trajectories pass through h_k, and ne_j is the number of related GPS trajectories pass through the coverage area of $\tau d_{i,j}$. We consider that the observation probability satisfies the Gauss distribution, here we set $\mu = 0, \sigma = 150$.

The state transition probability matrix $A = \{a_{j,k}\}$, where $a_{l,k}$ = $p(h_l \rightarrow h_k | \tau d_{i,j}, \tau d_{i,j+1})$, it refers to the probability of driving from h_l to h_k when the observation state changes from $\tau d_{i,j}$ to $\tau d_{i,j+1}$.

$$a_{l,k} = \begin{cases} \lambda_a, & \text{if} \quad h_l = h_k, \\ (1-\lambda_a) \cdot \frac{|ne_l \rightarrow ne_k|}{ne_j + ne_{j+1}}, & \text{if} \quad h_l \text{ connected } h_k, \\ 0, & \text{otherwise}. \end{cases} \tag{16}$$

where λ_a is a parameter, a bigger λ_a means people tend to drive along the same road. When $h_l \neq h_k$ and they are directly connected, the transition probability is $(1-\lambda_a) \cdot \frac{|ne_l \rightarrow ne_k|}{ne_j + ne_{j+1}}$, $|ne_l \rightarrow ne_k|$ is the number of related GPS trajectories which driven from h_l to h_k. ne_j and ne_{j+1} are the number of related GPS trajectories pass through the coverage area of τd_{ij} and $\tau d_{i,j+1}$ respectively. And when $h_l \neq h_k$ and they are not directly connected, the transition probability is 0.

The initial probability matrix $\pi = \{b_{0,k}\}$, here we simply use the observation probability of the first observation state $\tau d_{i,0}$ as the initial probability. Since the time factor is taken into account during the extraction of historical trajectories, so the spatiotemporal characteristics of historical trajectories data is introduced into our map matching model, which can improve the accuracy of the model.

4.3.3. Viterbi algorithm decoding

When the observation states sequence τd_i, hidden states collection H_i, observation probability matrix B, state transition probability matrix A and initial probability matrix π are known, the hidden states sequence with maximum probability can be decoded by Viterbi decoding algorithm. Here we show a simple example by Fig. 3.

As shown in Fig. 3(a), we can see a car driving from h1 to h8, and an observation trajectory from cell tower 1 to 4. So, the HMM can be built, observation states sequence τd_i=$\tau d_{i,1}, \tau d_{i,2}, \tau d_{i,3}, \tau d_{i,4}$, hidden states collection H_i = $\{h1, h2, h3, h4, h5, h6, h7, h8\}$, and B, A, π can be built as the details in Build HMM subsection. The details of Viterbi decoding process shown in Fig. 3(b). Each row represents a hidden state in H_i, the hollow circles represent the observation probability equal to 0. For the j-th observation state $\tau d_{i,j}$, the path with the greatest probability from $\tau d_{i,0}$ to $\tau d_{i,j-1}$ can be found, so we just need to consider the probability of each path from $\tau d_{i,j-1}$ to $\tau d_{i,j}$, there is no need to traverse all paths. Finally, we get the most probable result, the red path in Fig. 3(b).

4.4. Traffic speed estimation

Through previous modules, we can obtain large scale vehicle real trajectories τr in each time slice. In this section, we calculate the average speed of each road according to τr.

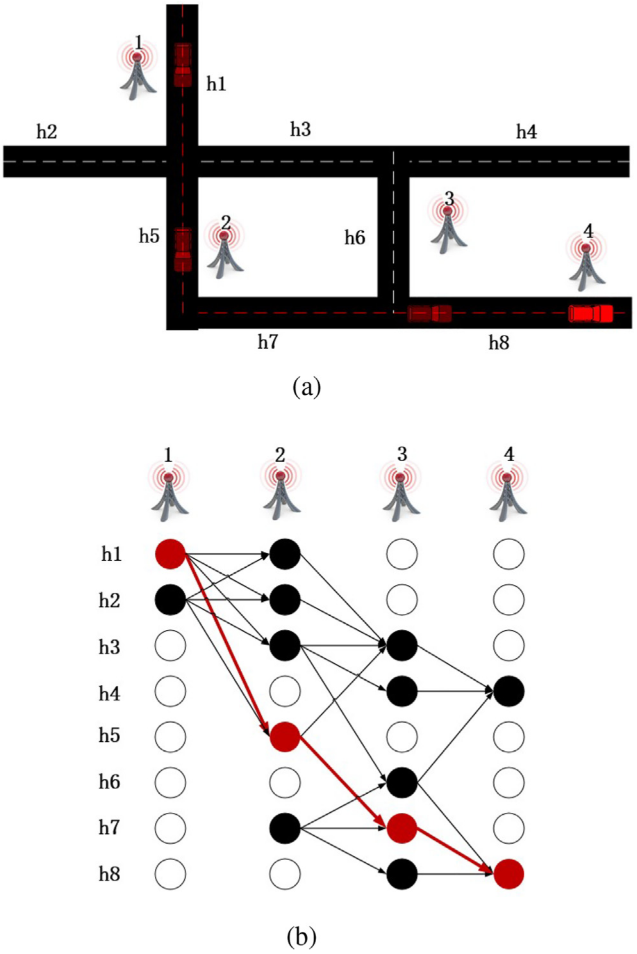


Fig. 3. Map-matching demonstration. (a) A trajectory on the road network; (b) Viterbi decoding process.

