Estimation of Link Travel Time Distribution With Limited Traffic Detectors

Peibo Duan, *Member, IEEE*, Guoqiang Mao, *Fellow, IEEE*, Jun Kang, and Baoqi Huang, *Member, IEEE*

***Abstract*— Motivated by the network tomography, in this paper, we present a novel methodology to estimate link travel time distributions (TTDs) using end-to-end (E2E) measurements detected by the limited traffic detectors at or near the road intersections. As it is not necessary to monitor the traffic in each link, the proposed estimator can be readily implemented in real life. The technical contributions of this paper are as follows: First, we employ the kernel density estimator (KDE) to model link travel times instead of parametric models, e.g., Gaussian distribution. It is able to capture the dynamic of link travel times that vary with the change of road conditions. The model parameters are estimated with the proposed C-shortest path algorithm, K-means-based algorithm, as well as expectation maximization (EM) algorithm. Second, to reduce the complexity of parameter estimation, we further propose a Q-opt and an X-means-based algorithm. Finally, we validate our proposed method using a dataset consisting of 3.0e + 07 GPS trajectories collected by the taxicabs in Xi'an, China. With the metrics of Kullback Leibler and Kolmogorov–Smirnov test, the experimental results show that the link TTDs obtained from our proposed model are in excellent agreement with the empirical distributions, provided that $\sim 70\%$ of the intersections are equipped with traffic detectors.**

***Index Terms*— Link travel time distribution, network tomography, kernel density estimator, expectation maximization (EM) algorithm.**

I. Introduction

TRAVEL time plays an important role in measuring traffic conditions of the road networks. In most studies, travel times are estimated at level of link or path, where a link is usually defined as a oneway road segment without any road intersections inside, while a path is composed of a sequence of links [1], [2]. In this paper, we focus on link travel time estimation since link travel times deliver more benefits to both travelers and traffic administrators. On the one hand, it allows travelers to make optimal route choice to minimize their overall travel times. On the other hand, traffic administrators can accurately locate where congestion happens, and carry out effective traffic management to improve road network performance accordingly.

Recently, travel time distribution (TTD) estimation has attracted considerable research attention. Unlike mean travel time estimation where travel time is estimated as a deterministic variable [1], [3]–[5], TTD estimation assumes travel time to be a random variable, which addresses the intuition that travel time is time-varying due to the heterogeneous and dynamic nature of traffic [4], [6], [7]. Moreover, the knowledge of the moments (e.g., mean and variance) obtained from a probability distribution can be used as the indicators to analyze travel time reliability [8].

Existing methods to estimate link TTDs suffer from the following shortcomings. First, many methods are based on the parametric models like Gaussian distribution or log-normal distribution [9], [10]. These models are easy for mathematical analysis, yet unable to capture all of interesting dynamics of travel times that vary with the change of road conditions [3], [5], [11]. Second, to estimate the model parameters such as the means and variances in a Gaussian distribution, it is necessary to guarantee that there are sufficient travel time data in the target links [6]. Unfortunately, this condition cannot be satisfied in an urban road network involving thousands of links, that are impractical to be fully covered by any type of data detector, e.g., global position systems (GPSs) or traffic cameras. Third, given the links where there is no observation, the travel times of these links are usually estimated based on contexts learned from their spatially and temporally correlated neighbors. However, the spatio-temporal correlation varies with the time of the day [12]. Therefore, not only is it hard to guarantee the estimation accuracy of link travel times, but also a large amount of computing resource are consumed on data modeling [13].

Network tomography and travel time estimation bear a strong resemblance to each other. They both face the problem that the internal features (link delay in the network tomography, link travel time in the road network) cannot be directly measured because of the limited coverage of observers (beacons in the network tomography, traffic cameras in the road network). Such similarity enables the techniques for network tomography to have the potential on estimating link travel times. To the best of our knowledge, the study closest to ours is Zhang *et al.*'s work [7]. The authors estimate mean travel times using a linear model, which is also feasible in the case that the travel times follow Gaussian distribution [14].

Manuscript received February 13, 2019; revised June 19, 2019; accepted July 12, 2019. The Associate Editor for this paper was W. Jin. *(Corresponding author: Baoqi Huang.)*

P. Duan and G. Mao are with the School of Computing and Communication, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: Peibo.Duan@student.uts.edu.au; g.mao@ieee.org).

J. Kang is with the Department of Internet of Things and Network Engineering, Chang'an University, Xi'an 710064, China (e-mail: junkang@chd.edu.cn).

B. Huang is with the College of Computer Science, Inner Mongolia University, Hohhot 010021, China (e-mail: cshbq@imu.edu.cn).

Digital Object Identifier 10.1109/TITS.2019.2932053

However, the Gaussian distribution has the shortages discussed above.

To cope with the aforementioned problems, in this paper, we estimate link TTDs with a non-parametric model, namely kernel density estimator (KDE). The model parameters are estimated using the data collected at or near the road intersections. The main contributions are briefly summarized as follows:

- With the proposed KDE based model, we are able to capture the dynamic of link travel times that vary with the change of road conditions. The model parameters are estimated with the proposed C-shortest path algorithm, K-means based algorithm, as well as the expectation maximization (EM) algorithm.
- We analyze the performance bottleneck of the proposed parameter estimation algorithms. To reduce the complexity and guarantee the estimation accuracy, we propose a Q-opt algorithm and an X-means based algorithm.
- We validate the proposed method based on a dataset including over $3.0\mathrm{e}+07$ GPS trajectories collected by the taxicabs in Xi'an, China. The experimental results show that the TTDs estimated using our proposed model are in excellent agreement with empirical distribution, provided that only $\sim 70\%$ of the intersections are equipped with traffic detectors.

The organization of the paper is as follows: Related work is summarized in Section II. Then we introduce the principles of network tomography and KDE in Section III. The proposed model and parameter estimation algorithms are described in Section IV and V respectively. We explore the performance of the proposed model in Section VI. Finally, we draw the conclusion in Section VII.

II. Related Work

Research on link travel time estimation are mainly divided into two categories: 1) mean travel time estimation, and 2) TTD estimation. In the remainder of this section, we first introduce the work in terms of mean travel time estimation according to the utilization of traffic detectors. After that, the research with respect to TTD estimation are summarized and analyzed according to the application of probabilistic models.

Data driven methods are well studied in the mean travel time estimation because of the coming of big data era for transportation. These traffic data are collected by different types of detectors, mainly including GPS, Bluetooth device, loop detector and traffic camera. A significant part of GPS based methods is devoted to solve map matching and data sparsity problems. Map matching aims to calibrate the GPS coordinates that did not fall into the roads where the vehicles were traveling in. The corresponding approaches are on the basis of geometric, topological, probabilistic, and artificial intelligence. The details of these approaches have been introduced in Sanaullah et al.'s work [15]. Data sparsity is mainly caused by the low sampling frequency and limited number of data. The solve this problem, the distinguished work was done by Wang *et al.* [13] and Tang *et al.* [1]. They both introduced the tensor, a technique utilized in the deep learning, to model travel times on different links as multi-linear manner geometric vectors. As the neighboring links are spatially correlated, the tensor without observed data were estimated based on the geospatial, temporal and historical contexts learned from the neighboring tensors.

Bluetooth device is an alternative to provide traffic data for mean travel time estimation. The Bluetooth device in each vehicle has the unique Media Access Control address (MAC address). The traffic information of these vehicles will be captured by the Bluetooth Traffic Monitoring Systems (BTMSs) installed in the roads. Bluetooth based methods focus on solving the following two issues: i) the measurement reliability produced by BTMSs, e.g., transmission power, and ii) the probability of detecting the same vehicle by two successive Bluetooth detectors. To get rid of these problems, Bhaskar *et al.* [16] calibrated Bluetooth data with the aid of loop detectors. José *et al.* [17] introduced a series of weights to adjust the travel times estimated from Bluetooth data. The weights were predefined according to different traffic patterns such as free flow or congestion.

Loop detector is also a widely used detector in mean travel time estimation. Unlike GPS and Bluetooth, it cannot identify the vehicles. As a result, related research are usually on the basis of traffic flow theory. In other words, travel times were inferred from traffic flow and travel speed [18], [19]. For instance, Li *et al.* [18] designed an a temporal-spatial queuing model with consideration of travel speed, headway time series and travel times. Yi and Williams [19] proposed a dynamic Nam-Drew model to estimate travel times under traffic conditions of free flow and congestion.

Traffic camera collect data through videos or images. With the rapid development of the techniques in the realm of artificial intelligence [20], [21], the improvement of vehicle recognition accuracy enable the traffic camera data to become more reliable. Unfortunately, it is impractical to monitor every link in the whole urban network by traffic cameras. To address this problem, Yeon *et al.* [22] developed a Discrete Time Markov Chains (DTMC) model with consideration of different road conditions like congestion and free flow. Rahmani *et al.* [23] proposed a method extended from kernel-based estimation by means of both traffic camera data and GPS data.

In recent years, growing interest is motivating a shift toward estimating TTD. The corresponding work usually assume travel times follow either Gaussian distribution or log-normal distribution. Specially, Li *et al.* [24] indicated Gaussian was appropriate to model travel time in the presence of free flow, small time interval (e.g., 5 minutes), whereas log-normal was appropriate to model travel time in the presence of congestion, large time interval (half an hour). Other probabilistic models were also used to model travel times such as Weibull distribution and Burr distribution [25]. To improve the estimation accuracy, Pu [26] used the log-normal model with consideration of the inter dependencies between the reliability measures of travel times such as standard deviation, coefficient of variation and frequency of congestion. Moylan and Rashidi [6] constructed multiple hazard-based models under different road conditions leveraging on the factors affecting the variation of

road conditions such as the weather, the wind speed were modeled as explanatory variables. Prokhorchuk *et al.* [4] proposed a Gaussian copula graphical model to transform the non-Gaussian characteristics of travel times into Gaussian. Yang *et al.* [8] developed a Gaussian mixture model by considering the delay in the signalized intersections. From above literature, we can observe that these methods are parametric model based. As the structure (the number of parameters) of a parametric model is fixed, it is difficult for them to capture all of interesting dynamic of travel times varying with the change of road conditions [12].

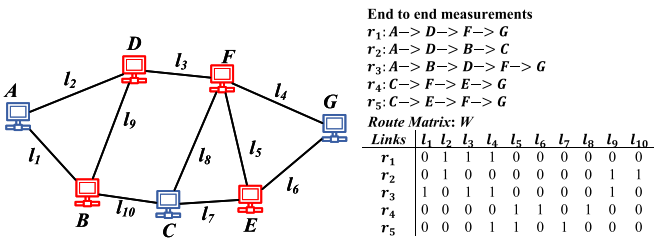
A common weakness in the existing research on TTD estimation is that they seldom consider impact with respect to the limited coverage of traffic detectors. This is because the proposed estimators are generally implemented in the typical study sites like the major roads in the urban city [6]. The traffic detectors deployed in these study sites are dense. Thus, there are always sufficient observations. However, consider the whole urban network, a traffic detector, e.g., traffic camera, is far away from another. It leads to a problem that the traffic states in the links between two traffic detectors are unobservable. To solve this problem, techniques of network tomography are the candidates. More concretely, network tomography uses the information derived from end-to-end (E2E) measurements to explore the internal characteristics of an internet network, e.g., the packet transmission delay. In the context of traffic network, the travel times detected by the two traffic detectors can be viewed as the E2E measurements. Thus, the TTDs of the links between the two traffic detectors can be inferred from the observations. Motivated by this idea, Zhang *et al.*'s [7] provided a traffic camera deployment strategy with which the accuracy of mean travel time estimation was improved, and meanwhile, the overall deployment cost on traffic cameras was minimized.

Different from Zhang *et al.*'s work, we focus on TTD estimation. Although similar problem like the estimation of link delay distribution has been researched in the network tomography, the techniques cannot be directly used in our work since most of them are based on parametric models such as Gaussian or exponential distributions with the shortcomings discussed in Section I. Note that bin size model is a kind of non-parametric model [27], [28] used in the network tomography, however, it can vary wildly with the different configuration of bins, especially with relatively small number of data. Therefore, in this paper, we use kernel density estimator which provides similar distribution even with varying bandwidth and/or kernel type.

III. Preliminary

A. Network Tomography

To illustrate the principle of network tomography, a concrete example is given in Fig.1, where there are 10 links, denoted by $\{l_i|i = 1, 2, \ldots, 10\}$. The E2E measurements are taken by the beacons configured at $\{A, C, G\}$. Suppose the packets are transmitted through the routes $r_j \subseteq \{r_1, r_2, r_3, r_4, r_5\}$ (on the upper right of Fig. 1), the E2E measurements on each route are denoted by $Y = \{y_1, y_2, y_3, y_4, y_5\}$. $\forall y_j \in Y$ can be formulated as $y_j = \sum_{i=1}^{10} w_{ji} x_i$, where x_i is the delay on l_i;



Links	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}
r_1	0	1	1	1	0	0	0	0	0	0
r_2	0	1	0	0	0	0	0	0	1	1
r_3	1	0	1	1	0	0	0	0	1	0
r_4	0	0	0	0	1	1	0	1	0	0
r_5	0	0	0	1	1	0	1	0	0	0

Fig. 1. An instance of network tomography.

$w_{ji} = 1$, if l_i is covered by r_j, otherwise $w_{ji} = 0$. Given a route matrix W where each row represents a route and each column represents a link (on the bottom right of Fig. 1), we formulate the delays on all the routes as:

$$Y^{\mathrm{T}} = WX^{\mathrm{T}}, \tag{1}$$

where $X = \{x_i|i = 1, 2, \ldots, 10\}$, Y^{T} and X^{T} are the transposes of Y and X respectively. A large number of methods has been proposed to get the solution of X [14], [29].

B. Kernel Density Estimator

The probability density function (PDF) of the kernel density estimator (KDE) is defined as:

$$p(\boldsymbol{x}) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x_i - \boldsymbol{x}}{h}), \tag{2}$$

where $\boldsymbol{x}$ is the random variable, n is the number of samples, $h > 0$ is called the smoothing bandwidth that controls the amount of smoothing, x_i is the i-th sample, $K(\boldsymbol{x})$ is named the kernel (function) that is generally a smooth and symmetric function. There are various choices among kernels, such as uniform, triangle, Gaussian, and Epanechnikov kernels. The best fitting performance (the lowest mean square error) is obtained with the Epanechnikov kernel. However, it will reduce the estimation efficiency. The fitting performances of uniform, triangle, and Gaussian kernels are similar. Therefore, for the sake of mathematical analysis, in this paper, we use the Gaussian kernel [30]. In particular, $K(\frac{x_i - \boldsymbol{x}}{h})$ follows the standard normal distribution of $\mathcal{N}(0, 1)$. Equivalently, we can rewrite (2) as follows:

$$p(\boldsymbol{x}) = \frac{1}{nh} \sum_{i=1}^{n} \mathcal{N}(\boldsymbol{x}|u_i, \sigma_i^2), \quad u_i = x_i, \ \sigma_i = h. \tag{3}$$

IV. KDE Based Model

To help readers keep track of symbols' meanings, we clarify the major notations in Table 1. Moreover, we provide a flowchart in Fig.2 to describe our proposed method. It includes:

- *Model building*: Given a study site, we use KDE to model travel time distributions across all the links (Section IV).
- *Parameter estimation*: The number of model parameters is closely related to the placement of traffic detectors. To this end, we design a C-shortest path algorithm (Section V-A) with which the maximal number of paths between any pair of traffic detectors is C. As there may be $C > 1$ paths between two traffic detectors, it is not

TABLE I
SYMBOLS TABLE

Symbols	Significance
$G = \{V, E\}$	The digraph model of the road network
V_{meas}	The set of vertices deployed with traffic detectors (measurement points)
V_{unmeas}	The set of vertices without traffic detectors
r	A path between a pair of measurement points
d_r	The number of edges (links) in r
$\boldsymbol{t}_{e_k}$	The random variable of travel time for link e_k
$\boldsymbol{t}_r$	The random variable of travel time for the path r
t (t_r)	An E2E measurement (collected from the path r)
M	The number of pairs of measurement points
R_m	The set of the paths between the m-th pair of measurement points
R	$\{R_m \vert \forall m \in M\}$
$\mathbb{R}$	$\{r_j \vert r_j \subseteq R_m, \forall m \in M\}$
T_m	The set of samples collected by the m-th pair of measurement points during a time interval
T	$\{t \in T_m \vert \forall m \in M\}$
$\hat{\mathrm{T}}$	The subset of T after sampling with X-means based algorithm
$p_{t\vert r_j}$	A binary variable $\{0, 1\}$ denotes whether t is collected on r_j ($p_{t\vert r_j} = 1$) or not ($p_{t\vert r_j} = 0$)
$P_{t\vert R_m}$	$\{p_{t\vert r_j} \vert \forall r_j \subseteq R_m, m \in M\}$
$\mathbf{P}_{\mathrm{T}\vert\mathrm{R}}$	$\cup P_{T_m \vert R_m}, \forall m \in M$
$\mathbf{P}_{\mathrm{T}\vert\mathbb{R}}$	$\{p_{t\vert r_j} \vert t \in \mathbf{T}, r_j \subseteq \mathbb{R}\}$
$\mathbb{T}$	$\{t \vert t \in \mathbf{T}, p_{t\vert r_j} = 1\}$
n_{e_k}	The number of vehicles traveling in e_k during a time interval
h_{e_k}	The bandwidth of the KDE model for e_k
$\mu_{e_k,i}$	The travel time for the i-th vehicle traveling in e_k
μ_{e_k}	$\{\mu_{e_k,i} \vert i \in n_{e_k}\}$
Θ_{e_k}	$\{n_{e_k}, h_{e_k}, \mu_{e_k}\}$
Θ_{r_j}	$\{\Theta_{e_k} \vert e_k \in r_j\}$
$\mathbf{\Theta}_{\mathrm{R}}/\mathbf{\Theta}_{\mathbb{R}}$	$\{\Theta_{e_k} \vert e_k \in E\}$
W	The route matrix where each row represents a path and each column represents an edge.
B_W	A basis of W
$\mathbf{B}_W$	The set of bases of W
$\boldsymbol{y}_{r_j}$	The latent variables where $\forall y_z \in \boldsymbol{y}_{r_j}$ satisfies $y_z = \{0, 1\}$ and $\sum_{y_z \in \boldsymbol{y}_{r_j}} y_z = 1$
$\gamma_{r_j}(y_z)$	The probability $p(y_z = 1 \vert t_{r_j})$

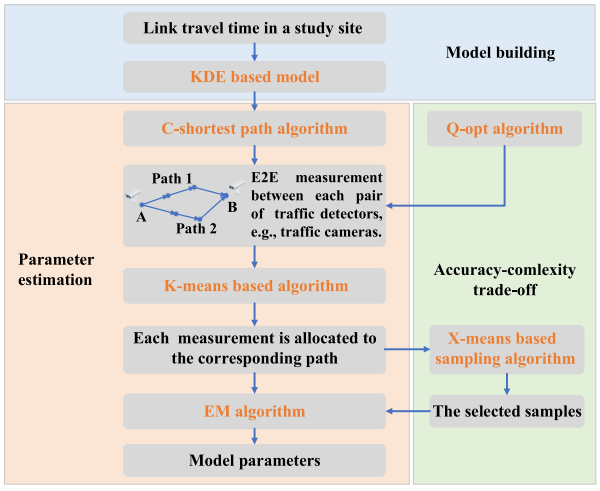


Fig. 2. Flowchart of the proposed method.

clear which path an E2E measurement is collected from. Therefore, we propose a data allocation strategy based on K-means algorithm (Section V-B). Lastly, EM algorithm (Section V-C) is implemented to estimate the parameters.