4.4.1. Speed statistic

For a real trajectory τr_i, only sparse trajectory records have time attribute, assume $\tau r_{i,j}$ and $\tau r_{i,j+q}$ have time attribute, the speed of road $\tau r_{i,j}$ to $\tau r_{i,j+q}$ is $\frac{\sum_{j=j}^{j+q} l(\tau r_{i,j})}{t_{j+q}-t_j}$, where $l(\tau r_{i,j})$ is the length of road $\tau r_{i,j}$, the average speed of all the roads are calculated as formulate 17.

$$E\_speed = \sum_{i=0}^{N} \text{speed}(\tau r_i) \o \phi\left(\sum_{i=0}^{N} \text{volume}(\tau r_i)\right) \tag{17}$$

Both $\text{speed}(\tau r_i)$ and $\text{volume}(\tau r_i)$ are vectors of length $|E|$ (the number of all roads in road network), they are speed and volume we get from τr_i and most of elements not in τr_i is 0. The function $\phi(*)$ used to change the element whose vector is 0 to 1. The symbol ø means dividing corresponding elements of two equal length vectors. *E_speed* is the average speed of all roads.

4.4.2. Speed correction

There is a problem that even though the signaling data can be converted into real trajectories accurately, we still do not know the precise location of each signaling record in time t. So, there is a certain amount of error in *E_speed*. To handle this, we collect a ground truth dataset that each signaling trajectory has the high sample rate GPS points, then we calculate the average speed of

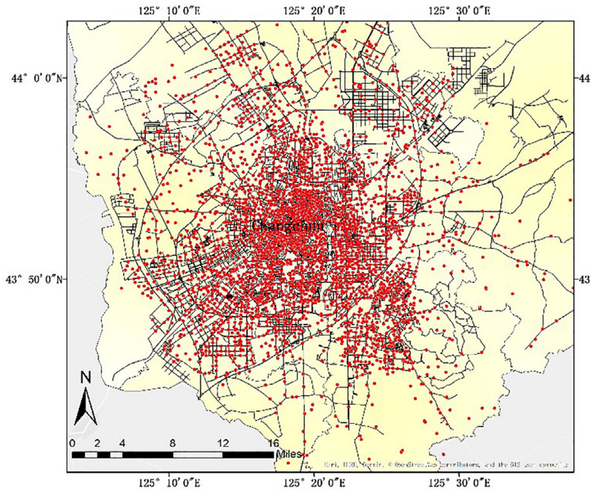


Fig. 4. The distribution of cell towers in Changchun city. The red points represent cell towers and the black lines represent road-network.

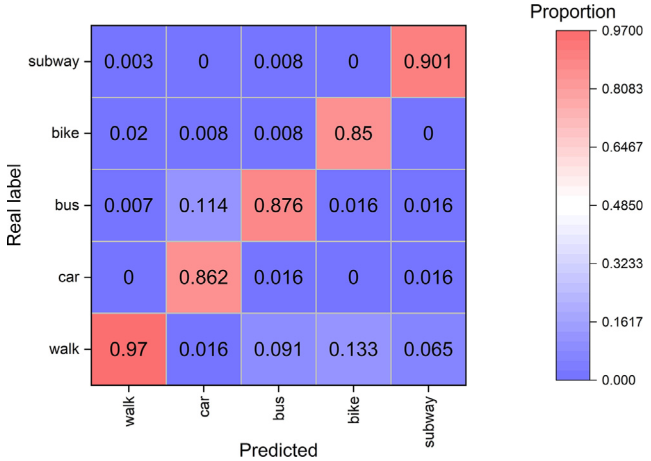


Fig. 5. The confusion matrix of trajectory pattern recognition.

the same road at the same time slice, then establish a linear regression model to correct the speed deviation.

4.4.3. Missing value imputation

Anyway, signaling data cannot cover all the roads at all the time, there will always be some missing values especially at night. We use the average speed of adjacent roads or time series to fill in the missing data.