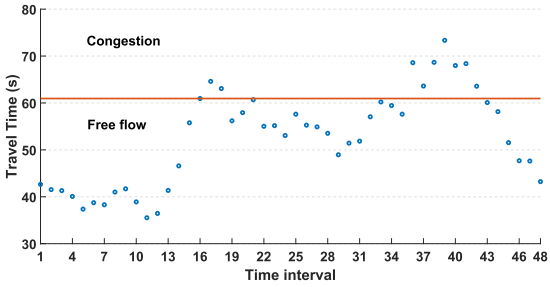


Fig. 3. The average travel time on a road in each time interval. The red line is the 80% of all the points.

- *Accuracy-complexity trade-off*: The complexity of EM algorithm depends on the number of links and the number of E2E measurements between two traffic detectors (the details will be illustrated in Section V-D). To guarantee the accuracy-complexity trade-off, we first design a Q-opt algorithm so that the maximal number of links in a path is Q. It is executed together with the C-shortest path algorithm. To filter the E2E measurements that contribute little to the parameter estimation, we propose an X-means based sampling algorithm, executed after K-means based algorithm.

To illustrate the advantage of KDE model, we analyze the cumulative density functions (CDFs) of travel times on a randomly selected link in Xi'an road network. The details of data will be illustrated in Section VI. In the analysis, we divide a day into 48 time intervals, each of which is half an hour (e.g., time interval 1 represents time period from 00:00:00am to 00:30:00am). We consider two road conditions: free flow and congestion. To identify the road condition in each time interval, we use the method proposed by Nguyen *et al.* [31]. specifically, a road is considered to be congested, if the mean travel time in a time period is greater than n-th percentile of the mean travel times in the whole time intervals. In this paper, we use $n = 80$. The red line in Fig.3 is the 80th percentile of all the mean travel times in 48 time intervals. The congestion happened in the time intervals where the points are above the red line. Otherwise, the road is in the state of free flow.

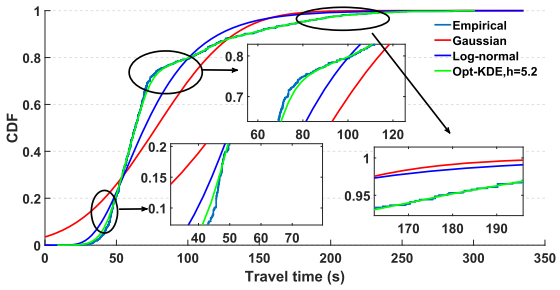
In Fig. 4a and Fig. 4b, we present the CDFs of the empirical data, Opt-KDE, Gaussian distribution and log-normal distribution under the road conditions of congestion and free flow respectively. Opt-KDE represents the KDE with optimal bandwidth estimated by the biased cross-validation method proposed by Scott and Terrell [32]. From Fig. 4, we can observe that the CDF of Opt-KDE matches the empirical data better than the other two models. Furthermore, we use the KS (Kolmogorov-Smirnov) test to measure the similarity between the CDFs of two distributions. The null hypothesis of the KS test is that the two distributions are the same. Given a significance level ($\alpha = 0.01$), we reject the null hypothesis, if the maximal distance between two CDFs is greater than the critical value, $c(\alpha)$, defined by:

$$c(\alpha) = \sqrt{-\frac{1}{2}\ln\alpha} \cdot \sqrt{\frac{n+m}{nm}}, \tag{4}$$

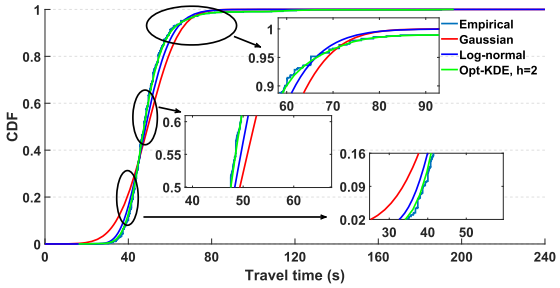
where n is the size of empirical data and m is the size of data used for probabilistic model estimation. Moreover, we use

TABLE II
THE KS TEST BASED ON DIFFERENT PROBABILISTIC MODELS WITH SIGNIFICANCE LEVEL $\alpha = 0.01$

$D_{em,es}$	Opt-KDE	KDE ($h = 3s$)	KDE ($h = 4s$)	KDE ($h = 5s$)	Gaussian	log-normal
Free flow ($c(\alpha) = 0.100$)	0.031	0.039	0.053	0.072	0.167 (reject)	0.100
Congestion ($c(\alpha) = 0.088$)	0.022	0.030	0.029	0.022	0.259 (reject)	0.169 (reject)



(a) CDFs under congestion (8:30am-9:00am, the 18th time interval in Fig.3)



(b) CDFs under free flow (14:00pm-14:30pm, the 29th time interval in Fig.3)

Fig. 4. CDFs based on empirical data, Opt-KDE, Gaussian and log-normal models under the congestion and free flow respectively.

$D_{em,es}$ to denote the maximal distance between the CDFs of empirical data and estimated distribution. From Table II, we find that the Gaussian model is rejected in the case of free flow and both Gaussian and log-normal models are rejected in the case of congestion. Moreover, with the bandwidth $2s \leq h \leq 5s$, KDE model also fit the data better than the Gaussian and log-normal models. Similar results can also be obtained based on the data in other links.

We model a road network as a digraph. Specifically, we partition the road network into a set of *links* where each link is an one-way road segment bounded by two road intersections and there is no intersection within a link. Drawing from the graph theory, the digraph model is represented as $G = (V, E)$ where V is the set of vertices and E is the set of directed edges. Each vertex $V_i \in V$ represents an intersection. There exists an edge $e_{ij} \in E, e_{ij} = (V_i, V_j)$ if there is a link with traveling direction from V_i to V_j. We name a vertex V_i as a measurement point if there are observations detected by the traffic detectors like traffic cameras at V_i. Then $V = \{V_{meas}, V_{unmeas}\}$ where V_{meas} is the set of measurement points and $V_{unmeas} = V - V_{meas}$. Obviously, the TTD of e_{ij} can be estimated if both end points of e_{ij}, $V_i, V_j \in V_{meas}$. However, in the real life, it is impractical to cover $\forall V_i \in V$ with traffic detectors. As a result, there is always a sequence of links between two measurement points. With the principle of graph theory, we define a travel route between one intersection and another as a path, denoted by:

$$r = \{e_1, e_2, \ldots, e_{d_r}\}, \tag{5}$$

where d_r is the number of links and the edges in r are all distinct from each other. Given a path r between $V_i, V_j \in V_{meas}$, we can obtain the travel time on r with the observations at V_i and V_j. For instance, a vehicle travels from V_i to V_j through r and is captured by the traffic cameras at V_i and V_j at time t_1 and t_2, then the travel time $t = t_2 - t_1$. In this paper, we also name t as an E2E measurement.

Consider the situation that the positions of the traffic data collected by some traffic detectors are not exactly located at the road intersections, but somewhere nearby, e.g., GPS data collected by the probe vehicles. We use a distance and time proportion method to estimate E2E measurements. More details will be illustrated in Section VI. We use the bold-faced letter $\boldsymbol{t}$ to represent a random variable of travel time. The objective of our work is to estimate the distribution of $\boldsymbol{t}_{e_k}$ for $\forall e_k \in E$ with the E2E measurements detected by the limited traffic detectors.

We assume the travel times for the vehicles traveling in different links are spatially independent. Meanwhile, we assume different vehicles traveling in the same link will experience independent travel times. We respectively term these two assumptions as spatial independence and temporal independence. In practice, travel times in the links are generally spatially and temporally correlated to a greater or lesser extent. However, these correlations are usually not strong enough. In addition, ignoring dependencies can also have benefits on the analysis. For instance, in [13], to simply the objective function of path travel time estimation, Wang et al. assumed the travel times on different links are independent. Based on the above analysis, it is sufficient for us to use these assumptions to derive the estimates of TTD.

We model the distribution of $\boldsymbol{t}_{e_k}$ with KDE as follows:

$$p(\boldsymbol{t}_{e_k}|\Theta_{e_k}) = \frac{1}{n_{e_k} h_{e_k}} \sum_{i=1}^{n_{e_k}} \mathcal{N}(\boldsymbol{t}_{e_k}|\mu_{e_k,i}, h_{e_k}^2), \tag{6}$$

where $\Theta_{e_k} = \{n_{e_k}, h_{e_k}, \mu_{e_k}\}$ is the set of parameters. More precisely, n_{e_k} is the number of vehicles traveling through e_k during a time interval, h_{e_k} is the bandwidth and $\mu_{e_k} = \{u_{e_k,i} | i = 1, 2, \ldots, n_{e_k}\}$ where $u_{e_k,i} \in \mu_{e_k}$ is the travel time when the i-th vehicle traverses e_k. As $\boldsymbol{t}_r = \sum_{k=1}^{d_r} \boldsymbol{t}_{e_k}$, the distribution of $\boldsymbol{t}_r$ conditioned on Θ_{e_k} can be parameterized as follows:

$$p(\boldsymbol{t}_r|\Theta_r) = p(\boldsymbol{t}_{e_1}|\Theta_{e_1}) * \ldots * p(\boldsymbol{t}_{e_{d_r}}|\Theta_{e_{d_r}}), \tag{7}$$

where $*$ represents the convolution operation and $\Theta_r = \{\Theta_{e_k} | k \in d_r\}$.

In the network tomography, the transmission route of a packet is always known. However, in our work, the path r where an E2E measurement is collected is usually unknown because of the following two reasons: i) the limited coverage of traffic detectors makes the travel route unobservant, and ii) there may be multiple paths between two measurement points. We use $R = \{r_1, r_2, \ldots, r_{|R|}\}$ to denote the alternative paths between two measurement points where $|\cdot|$ is the cardinality of a set. Given an E2E measurement t, we introduce a binary variable $p_{t|r_j}, r_j \in R$ where $p_{t|r_j} = 1$ if t is collected from $r_j \subseteq R$ and $p_{t|r_j} = 0$ otherwise. Obviously, $\sum_{r_j \in R} p_{t|r_j} = 1$ since an E2E measurement is collected only from a unique route. We use $P_{t|R} = \{p_{t|r_1}, p_{t|r_2}, \ldots, p_{t|r_{|R|}}\}$ to represent the set of binary variables for t based on routes R, so that the probability of t conditioned on $P_{t|R}$ and Θ_R is modeled by:

$$p(t|P_{t|R}, \Theta_R) = \prod_{r_j \in R} p(t_{r_j}|\Theta_{r_j})^{p_{t|r_j}}, \tag{8}$$

where $\Theta_R = \{\cup\Theta_{r_j} | r_j \subseteq R\}$. Given the set of E2E measurements between two measurement points in a time interval, T. We define $P_{T|R} = \cup_{t \in T} P_{t|R}$, then the log-likelihood of T is formulated as:

$$\mathcal{L}(T|P_{T|R}, \Theta_R) = \sum_{t \in T} \ln p(t|P_{t|R}, \Theta_R). \tag{9}$$

In a road network, suppose we have M pairs of measurement points, then we use $\mathrm{T} = \{t \in T_m | m \in M\}$ to denote the set of all E2E measurements over the whole study site. The log-likelihood of T is formulated as:

$$\mathcal{L}(\mathrm{T}|\mathbf{P}_{\mathrm{T|R}}, \mathbf{\Theta}_{\mathrm{R}}) = \sum_{m \in M} \mathcal{L}(T_m|P_{T_m|R_m}, \Theta_{R_m}), \tag{10}$$

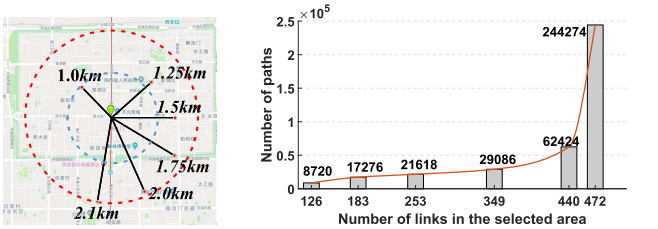
where $\mathrm{R} = \{R_m | m \in M\}$ is the set of the paths with measured data in the road network; $\mathbf{P}_{\mathrm{T|R}} = \cup_{m \in M} P_{T_m|R_m}$ and $\mathbf{\Theta}_{\mathrm{R}} = \cup_{m \in M} \Theta_{R_m}$. By substituting (7) and (9) into (10), we obtain $\mathcal{L}(\mathrm{T}|\mathbf{P}_{\mathrm{T|R}}, \mathbf{\Theta}_{\mathrm{R}})$ as follows:

$$\mathcal{L}(\mathrm{T}|\mathbf{P}_{\mathrm{T|R}}, \mathbf{\Theta}_{\mathrm{R}}) = \sum_{m \in M} \sum_{t \in T_m} \sum_{r_j \in R} p_{t|r_j} \ln p(t_{r_j}|\Theta_{r_j}). \tag{11}$$

To simplify (11), we introduce $\mathbb{R} = \cup_{m \in M} R_m$. Moreover, we define $\mathbf{P}_{\mathrm{T}|\mathbb{R}} = \{p_{t|r_j} | t \in \mathrm{T}, r_j \subseteq \mathbb{R}\}$ where $p_{t|r_j} = 1$ if and only if t is collected on route r_j and otherwise, $p_{t|r_j} = 0$. Obviously, the significance of $\mathbb{R}$ and $\mathbf{P}_{\mathrm{T}|\mathbb{R}}$ are equivalent to R and $P_{\mathrm{T|R}}$. Meanwhile, we introduce $\mathbf{\Theta}_{\mathbb{R}} = \{\cup\Theta_{r_j} | r_j \subseteq \mathbb{R}\}$. As both R and $\mathbb{R}$ should cover all the edges in G, we have $\mathbf{\Theta}_{\mathbb{R}} = \mathbf{\Theta}_{\mathrm{R}} = \{\cup\Theta_{e_k} | e_k \in E\}$. In this case, (11) can be represented as

$$\mathcal{L}(\mathrm{T}|\mathbf{P}_{\mathrm{T}|\mathbb{R}}, \mathbf{\Theta}_{\mathbb{R}}) = \sum_{t \in \mathbf{T}} \sum_{r_j \in \mathbb{R}} p_{t|r_j} \ln p(t_{r_j}|\Theta_{r_j}). \tag{12}$$

From (10), we can observe that the estimation of $\{\mathbf{P}_{\mathrm{T}|\mathbb{R}}, \mathbf{\Theta}_{\mathbb{R}}\}$ relies on $\mathbb{R}$. In the next section, we first illustrate the approaches to estimate $\mathbb{R}$, followed by the estimation of $\{\mathbf{P}_{\mathrm{T}|\mathbb{R}}, \mathbf{\Theta}_{\mathbb{R}}\}$.



(a) The selected areas (b) The number of paths in different selected area

Fig. 5. The selected area in Xi'an, China and the number of paths in different areas.

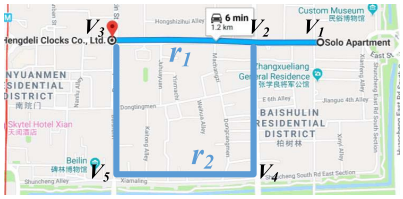


Fig. 6. A travel route between two intersections using $\mathbf{B}_W$ and Google Maps.

V. Parameter Estimation

In this section, we estimate $\mathbb{R}$ using a C-shortest paths based algorithm. After that, we estimate $\mathbf{P}_{\mathrm{T}|\mathbb{R}}$ with a K-means algorithm based approach. Next, we estimate $\mathbf{\Theta}_{\mathbb{R}}$ with the EM (Expectation Maximization) algorithm. Finally, to make a trade-off between the complexity and accuracy of the EM algorithm, we design a Q-opt algorithm together with a X-means algorithm.

A. The Estimation of $\mathbb{R}$

$\mathbb{R}$ has a close relationship with the placement of traffic detectors as well as the road topology. In [7], the authors proposed a traffic camera placement strategy based on the routing matrix W. Particularly, they calculated the bases of W, denoted by $\mathbf{B}_W$, where each basis $B_W \in \mathbf{B}_W$ was defined as a maximal subset of linearly independent routes. After that, the optimal basis $B_{opt} \in \mathbf{B}_W$ was obtained with the minimum cost on the deployment of traffic cameras.

However, the above routing matrix based method is faced with two problems. First, W is usually a high dimensional matrix, especially in the urban road networks. To show the relationship between the number of links and the scale of W, in Fig.5a, we select six regions in Xi'an, each of which is a disk centered at Zhonglou with the radius taking a value among {1.0km, 1.25km, 1.5km, 1.75km, 2.0km, 2.1km}. The larger the radius is, the more links are contained in the region. In Fig.5b, the x-coordinate presents the numbers of the links in these six regions are {126, 183, 253, 349, 440, 472}.

After modeling each region as a digraph G, we present the number of paths in each region in Fig.5. Obviously, with the growth of the number of links, the number of rows (paths) in W experiences an approximately exponential growth. The high dimension of W has an negative impact on the estimation of $\mathbf{B}_W$. Such phenomenon is mainly attributable to the fact that there may be hundreds or thousands of paths between two intersections. Second, the route in each row of B_{opt} does not always have observation data. For instance, in Fig.6,

Algorithm 1 The Estimation of $\mathbb{R}$

Input: $G = \{V, E\}$, C
Initialization: $\mathbb{R} \leftarrow \varnothing$, $B_W \leftarrow \varnothing$, $\hat{\mathbb{R}} \leftarrow \varnothing$
1 for $\forall V_i, V_j \in V_{meas}, i \neq j$
2 Estimating R_{ij}^C using Yen's algorithm [33]
3 $\hat{\mathbb{R}} \leftarrow \hat{\mathbb{R}} \cup R_{ij}^C$
4 endfor
5 Obtaining W using E and $\hat{\mathbb{R}}$
6 $B_W \leftarrow$ one basis of W
7 for $\forall V_i, V_j \in V_{meas}, i \neq j$
8 for each path r in each row of B_W
9 if V_i and V_j are the end points of r
10 $R \cup r$
11 endif
12 endfor
13 $\mathbb{R} \leftarrow \mathbb{R} \cup R$
14 endfor

we present an example where route r_2 (path V_1->V_2->V_4->V_5->V_3) is obtained from B_{opt}, whereas r_1 (path V_1->V_2->V_3) is achieved with Google Maps. Although r_2 covers more links than r_1, there is no observation data on r_2 since the motorists prefer traveling in r_1 due to the few travel time and short distance.