5. Experiments

5.1. Datesets

The large scale anonymous signaling data was provided by Jilin Branch of China Unicom. There are 21,098 roads and 6337 cell towers distributed in Changchun city as shown in Fig. 4. There are about 2 million mobile users in our dataset, and nearly 60 billion signaling records will generate in one day. In addition, we integrated the historical GPS trajectories of 2000 taxicabs in one month, which can be used to interpolation, build HMM and correct the speed deviation. Besides, for ground truth, we collected 2,000 signaling trajectories labeled with travel type and high sampling

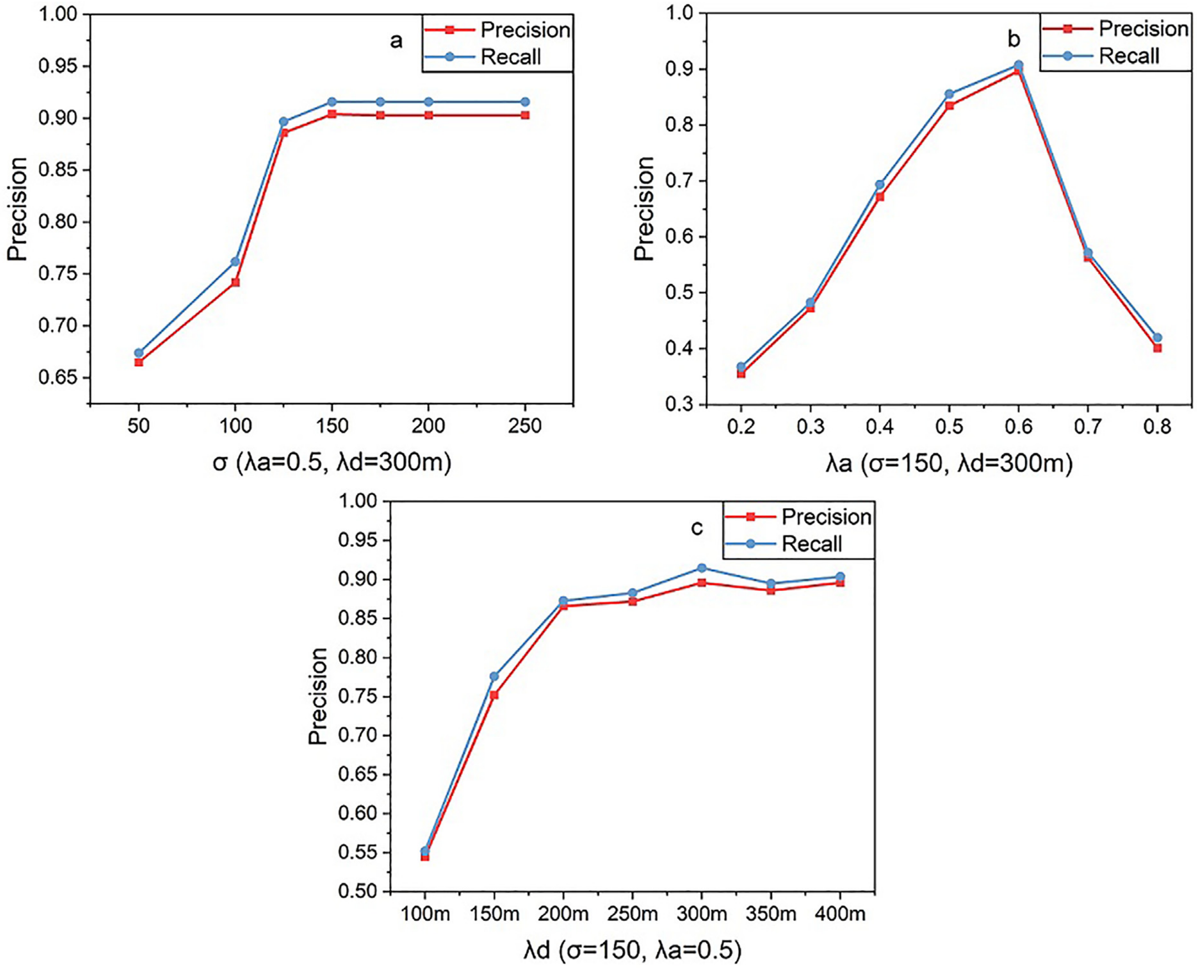


Fig. 6. The effects of changing parameters.

rate GPS trajectory points. We use 15 min as a time slice to monitor the average speed of the road network.

5.2. Map-matching baselines

To evaluate the accuracy of map-matching, we compare the map-matching results with the following baseline methods:

- SnapNet system [24]: A real-time map matching for challenging environments, which adopted a series of consecutive filter modules and a linear interpolation module to preprocess the trajectory data, then applied an incremental HMM to compute the maximum likelihood sequence of hidden states.
- A standard HMM map-matching model [30]: A standard HMM map-matching technique proposed by Microsoft, which used the standard normal distribution and exponential probability distribution to build the observation probability matrix and transition probability matrix, respectively.
- Shortest path algorithm: In general, people prefer to take shorter routes between origin and destination, so we use this method as a baseline.

We use precision and recall as evaluation metrics to measure the efficiency of these models (detail in Section 5.3.2).