To solve the above problems, we propose a C-shortest paths based algorithm (Algorithm 1) to estimate $\mathbb{R}$. Therein, C is a manually set parameter which is used to control the maximal number of paths between two vertices. From line 1 to line 4, Yen's algorithm [33] is applied to find C-shortest paths between any pair of vertices in V_{neas}, denoted by $\hat{\mathbb{R}}$. The complexity is $O(C|V|^3(|E| + |V|\log|V|))$. After that, we obtain W, each row of which is a route in $\hat{\mathbb{R}}$ and each column of which is an edge in E. In line 6, a basis B_W of W is estimated using the Bareiss algorithm with the complexity of $O((|V| \cdot |E|)^2)$. Then we estimate $\mathbb{R}$ from line 7 to line 14 with the complexity of $O(|V|^2)$. Clearly, the complexity of Algorithm 1 is polynomial.

B. The Estimation of $\mathbf{P}_{\mathrm{T}|\mathbb{R}}$

Given an E2E measurement $t \in \mathrm{T}$ detected by the m-th pair of measurement points, we have $p_{t|r_j \subseteq \mathbb{R}-R_m} = 0$ and there exists only one path $r_j \in R_m$ with which $p_{t|r_j \in R_m} = 1$. In this case, The problem of estimating $p_{t|r_j} \in \mathbf{P}_{\mathrm{T}|\mathbb{R}}$ can be interpreted as a clustering problem, that is, to allocate t to a path in $\mathbb{R}$. To this end, we use the K-means algorithm, an unsupervised learning method, which is available for clustering data without labels.

We implement the K-means algorithm on each pair of measurement points parallelly. More precisely, for $T_m \subseteq \mathrm{T}$, we define $K = |R_m|$. Note that a problem we should address is that we do not know which path a cluster represents. As the shorter a path is, the less is the time that a vehicle needs to travel through. We allocate different paths into the clusters using the K-means algorithm in the following way:

- We classify the E2E measurements in T_m into $|R_m|$ clusters with the K-means algorithm.
- We evaluate the average travel time in each cluster and sort the clusters according to their average travel times.
- We sort the paths according to their lengths, then we map each cluster to each path according to the lengths of paths.

C. The Estimation of $\mathbf{\Theta}_{\mathbb{R}}$

Recall (6), $\forall\Theta_{e_k} \subseteq \mathbf{\Theta}_{\mathbb{R}}$ has the parameters $\{n_{e_k}, h_{e_k}, \mu_{e_k}\}$. Particularly, n_{e_k} is related to the number of E2E measurements collected on the paths that cover e_k. We define a $|\mathbb{R}|$ dimensional vector $P_{t|\mathbb{R}} = (p_{t|r_j}|r_j \subseteq \mathbb{R})$. Then n_{e_k} can be estimated by

$$n_{e_k} = \sum_{t \in \mathrm{T}} P_{t|\mathbb{R}} \cdot W^k, \tag{13}$$

where W is the route matrix estimated using line 5 in Algorithm 1, and W^k is the k-th column of W. n_{e_k} is the function of $\mathbf{P}_{\mathrm{T}|\mathbb{R}}$. In this case, the parameters in $\forall\Theta_{e_k} \subseteq \mathbf{\Theta}_{\mathbb{R}}$ are essentially $\{h_{e_k}, \mu_{e_k}\}$.

In order to estimate h_{e_k} and μ_{e_k}, we first simplify the representation of (6) based on: 1) the associative property of convolution, that is, $f_1(x) * (f_2(x) + f_3(x)) = f_1(x) * f_2(x) + f_1(x) * f_3(x)$, and 2) the property that the convolution of two Gaussian distributions, i.e. $\mathcal{N}(\mu_1, \sigma_1^2) * \mathcal{N}(\mu_2, \sigma_2^2)$, is also a Gaussian distribution in the format of $\mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$. With these two properties, (7) can be rewritten as (14), shown at the bottom of this page, where $\mathcal{Z}_r = \prod_{k=1}^{d_r} n_{e_k}$, $\mu_{r,z} = \sum_{k=1}^{d_r} u_{e_k,i}, \forall i \in n_{e_k}$ and $h_r^2 = \sum_{k=1}^{d_r} h_{e_k}^2$. To better understand (14), consider the following case:

Consider a route r covering two links, each link of which has two E2E measurements, that is, $n_{e_1} = 2$ and $n_{e_2} = 2$. Then, $p(\boldsymbol{t}_r|\Theta_r) = \frac{1}{2h_{e_1}}(\mathcal{N}(\boldsymbol{t}_{e_1}|\mu_{e_1,1}, h_{e_1}^2) + \mathcal{N}(\boldsymbol{t}_{e_1}|\mu_{e_1,2}, h_{e_1}^2)) * \frac{1}{2h_{e_2}}(\mathcal{N}(\boldsymbol{t}_{e_2}|\mu_{e_2,1}, h_{e_2}^2) + \mathcal{N}(\boldsymbol{t}_{e_2}|\mu_{e_2,2}, h_{e_2}^2))$. Based on (14)*, $\mathcal{Z}_r = 2 \times 2$. For each $z \in \mathcal{Z}_r$, we calculate $\mu_{r,z}$ by $\mu_{r,1} = \mu_{e_1,1} + \mu_{e_2,1}$, $\mu_{r,2} = \mu_{e_1,1} + \mu_{e_2,2}$, $\mu_{r,3} = \mu_{e_1,2} + \mu_{e_2,1}$, $\mu_{r,4} = \mu_{e_1,2} + \mu_{e_2,2}$, and calculate $h_{r,z}$ by $h_r^2 = h_{e_1}^2 + h_{e_2}^2$.*

Given the natural log of $p(\boldsymbol{t}_r|\Theta_r)$:

$$\ln p(\boldsymbol{t}_r|\Theta_r) = \sum_{k=1}^{d_r} \ln \frac{1}{n_{e_k}} + \sum_{k=1}^{d_r} \ln \frac{1}{h_{e_k}} + \ln \sum_{z=1}^{\mathcal{Z}_r} \mathcal{N}(\boldsymbol{t}_r|\mu_{r,z}, h_r^2). \tag{15}$$

$$\begin{aligned} p(\boldsymbol{t}_r|\Theta_r) &= \frac{1}{n_{e_1} h_{e_1}} \sum_{i=1}^{n_{e_1}} \mathcal{N}(\boldsymbol{t}_{e_1}|\mu_{e_1,i}, h_{e_1}^2) * \frac{1}{n_{e_2} h_{e_2}} \sum_{i=1}^{n_{e_2}} \mathcal{N}(\boldsymbol{t}_{e_2}|\mu_{e_2,i}, h_{e_2}^2) * \ldots * \frac{1}{n_{e_{d_r}} h_{e_{d_r}}} \sum_{i=1}^{n_{e_{d_r}}} \mathcal{N}(\boldsymbol{t}_{e_{d_r}}|\mu_{e_{d_r},i}, h_{e_{d_r}}^2) \\ &= \Big(\prod_{k=1}^{d_r} \frac{1}{n_{e_k} h_{e_k}}\Big) \cdot \sum_{z=1}^{\mathcal{Z}_r} \mathcal{N}(\boldsymbol{t}_r|\mu_{r,z}, h_r^2), \end{aligned} \tag{14}$$

Algorithm 2 EM Algorithm

Input:$\mathbb{R}$
Initialization: $\mathbf{\Theta}_{\mathbb{R}}^{(0)}$
1 for $q \in 1, 2, \ldots$
E-step:
2 $\gamma_{t_{r_j}}^{(q)}(y_z)$: Being updated using (19) with $\mathbf{\Theta}_{\mathbb{R}}^{(q-1)}$
M-step:
3 for each $\mu_{e_k,i}^{(q)}$ in $\Theta_{e_k}^{(q)} \subseteq \mathbf{\Theta}_{\mathbb{R}}^{(q)}$ and $h_{e_k}^{(q)}$, $e_k \in E$
4 $\mu_{e_k,i}^{(q)} \leftarrow \frac{\sum_{r_j \subseteq \mathbb{R}_{e_k}} \sum_{t_{r_j} \in \mathbb{T}_{r_j}} \sum_{z \in \mathcal{Z}_{r_j}(\mu_{e_k,i})} \gamma_{t_{r_j}}^{(q)}(y_z) t_{r_j}}{N_{r_j}}$
5 $(h_{e_k}^{(q)})^2 \leftarrow \frac{\sum_{r_j \subseteq \mathbb{R}_{e_k}} \sum_{t_{r_j} \in \mathbb{T}_{r_j}} \sum_{z \in \mathcal{Z}_{r_j}} \gamma_{t_{r_j}}^{(q)}(y_z)(t_{r_j} - u_{e_k,i})^2}{N_{r_j}}$
6 endfor
Terminal:
7 if $\mathbf{\Theta}_{\mathbb{R}}^{(q)}$ converges to a local optimum
8 return $\mathbf{\Theta}_{\mathbb{R}}^{(q)}$
9 endif
10 endfor

we obtain $\mathcal{L}(\mathrm{T}|\mathbf{P}_{\mathrm{T}|\mathrm{R}}, \mathbf{\Theta}_{\mathrm{R}})$ in (16), shown at the bottom of this page, where $\mathbb{T} = \{t | t \in \mathrm{T}, p_{t|r_j} = 1\}$.

From (16), we can observe that the parameters $\mathbf{\Theta}_{\mathbb{R}}$ are only included in $\mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}})$. Thus, setting the derivative of $\mathcal{L}(\mathbf{T}|\mathbf{P}_{\mathbf{T}|\mathbb{R}}, \mathbf{\Theta}_{\mathbb{R}})$ with respect to $\mathbf{\Theta}_{\mathbb{R}}$ to zero, we have

$$\frac{\mathrm{d}\mathcal{L}(\mathrm{T}|\mathbf{P}_{\mathrm{T}|\mathrm{R}}, \mathbf{\Theta}_{\mathrm{R}})}{\mathrm{d}\mathbf{\Theta}_{\mathbb{R}}} = \frac{\mathrm{d}\mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}})}{\mathrm{d}\mathbf{\Theta}_{\mathbb{R}}} = 0. \tag{17}$$

Unfortunately, there is no closed form solution for (17) due to the log of cumulative Gaussian distribution. As a result, the Maximum Likelihood (ML) method does not work here. To address this problem, we employ the EM algorithm to estimate $\mathbf{\Theta}_{\mathbb{R}}$ (Algorithm 2) based on the following assumption:

Assumption 1: The $h_{e_k}s$ of the KDE models for the travel time in $\forall e_k \in E$ are same.

To begin with, we introduce the latent variables. For $\forall r_j \subseteq \mathbb{R}$, we define $\mathcal{Z}_{r_j}$-dimensional latent variables as $\boldsymbol{y}_{r_j}$ in which $\forall y_z \in \boldsymbol{y}_{r_j}$ satisfies $y_z = \{0, 1\}$ and $\sum_{y_z \in \boldsymbol{y}_{r_j}} y_z = 1$. Given the definition that the marginal distribution over $\boldsymbol{y}_{r_j}$ is $p(y_z = 1) = \mathcal{Z}_{r_j}^{-1}$, we formulate the distribution of $\boldsymbol{y}_{r_j}$ as $p(\boldsymbol{y}_{r_j}) = \prod_{y_z \in \boldsymbol{y}_{r_j}} \mathcal{Z}_{r_j}^{-y_z}$. We also define the conditional distribution of an E2E measurement t_{r_j} as a Gaussian distribution with $p(t_{r_j}|y_z = 1) = \mathcal{N}(t_{r_j}|\mu_{r_j,z}, h_{r_j}^2)$. The joint distribution of t_{r_j} is given by:

$$\begin{aligned} p(t_{r_j}) &= \sum_{\boldsymbol{y}_{r_j}} p(\boldsymbol{y}_{r_j}) p(t_{r_j}|\boldsymbol{y}_{r_j}) \\ &= \mathcal{Z}_{r_j}^{-1} \sum_{z \in \mathcal{Z}_{r_j}} \mathcal{N}(t_{r_j}|\mu_{r_j,z}, h_{r_j}^2) \end{aligned} \tag{18}$$

We define $\gamma_{r_j}(y_z) \equiv p(y_z = 1|t_{r_j})$, which can be calculated based on Bayes theorem:

$$\begin{aligned} \gamma_{r_j}(y_z) &= \frac{p(y_z = 1) p(t_{r_j}|y_z = 1)}{p(t_{r_j})} \\ &= \frac{\mathcal{N}(t_{r_j}|\mu_{r_j,z}, h_{r_j}^2)}{\sum_{z \in \mathcal{Z}_{r_j}} \mathcal{N}(t_{r_j}|\mu_{r_j,z}, h_{r_j}^2)} \end{aligned} \tag{19}$$

In Algorithm 2, $\mathbf{\Theta}_{\mathbb{R}}^{(0)}$ are the initial values of $\mathbf{\Theta}_{\mathbb{R}}$. In line 4, $\mathbb{R}_{e_k} = \{r_j | e_k \in r_j, r_j \subseteq \mathbb{R}\}$, $\mathcal{Z}_{r_j}(u_{e_k,i}) \triangleq \{z | z \in \mathcal{Z}_{r_j}, \mu_{e_k,i} \in u_{r_j,z}\}$ and N_{r_j} is

$$N_{r_j} = \sum_{r_j \subseteq \mathbb{R}_{e_k}} \sum_{t_{r_j} \in \mathbb{T}_{r_j}} \sum_{z \in \mathcal{Z}_{r_j}(\mu_{e_k,i})} \gamma_{r_j}(y_z(t_{r_j})). \tag{20}$$

As the performance of the EM algorithm heavily relies on $\mathbf{\Theta}_{\mathbb{R}}^{(0)}$, we use the initialization strategy given in [34]. Convergence is achieved when $\mathbf{\Theta}_{\mathbb{R}}^{(q)} \approx \mathbf{\Theta}_{\mathbb{R}}^{(q-1)}$. The proof of parameters update in line 4 and 5 is presented in the appendix A.

D. Q-opt and X-Means Based Sampling Algorithm

In the proposed EM algorithm, we can observe that the computational complexity in each iteration depends on $\mathcal{Z}_r$. Further, (14) indicates that $\mathcal{Z}_r$ is determined by d_r and n_{e_k}. Therefore, the way to reduce the computational complexity is to limit the path length d_r and reduce the value of n_{e_k}.

Intuitively, the smaller d_r is, the less computational complexity it costs. However, more traffic detectors are needed in the road network if d_r is smaller. Consider the worst case that $d_r = 1$, each path can be viewed as a link. In this case, each intersection should be configured with a traffic detector. To guarantee the accuracy-complexity trade-off, and to control the number of traffic detectors, we propose a Q-opt Algorithm, termed Algorithm 3, where Q means the maximal number of links in a path $d_r \leq Q$.

In Algorithm 3, $numTD$ is the number of traffic detectors needed in a road network. Line 3 to 7 guarantee the number

$$\begin{aligned} \mathcal{L}(\mathrm{T}|\mathbf{P}_{\mathrm{T}|\mathrm{R}}, \mathbf{\Theta}_{\mathrm{R}}) &= \sum_{t \in \mathbb{T}} \left(\sum_{k=1}^{d_{r_j}} \ln \frac{1}{n_{e_k}} + \sum_{k=1}^{d_{r_j}} \ln \frac{1}{h_{e_k}} + \ln \sum_{z=1}^{\mathcal{Z}_{r_j}} \mathcal{N}(t|\mu_{r_j,z}, h_{r_j}^2) \right) \\ &= \sum_{t \in \mathbb{T}} \sum_{k=1}^{d_{r_j}} \ln \frac{1}{n_{e_k}} + \sum_{t \in \mathbb{T}} \left(\sum_{k=1}^{d_{r_j}} \ln \frac{1}{h_{e_k}} + \ln \sum_{z=1}^{\mathcal{Z}_{r_j}} \mathcal{N}(t|\mu_{r_j,z}, h_{r_j}^2) \right) \\ &= \sum_{t \in \mathbb{T}} \sum_{k=1}^{d_{r_j}} \ln \frac{1}{n_{e_k}} + \mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}}), \end{aligned} \tag{16}$$

Algorithm 3 Q-opt

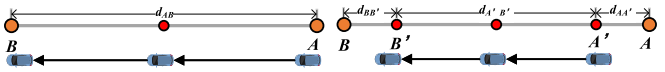
Input: $G = \{V, E\}$, Q
Initialization: $V_{meas} \leftarrow \varnothing$, $numTD$, $\hat{B} \leftarrow \varnothing$
1 Estimate $\hat{\mathbb{R}}$ using line 1 to 4 in Algorithm 1
2 for $q = 1$ to Q
3 for each $r \subseteq R_{ij}^K$, $R_{ij}^K \subseteq \hat{\mathbb{R}}$
4 if $d_r > Q$
5 Remove r from R_{ij}^K
6 endif
7 endfor
8 Calculate $\hat{B}$ based on line 5 and line 6 in Algorithm 1
9 for each column $\hat{B}_k$
10 if $\hat{B}_k = \mathbf{0}$
11 $\hat{B} = \left[\hat{B}; I^k\right]$
12 endif
13 endfor
14 for each row $\hat{B}_j$
15 $\boldsymbol{V}_k \leftarrow$the end points of $\hat{B}_j$
16 $V_{meas} \cup \boldsymbol{V}_k$
17 endfor
18 if $|V_{meas}| < numTD$
19 $numTD \leftarrow |V_{meas}|$
20 $B \leftarrow \hat{B}$
21 endif
22 endfor
23 Estimate $\mathbb{R}$ based on line 7 to 14 in Algorithm 1

of paths between two vertices is no more than C and the length of each path is no more than Q. Note that a link may not be covered by any row in $\hat{B}$ (line 10). To address this issue, in line 11 we expand $\hat{B}$ by a $|E|$ dimensional vector I^k where the k-th element in I^k is 1 and the other elements are 0. From line 18 to 21, we obtain B with minimum $numTD$. Given C and Q, the complexity of Algorithm 3 is the same as Algorithm 1.

n_{e_k}, as the function of $\mathbf{P}_{\mathrm{T}|\mathrm{R}}$, is related with $\mathbb{R}$ and T. As $\mathbb{R}$ has been estimated using Algorithm 1, the way to reduce n_k is to use the subset of T, denoted by $\hat{\mathrm{T}}$, following the principle that the dynamic characteristics of E2E measurements in T can be perfectly captured by the selected E2E measurements in $\hat{\mathrm{T}}$. To obtain $\hat{\mathrm{T}}$, we first employ the X-means algorithm [35] to classify the E2E measurements on each path $T_{r_j \in \mathbb{R}}$ into X clusters, each of which represents a feature of the data. The procedure of the X-means algorithm contains the following three steps:

- *Step 1:* Given an initial value of $X = X_{in}$, we run the conventional K-means algorithm till reaching convergence.
- *Step 2:* To find out whether there is a new centroid using the splitting strategy in [35]. More precisely, we randomly select a centroid and run the K-means algorithm. We will accept such a new centroid if the resulting model score is better than before. After that, we have $X = X + 1$.
- *Step 3:* Repeat the second step until X reaches a given threshold X_{thre} or there is no improvement on the resulting model score.