5.3. Results analysis

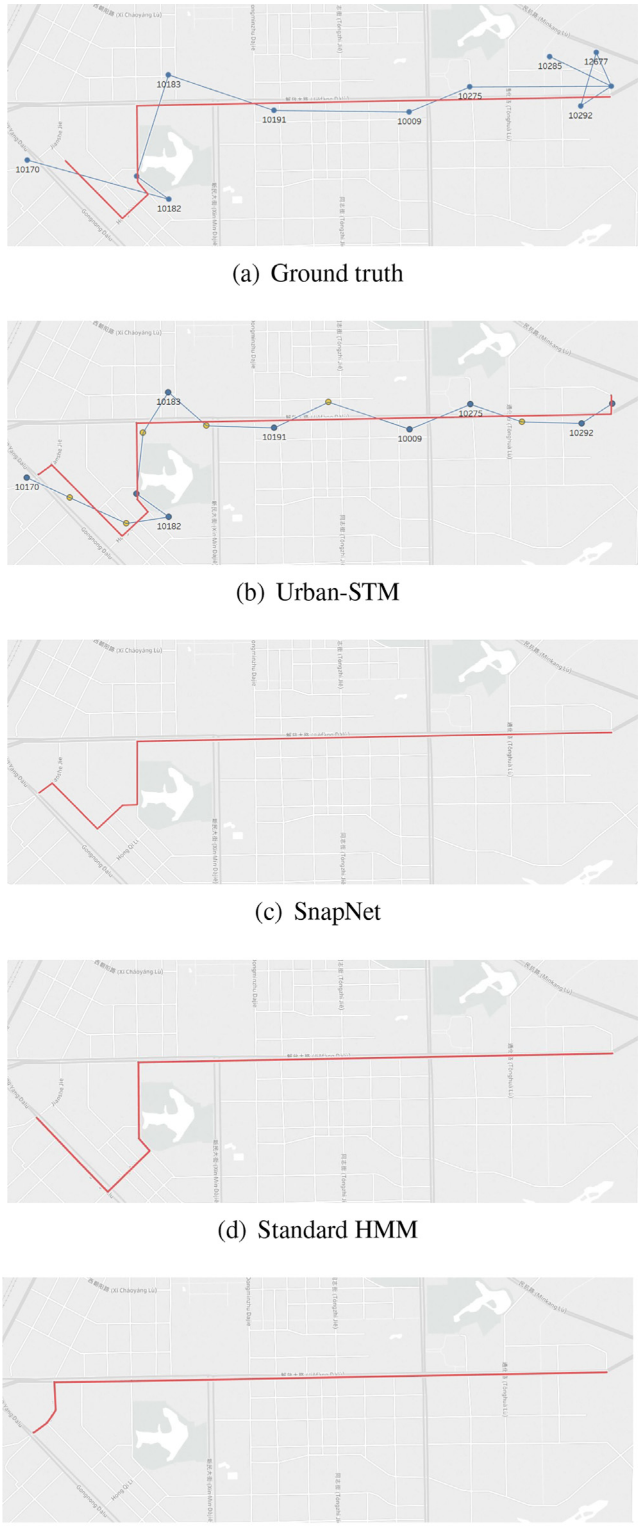
We analyze the effectiveness of each module of urban-STM in this subsection. We first verify the validity of trajectory pattern recognition, then quantify the accuracy and recall of different HMM parameters to find the best combination of parameters. Afterward, we compare our map matching results with baselines as mentioned before. Finally, we show the traffic monitoring results, and compare with the results based on taxicabs GPS trajectories.

5.3.1. Trajectory pattern recognition

There are 5 types of trajectories in the ground truth dataset, i.e. walk, car, bus, bike and subway. We use cross-validation method to verify the effective of the trajectory pattern recognition model. The result is shown in Fig. 5, the horizontal axis is the predicted value and the vertical axis is the true value. We can see the values in diagonal are significantly higher than others, it demonstrates the classification of each travel mode is effective. In Urban-STM, we only use car trajectories to monitor the whole road-network.

5.3.2. Map-matching

In this subsection, we use two evaluation metrics, precision and recall, to evaluate our map-matching model and other baselines. The precision and recall are formulated as follows:



(a) Ground truth

(b) Urban-STM

(c) SnapNet

(d) Standard HMM

(e) Shortest path

Fig. 7. The map-matching results of different algorithms. (a) The blue dots are origin observation signaling records, the red line is the true path; (b) The red line is the matching result of Urban-STM, the blue points are sparse observation points and the yellow dots are the interpolation points; (c) The matching result of SnapNet; (d) The matching result of standard HMM; (e) The matching result of the shortest path.

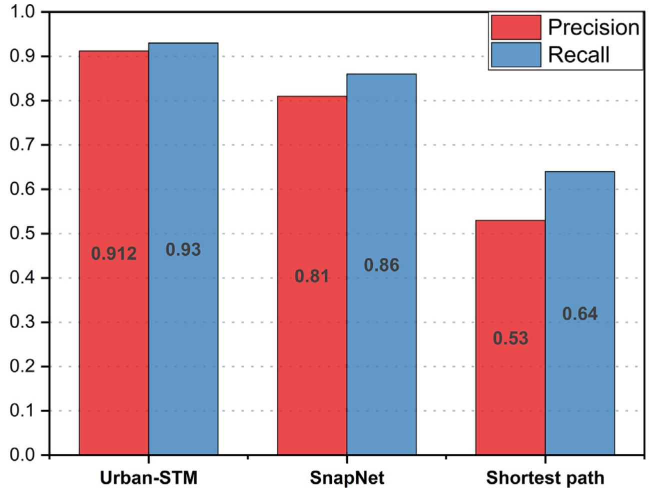


Fig. 8. The precision and recall of Urban-STM and baselines.

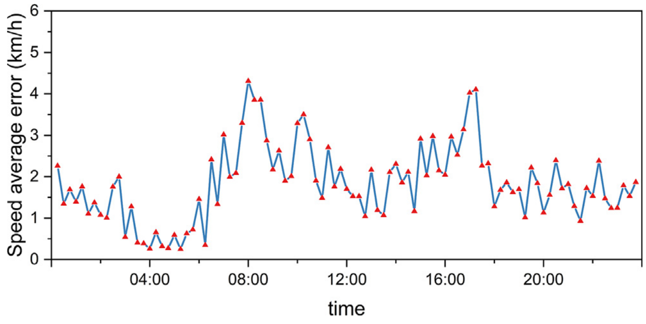


Fig. 9. The average error of speed in each time slice.

$$Precision = \frac{\text{true}(\tau r)}{|\tau r|} \quad (18)$$

$$Recall = \frac{\text{true}(\tau r)}{|\tau r'|} \quad (19)$$

where τr is the map-matching result, $\tau r'$ is the ground-truth trajectory based on high sampling rate GPS points. $\text{true}(\tau r)$ is a function returns the number of correctly matched roads in τr. $|\tau r|$ and $|\tau r'|$ means the total number of roads in τr and $\tau r'$.

Next, we set the default value for the map-matching model (σ = 200, λ_a = 0.5, λ_d = 300 m), then use these two indicators to test the performance of the model under different values of σ, λ_a and λ_d. The results are shown in Fig. 6.Fig. 6a shows that, under the default value of λ_a and λ_d, the precision and recall increase with increasing σ, until σ = 150 it reaches its maximum value and remains stable. Fig. 6b shows the effect of changing λ_a from 0.2 to 0.8. The performance of the model significant increases from 0.2 to 0.6, then decreases dramatically. In Fig. 6c, it is obviously that the model performs best when λ_d = 300 m. In addition, it can be seen that the precision of the model is slightly lower than recall. This is due to the large coverage of the cell tower which makes more roads in map-matching results than the ground truth trajectory. Finally, we use the optimal parameters that σ = 150, λ_a = 0.6, λ_d = 350 m, which can achieve the best performance of the map matching model.