Unlike the K-means algorithm where the number of clusters K is manually set in advance, the number of clusters in the X-means algorithm is identified automatically. Thus, the X-means algorithm is able to better capture the dynamic features of E2E measurements in T_{r_j} than the K-means algorithm. After that, we obtain a subset of T_{r_j}, denoted by $\hat{T_{r_j}}$, by selecting data from each cluster using the simple random sampling algorithm [36]. Finally, we obtain $\hat{\mathrm{T}} = \cup \hat{T}_{r_j \subseteq \mathbb{R}}$.



(a) Only one GPS data record reported by a vehicle (b) More than one GPS data record reported by a vehicle

Fig. 7. The instance of calculating link travel time for a vehicle using GPS data.

VI. Experimental Results

A. Experiment Setup

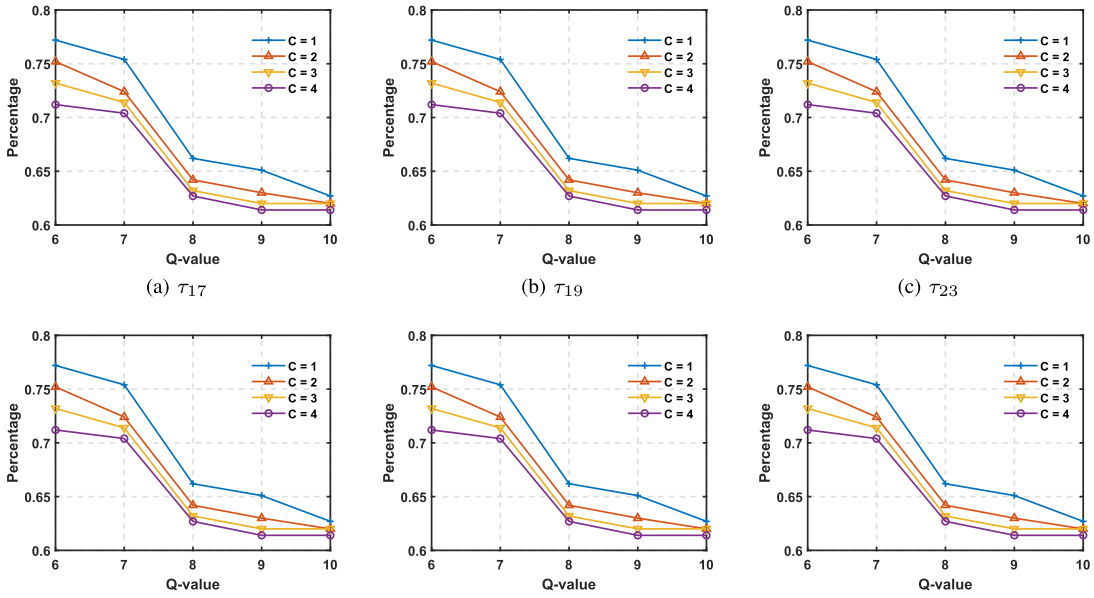
In this paper, the study site is based on the citywide road network in Xi'an, China, which covers 30,549 links. To validate our proposed method, we use the GPS trajectories anonymously reported by over 11,000 taxicabs on Sep. 5th, 2016 (Mon.). With the average sampling frequency of 30 seconds, we yield over 3.0e+07 raw data records. Each data has the travel information including the time stamp when the data was sent to the server, the location coordinates (longitude and latitude), the instantaneous travel speed and travel state that takes values from **{stop, cruising, occupied}**. We divide the day into 48 equal time intervals, denoted by $\{\tau_i | i = 1, 2, \ldots, 48\}$ where $\forall \tau_i$ represents half an hour, e.g., the time interval between 8:00am-8:30am. After that, we set up the model in each τ_i and implement the TTD estimation in Java and Matlab.

Noises exists in the collected GPS data, mainly on account of the precision of GPS. Thus we carry out data preprocessing as follows:

- **Map matching**: We employ a weight-based topological algorithm proposed by Velaga *et al.* [37], Zou *et al.* [38]. There are two stages in the algorithm: i) calculating the weight score for each of the candidate links where a GPS data record is probably in; ii) selecting the link with the highest weight as the correct link for a GPS data record.
- **Outliers filtering**: We filter the outliers mainly including: i) the locations of the GPS data that are out the scope of Xi'an city; ii) the data where the travel speeds exceed the speed limitation, i.e. 120km/h; iii) the data where there are conflicts between the travel state and travel speed (e.g., the state of vehicle is "stop" but the travel speed is not 0); iv) the data where the taxicabs are not in service.

B. Ground Truth

In this paper, we adopt the following two ground truths: *1) Opt-KDE*: link travel time distributions estimated by KDE with the optimal bandwidth. As discussed in Section IV, Opt-KDE can fit the distribution of empirical data better than any other models. Thus, we compare the results of our proposed method with Opt-KDE to validate estimation accuracy; *2) Empirical CDF*: the CDF of empirical data.



(a) τ_{17} (b) τ_{19} (c) τ_{23}

(d) τ_{31} (e) τ_{35} (f) τ_{41}

Fig. 8. The percentage of intersections that should deploy traffic detectors with different configurations of Q and C in different time intervals.

TABLE III

THE NUMBER OF LINKS AND TRAVEL STATES IN EACH TIME INTERVAL

	Time intervals	Travel state	No. of links	No. of intersections
τ_{17}	8:00am-8:30am	Congestion	4934	3545
τ_{19}	9:00am-9:30am	Congestion	5271	4031
τ_{23}	11:00am-11:30am	Free flow	5178	3804
τ_{31}	15:00pm-15:30pm	Free flow	5023	3752
τ_{35}	17:00pm-17:30pm	Congestion	5201	3957
τ_{41}	20:00pm-20:30pm	Free flow	4885	3524

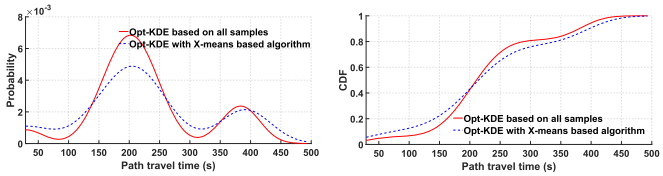
TABLE IV

AN INSTANCE OF K-MEANS BASED ALGORITHM USING THE DATA COLLECTED FROM THE PATHS BETWEEN A AND B IN τ_{19}

	Path	No. of data
1	A->B->C->F	16
2	A->B->D->E->F	12

Path	1	2
1	13	3
2	2	10

Fig. 9. The paths between two endpoints A and F.



(a) PDFs (b) CDFs

Fig. 10. The performance of X-means based algorithm based on the instance in Fig.9.

TABLE V

THE $AACC$ OF K-MEANS BASED ALGORITHM AND GREEDY APPROACH

Time interval	τ_{17}	τ_{19}	τ_{23}	τ_{31}	τ_{35}	τ_{41}
K-means	81.2%	80.7%	79.5%	86.2%	87.4%	81.6%
Greedy	74.8%	73.1%	68.3%	71.9%	69.5%	74.7%

TABLE VI

AVERAGE KL DIVERGENCE FOR DIFFERENT MODELS IN THE SELECTED TIME INTERVALS

Time interval	KDE-E2E	Gaussian	log-normal	GMM	Hazard
τ_{17}	0.92	1.51	1.24	0.93	1.13
τ_{19}	**1.03**	1.46	1.37	**0.89**	1.07
τ_{23}	0.96	1.73	1.19	1.01	1.15
τ_{31}	**0.90**	1.93	0.91	**0.87**	0.97
τ_{35}	0.94	1.49	1.25	0.98	1.14
τ_{41}	0.86	1.28	0.94	0.91	1.02

With KS test defined in Section IV, we can observe whether the estimated probability distribution is accepted or not.