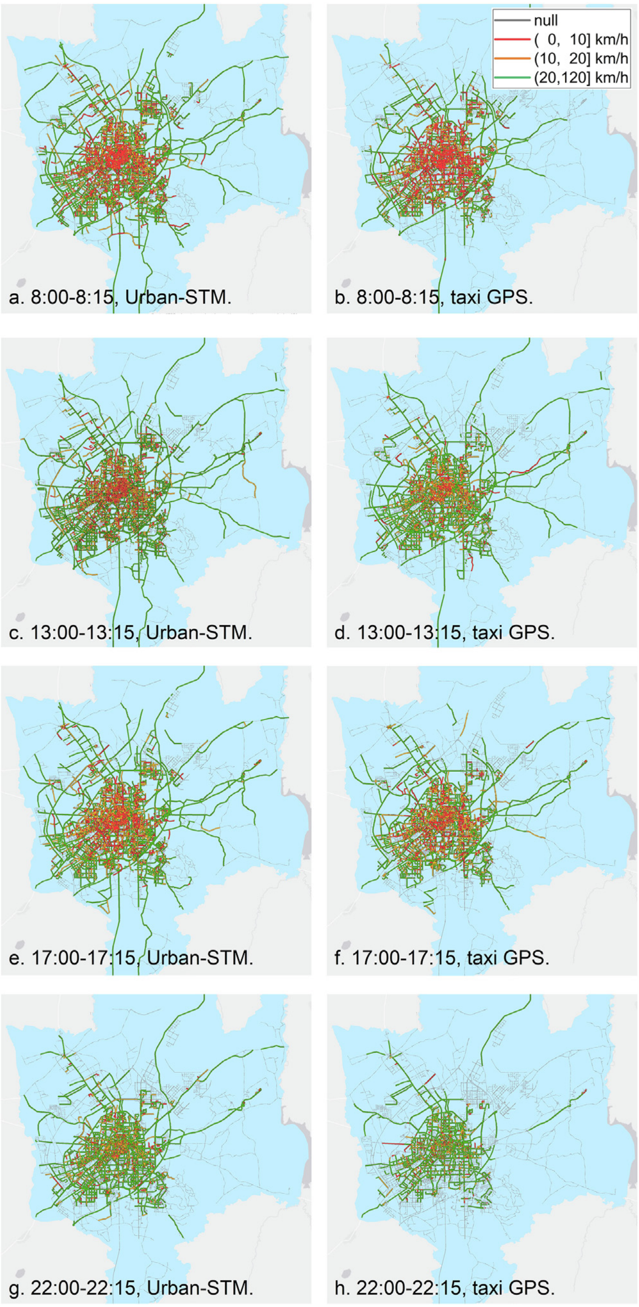

null
(0, 10] km/h
(10, 20] km/h
(20,120] km/h
a. 8:00-8:15, Urban-STM.
b. 8:00-8:15, taxi GPS.
c. 13:00-13:15, Urban-STM.
d. 13:00-13:15, taxi GPS.
e. 17:00-17:15, Urban-STM.
f. 17:00-17:15, taxi GPS.
g. 22:00-22:15, Urban-STM.
h. 22:00-22:15, taxi GPS.

Fig. 10. The monitoring results of road network average speed in different time slice.

After that, we compare our map-matching model with three baselines we mentioned before. We first show a demonstration of the matching results of different methods, then use ground truth dataset to measure the precision and recall of these algorithms. It can be observed in Fig. 7 that the performance of our model has the highest accuracy, and Fig. 8 shows that our map-matching method has the highest precision and recall.

Compare with baseline methods, our preprocessing and interpolation method are effective and can make the original signaling data more suitable for map-matching. Besides, we introduce the spatiotemporal characteristics of historical trajectories data into the interpolation and HMM construction process, which is the key point of improving the accuracy of our model.

5.3.3. Traffic monitoring

To verify the accuracy of road network average speed monitoring, we select 20 main roads to calculate the average speed of each road at each time slice by using taxicabs GPS and our method respectively, then the mean absolute error of each time slice is calculated as follows:

$$error_t = \frac{1}{N}\sum_{i=0}^{N}\left(\left|E\_speed_{i,t} - E\_speed\prime_{i,t}\right|\right) \tag{20}$$

where $E\_speed_{i,t}$ is the average speed of road i calculated by our method in time slice t, and $E\_speed\prime_{i,t}$ is the average speed of road i calculated by using taxicabs GPS data. The error in each time slice is shown in Fig. 9, we can see the errors are higher at 8:00 (morning peak) and 17:00 (evening peak) around, which are no more than 4.5 km/h, the errors in other time slices are no more than 3 km/h.

Finally, we show our traffic average speed monitoring results (before missing value imputation) in Fig. 10. We choose four different time slices and compare with the monitoring results calculated by using taxicabs GPS data. We can see that Urban-STM covers significantly more roads than the monitoring results of taxicabs GPS data in all of the aforementioned time slices. In the time slice 8:00–8:15, Urban-STM covers 87.2% roads, taxicabs GPS data only covers 66.7% roads. We can also find different traffic condition in different time slices. We divide the traffic condition into three levels according to the average speed as shown in the legend of Fig. 10. It can be seen that the number of roads with slow speed are much higher in morning peek and evening peak.

6. Conclusion

In this paper, the Urban-STM scheme was proposed to achieve citywide road-network traffic monitoring by using large-scale mobile signaling data. Experimental results show that each module of the scheme is effective and accurate. And Figs. 9 and 10 can prove that the Urban-STM has high accuracy and larger coverage in traffic monitoring.

Compare with other schemes, the effective usage of large-scale coarse-grained mobile phone signaling data makes Urban-STM have the following advantages: First, a large number of users and the extensive deployment of cell towers provide massive signaling data, compared to the schemes using the data collected from sensors deployment on the limit roads segments, Urban-STM achieves a wider range of roads traffic monitoring services in citywide; Second, compare to the common traffic monitoring schemes which using GPS trajectories data collected from taxicabs, buses or some cellphone apps, multiple traffic modes (i.e. walk, car, bus, bike, and subway) can be extracted through trajectory pattern recognition module in Urban-STM, so that we can obtain more valuable traffic information and handle the problem of data deviation; Last but not least, the signaling data is obtained in passively way, with low energy consumption.

Also, the performance of our scheme could be improved by overcoming the following limitations: The coarse-grained signaling data requires special and complex data preprocessing, and due to the large coverage area of a cell tower, it is difficult to confirm the start and end road segments of the trajectory, which is the main cause of the errors; Besides, the large-scale of signaling data requires more storage space and computing resources, in this paper, we implement Urban-STM by using a Hadoop platform built by 3 servers (with 32 GB RAM, 1.80 GHz core i5 processor and 7 TB hard disk storage); In addition, the effectiveness of Urban-STM depends on the density of the cell towers distribution, which has higher accuracy in developed cities especially in the urban center, but it does not apply to suburban areas or rural areas.

With the development of 5G technology in the future, the quality of mobile phone signaling data will be improved, so that it will be more valuable for many other research fields, such as traffic congestion identification and prediction, human mobility, recommendation system, etc.

CRediT authorship contribution statement

Qiuyang Huang: Conceptualization, Methodology, Software, Writing - original draft. **Yongjian Yang:** Supervision, Resources, Funding acquisition. **Yuanbo Xu:** Writing - review & editing, Project administration. **Funing Yang:** Investigation, Data curation. **Zhilu Yuan:** Software, Validation. **Yongxiong Sun:** Investigation, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundations of China under Grant No. 61772230, No. 61972450, No. 61976102, Natural Science Foundation of China for Young Scholars No. 61702215 and No. 62002132, No. 71901147, China Postdoctoral Science Foundation No. 2020M681040, No. 2017M611322 and No. 2018T110247, Natural Science Foundation of Jiin Provincial Science and Technology Department No. 20180101332JC, No. 20190201102JC and No. 20190201022JC, Changchun Science and Technology Development Project No.18DY005, National Defense Science and Technology Key Laboratory Fund Project No. 61421010418, Open Fund of Key Laboratory of Urban Land Resources Monitoring and Simulation, No. KF-2019-04-014, and China National Postdoctoral Program for Innovative Talents No. BX20190140.

References

[1] S.S.M. Ali, B. George, L. Vanajakshi, Mutually coupled multiple inductive loop system suitable for heterogeneous traffic, IET Intell. Transp. Syst. 8 (5) (2014) 470–478.
[2] R. Du, P. Santi, M. Xiao, A.V. Vasilakos, C. Fischione, The sensable city: a survey on the deployment and management for smart city monitoring, IEEE Commun. Surveys Tutorials 21 (2) (2019) 1533–1560.
[3] S. Chaudhary, S. Indu, S. Chaudhury, Video-based road traffic monitoring and prediction using dynamic bayesian networks, IET Intell. Transp. Syst. 12 (3) (2018) 169–176.
[4] P. Barsocchi, P. Cassara, F. Mavilia, D. Pellegrini, Sensing a city's state of health: structural monitoring system by internet-of-things wireless sensing devices, IEEE Consumer Electron. Mag. 7 (2) (2018) 22–31.
[5] P. Szymański, M. Żołnieruk, P. Oleszczyk, I. Gisterek, T. Kajdanowicz, Spatio-temporal profiling of public transport delays based on large-scale vehicle positioning data from gps in wrocław, IEEE Trans. Intell. Transp. Syst. 19 (11) (2018) 3652–3661.
[6] W. Shi, Y. Liu, Real-time urban traffic monitoring with global positioning system-equipped vehicles, IET Intell. Transp. Syst. 4 (2) (2010) 113–120.
[7] S. Hadavi, S. Verlinde, W. Verbeke, C. Macharis, T. Guns, Monitoring urban-freight transport based on gps trajectories of heavy-goods vehicles, IEEE Trans. Intell. Transp. Syst. 20 (10) (2019) 3747–3758.
[8] A. Abadi, T. Rajabioun, P.A. Ioannou, Traffic flow prediction for road transportation networks with limited traffic data, IEEE Trans. Intell. Transp. Syst. 16 (2) (2015) 653–662.
[9] Qiuyang Huang, Yongjian Yang, Zhilu Yuan, Hongfei Jia, Liping Huang, Zhanwei Du, The temporal geographically-explicit network of public transport in Changchun city, Northeast China, Sci. Data 6 (1) (2019) 190026.
[10] R. Du, C. Chen, B. Yang, N. Lu, X. Guan, X. Shen, Effective urban traffic monitoring by vehicular sensor networks, IEEE Trans. Veh. Technol. 64 (1) (2015) 273–286.
[11] K. Zheng, E. Yao, J. Zhang, Y. Zhang, Traffic flow estimation on the expressway network using toll ticket data, IET Intell. Transp. Syst. 13 (5) (2019) 886–895.
[12] D. Cerotti, S. Distefano, G. Merlino, A. Puliafito, A crowd-cooperative approach for intelligent transportation systems, IEEE Trans. Intell. Transp. Syst. 18 (6) (2017) 1529–1539.
[13] A. Lesani, L. Miranda-Moreno, Development and testing of a real-time wifi-bluetooth system for pedestrian network monitoring, classification, and data extrapolation, IEEE Trans. Intell. Transp. Syst. 20 (4) (2019) 1484–1496.
[14] R. Zhang, S. Newman, M. Ortolani, S. Silvestri, A network tomography approach for traffic monitoring in smart cities, IEEE Trans. Intell. Transp. Syst. 19 (7) (2018) 2268–2278.
[15] Yongjian Yang, Xu. Yuanbo, Jiayu Han, En Wang, Weitong Chen, Lin Yue, Efficient traffic congestion estimation using multiple spatio-temporal properties, Neurocomputing 267 (2017) 344–353.
[16] Bailin Yang, Shulin Sun, Jianyuan Li, Xianxuan Lin, Yan Tian, Traffic flow prediction using lstm with feature enhancement, Neurocomputing 332 (2019) 320–327.
[17] E. Necula, Dynamic traffic flow prediction based on gps data, Nov 2014..
[18] P. Rathore, D. Kumar, S. Rajasegarar, M. Palaniswami, J.C. Bezdek, A scalable framework for trajectory prediction, IEEE Trans. Intell. Transp. Syst. 20 (10) (2019) 3860–3874.
[19] K. Tang, S. Chen, Z. Liu, Citywide spatial-temporal travel time estimation using big and sparse trajectories, IEEE Trans. Intell. Transp. Syst. 19 (12) (2018) 4023–4034.
[20] T. Hunter, P. Abbeel, A. Bayen, The path inference filter: model-based low-latency map matching of probe vehicle data, IEEE Trans. Intell. Transp. Syst. 15 (2) (2014) 507–529.
[21] Y. Gong, E. Chen, X. Zhang, L.M. Ni, J. Zhang, Antmapper: an ant colony-based map matching approach for trajectory-based applications, IEEE Trans. Intell. Transp. Syst. 19 (2) (2018) 390–401.
[22] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, Yan Huang, Map-matching for low-sampling-rate gps trajectories, 2009..
[23] E. D'Andrea, P. Ducange, B. Lazzerini, F. Marcelloni, Real-time detection of traffic from twitter stream analysis, IEEE Trans. Intell. Transp. Syst. 16 (4) (2015) 2269–2283.
[24] R. Mohamed, H. Aly, M. Youssef, Accurate real-time map matching for challenging environments, IEEE Trans. Intell. Transp. Syst. 18 (4) (2017) 847–857.
[25] Essam Algizawy, Tetsuji Ogawa, Ahmed El-Mahdy, Real-time large-scale map matching using mobile phone data, ACM Trans. Knowl. Discovery Data 11 (4) (2017) 1–38.
[26] G. Schulze, C. Horn, R. Kern, Map-matching cell phone trajectories of low spatial and temporal accuracy, Sep. 2015..
[27] A. Janecek, D. Valerio, K.A. Hummel, F. Ricciato, H. Hlavacs, The cellular network as a sensor: from mobile phone data to real-time road traffic monitoring, IEEE Trans. Intell. Transp. Syst. 16 (5) (2015) 2551–2572.
[28] N. Caceres, L.M. Romero, F.G. Benitez, J.M. Del Castillo, Traffic flow estimation models using cellular phone data, IEEE Trans. Intell. Transp. Syst. 13 (3) (2012) 1430–1441.
[29] Feng Lin, Mingqi Lv, Ting Wang, Tieming Chen, Map matching based on cell-id localization for mobile phone users, Cluster Comput. 22 (3) (2019) 6231–6239.
[30] Paul Newson, John Krumm, Hidden markov map matching through noise and sparseness, in: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2009, pp. 336–343.

Qiuyang Huang received his B.E. degree in software engineering from Jilin University, Changchun, in 2013, his M.E. degree in software engineering from Jilin University, Changchun, in 2016. He is currently an Ph.D. in the computer science and technology at Jilin University, Changchun. His research interests include applications of data mining, urban computing, and mobile computing. He has published some research results on journals such as scientific data.

Yongjian Yang is currently a professor and a Ph.D. supervisor at Jilin University, the Vice Dean of Software College of Jilin University, also Director of Key lab under the Ministry of Information Industry, Standing Director of Communication Academy, member of the Computer Science Academy of Jilin Province. His research interests include: Network intelligence management; Wireless mobile communication and services; research and exploiture for next generation services foundation and key productions on wireless mobile communication. He participated 3 projects of NSFC, 863 and funded by National Education Ministry for Doctoral Base Foundation. He has authored 12 projects of NSFC, key projects of Ministry of Information Industry, Middle and Young Science and Technology Developing Funds, Jilin provincial programs, ShenZhen, ZhuHai, and Changchun.

Yuanbo Xu received his B.E. degree in computer science and technology from Jilin University, Changchun, in 2012, his M.E. degree in computer science and technology from Jilin University, Changchun, in 2015, and his Ph.D. in computer science and technology from Jilin University, Changchun, in 2019. He is currently an Postdoc in the Department of Artificial Intelligence at Jilin University, Changchun. His research interests include applications of data mining, recommender system, and mobile computing. He has published some research results on journals such as TMM, TNNLS and conference as ICDM.

Funing Yang received her B.E. degree in college of software engineering from Jilin University, Changchun, in 2010, received the master's degree from the school of computer science, Beijing University of Posts and Telecommunications,Beijing,China,in 2013. She is currently a teacher in institute of applied technology,Jilin University, Changchun. Her current research interests include management, integration and mining of massive traffic data.

Zhilu Yuan was born in Heilongjiang, China in 1986. He received his Ph.D. degree from School of Transportation, Jilin University in 2017. He is currently a Postdoc in the Research Institute for Smart Cities, School of Architecture and Urban Planning, Shenzhen University, Shenzhen, China. His research interests is transportation planning, including traffic flow modeling, pedestrian flow modeling, pedestrian simulation research, related software and theories and applications of transportation planning.

Yongxiong Sun was born in November 1970. He is currently an associate professor at the college of computer science and technology of Jilin University. The research direction is computer application technology. The main research areas are: Internet of Things, big data, intelligent transportation.