The travel time when a vehicle traverses a link is calculated in two different ways, which depend on the number of GPS data records reported by this vehicle:

- Fig.7a illustrates the case with only one GPS data record is reported by a vehicle. The travel time t_{AB} is calculated by d_{AB}/v_{AB}, where d_{AB} is the length of link AB and v_{AB} is the space mean speed inferred from the instantaneous speed using the method in Appendix B.
- Fig.7b presents the case that multiple GPS data records reported by a vehicle. These data might not exactly reside at the endpoints of the link. In this example, GPS data records are transmitted at the points labeled by red. Assuming A' and B' are the two GPS data records closest to A and B, respectively. Furthermore, the timestamps

TABLE VII
THE KS TEST BASED ON DIFFERENT PROBABILISTIC MODELS WITH SIGNIFICANCE LEVEL $\alpha = 0.01$

$D_{em,es}$	τ_{17} $c(\alpha) = 0.120$	τ_{19} $c(\alpha) = 0.085$	τ_{23} $c(\alpha) = 0.093$	τ_{31} $c(\alpha) = 0.115$	τ_{35} $c(\alpha) = 0.960$	τ_{41} $c(\alpha) = 0.120$
Opt-KDE	0.027	0.033	0.021	0.029	0.045	0.030
KDE-E2E	0.042	**0.059**	0.037	0.045	0.064	0.038
Gaussian	0.231 (reject)	0.189 (reject)	0.204 (reject)	0.176 (reject)	0.271 (reject)	0.266 (reject)
log-normal	0.117	0.134 (reject)	0.802	0.095	0.131 (reject)	0.128 (reject)
GMM	0.051	**0.046**	0.043	0.055	0.088	0.039
Hazard	0.109	0.087	0.091	0.103	0.882	0.130 (reject)

when the taxicab sent GPS data at A' and B' are $t_{A'}$ and $t_{B'}$. Clearly, $t_{AB} \neq t_{A'B'} = t_{A'} - t_{B'}$. To counter this effect, we apply the method, namely distance and time proportion proposed by Sanaullah *et al.*'s [15], to calculate link travel time. Take Fig.7 as an example, t_{AB} is calculated by:

$$t_{AB} = \frac{d_{AB}}{d_{A'B'}} t_{A'B'}. \tag{21}$$

Similarly, we use the method to evaluate the path travel times.

C. Results

Due to the page limit, we only present the estimated results in the six representative time intervals including the traffic states of free flow and congestion. Note that the parameters of Opt-KDE and other counterparts (including Gaussian, log-normal, GMM, and hazard-based methods) are estimated assuming that there are traffic detectors deployed on all the links. In this case, we should select a set of links from the whole road network, each of which has sufficient observed data (Table III). By comparing with these methods, we are able to observe the advantage of our proposed method, which only uses a limited number of traffic detectors. In Fig.8, we present the percentage of intersections that should deploy traffic detectors. We then observe that fewer traffic detectors are needed when the values of Q and C become larger. In addition, the trend of curves is similar when $C = 2$, 3 and 4. This can be explained by the fact that given a Q, there are at most two candidate paths between most measurement points. Moreover, the percentage converges when $Q = 10$. In the best scenario, approximately 63% of intersections require traffic detectors, and in the worst scenario, approximately 70% of intersections require deploying traffic detectors (in Fig. 8f). The convergence reflects the fact that the basis (B_W) obtained from route matrix W changes little when $Q \geq 8$. Based on the above analysis, the following experiments are implemented based on the results obtained from the algorithm with $Q = 10$ and $C = 2$. It is worth noting that the optimal strategy of traffic detector placement is the problem that should be discussed distinctively. It will be studied in our future work.

In Table IV, we present the experimental results of K-means based algorithm using the data collected by the traffic detectors at the endpoints A and F shown in Fig.9. The first path has the links AB, BC, and CF. The second path has the links AB, BD, DE, and EF. In τ_{19}, we have 16 E2E measurements collected on path 1 and 12 on path 2 (the left side of Table IV). The right side of Table IV shows that 13 out of 16 (10 out of 12) E2E measurements collected on path 1 (2) are accurately allocated to the corresponding links. We define ACC as clustering accuracy (the percentage of correct decisions). Then, the ACC of K-means for the given instance is 82.1%. To present the performance of K-means based algorithm over the whole study site in each time interval, we use the average clustering accuracy ($AACC$) calculated by $AACC = \frac{\sum_{m \in M} ACC_m}{M}$, where ACC_m is the ACC of K-means algorithm implemented on the data collected from the paths between the m-th pair of measurement points. In Table V, we can observe that the best result is obtained in τ_{35} with $AACC = 87.4\%$. The worst result is $AACC = 79.5\%$ in τ_{23}. Compared to the greedy approach used in [7], the experimental results show a good performance of our proposed K-means based algorithm.

TABLE VIII
AVERAGE KL DIVERGENCE AND KS TEST OVER 48 TIME INTERVALS

Models	KDE-E2E	Gaussian	log-normal	GMM	Hazard
KL	0.89	1.62	1.31	0.93	1.07
KS	0	37	25	0	5

Taking path 1 in Fig. 9 as an instance, we present the performance of X-means based algorithm in Fig. 10. More precisely, the red solid lines show the PDF and CDF of path TTD using Opt-KDE model with 16 E2E measurements. However, only half (8) E2E measurements are needed using X-means based sampling algorithm. From the figures, we can observe that the distributions using the selected E2E measurements is similar with the ones using the whole data. Similar experimental results are obtained based on the E2E measurements on other paths. Therefore, it is sufficient for us to believe that the E2E measurements filtered by X-means based sampling algorithm can be viewed as the representatives of the whole E2E measurements, and used for parameter estimation without losing too much accuracy. As discussed in Section V-D, with the limited number of E2E measurements, the efficiency of EM algorithm will be improved.

To assess the deviation between an estimated TTD and the ground truth (Opt-KDE) in a link, we use the metric named Kullback Leibler (KL) divergence, which is defined as follows:

$$D_{KL}(P_{opt} || P_{es}) = \sum_{t \in T_{e_k}} p_{em}(t) \ln \frac{p_{em}(t)}{p_{es}(t)}, e_k \in E. \tag{22}$$

In (22), p_{opt} represents the TTD of Opt-KDE, and p_{es} is the estimated TTD with our proposed model (namely KDE-E2E) and its counterparts. In particular, GMM has three components

TABLE IX
AVERAGE KL DIVERGENCE AND KS TEST OVER ALL THE LINKS IN EACH TIME INTERVAL

Time interval	Gaussian		log-normal		GMM		Hazard	
	KL	KS	KL	KS	KL	KS	KL	KS
τ_{17}	100%	86.2%	99.8%	54.8%	13.5%	0.5%	99.1%	3.4%
τ_{19}	100%	90.4%	98.3%	58.5%	4.6%	0	97.0%	5.3%
τ_{23}	100%	83.2%	100%	63.0%	9.2%	0.3%	99.5%	4.7%
τ_{31}	100%	75.9%	100%	55.3%	6.9%	0	100%	3.4%
τ_{35}	100%	92.2%	99.4%	58.3%	12.1%	0.3%	98.6%	4.5%
τ_{41}	100%	80.4%	100%	61.4%	14.3%	0.8%	99.7%	9.2%

of Gaussians. The hazard-based model was proposed by Emily and Taha in [6]. It has a good performance in estimating TTD by considering the factors like travel speed, weather, road condition, etc. As we do not have the data like the weather, we cannot re-build the model as the one in [6]. In our paper, we only use the traffic speed and road condition to set up the hazard-based model. To evaluate the performance of the estimated results, we define the average KL divergence by:

$$\bar{D}_{KL}(P_{opt}||P_{es}) = \frac{\sum_{e_k \in E} D_{KL}(P_{opt}||P_{es})}{|E|}. \tag{23}$$

From Table VI, we can observe that the performance of KDE-E2E is always better than Gaussian, log-normal and hazard-based model in each time interval, but a little worse than GMM in τ_{17} and τ_{31}. This can be explained by the fact that GMM has the similar structure with Opt-KDE. However, in the real world, it is difficult to estimate the parameters of GMM because there is usually a lack of data on the target links. By comparing $\bar{D}_{KL}$s of KDE-E2E under different road conditions, we can also find out that the minimal and maximal $\bar{D}_{KL}$s under free flow are 0.86 (τ_{41}) and 0.96 (τ_{23}) respectively, whereas the minimal and maximal $\bar{D}_{KL}$s under the congestion are 0.92 (τ_{17}) and 1.03 (τ_{19}) respectively. The smaller is$\bar{D}_{KL}$, the better is the estimation accuracy. Therefore, the estimation accuracy of our proposed KDE-E2E method is superior under the free flow. This can be explained by the fact that the fluctuation of travel times is usually large under congestion. Thus, it is difficult to capture all the features of the variation of travel times.

In Table VII, we use the KS test to measure the similarity between the empirical CDF and the estimated one based on our proposed model and the counterparts. Particularly, we present the results of a randomly selected link. Obviously, our proposed model is accepted in each time interval. Compared to KDE-E2E and GMM, we can find that $D_{em,es}$ of GMM in τ_{19} is smaller than $D_{em,es}$ obtained from our proposed model. This result is consistent with the result in Table VI.

In Table VIII, we calculate KL divergence and implement KS test over 48 time intervals based on KDE-E2E and its counterparts. The second row denotes average KL divergence ($\bar{D}_{KL}$) and the third row denotes the number of time intervals in which the model is rejected. From Table VIII, we can observe that $\bar{D}_{KL}$s of KDE-E2E are smaller than the other models over 48 time intervals. Meanwhile, KDE-E2E models are all accepted in the whole time intervals. The second best model is GMM. Not surprisingly, the Gaussian model has the worst performance since it is rejected in 37 time intervals, whereas our proposed methods are accepted in each time interval. In Table IX, we compare KDE-E2E with other models using the same metrics over all the links in each time interval. More precisely, the column labeled by "KL" shows the percentage of links that our model is better than the other methods based on KL divergence. The column labeled by "KS" means the percentage of the links that our model is better than the other models based on KS test. Apparently, our method is much better than Gaussian and log-normal models. According to KS test results, GMM and the hazard-based methods have good performance for link TTD estimation. However, $\bar{D}_{KL}$s of hazard-based model are still smaller than those obtained by our proposed KDE-E2E over most links. As for GMM, there are still at most 14.3% links whose $\bar{D}_{KL}s$ are smaller than our method. This can be explained by the fact that the fixed number of Gaussian components in GMM is limited in capturing all the features of TTD. Combining with experimental results in Table VI and VII, the efficiency of our proposed method is further validated.

VII. CONCLUSION

Motivated by the network tomography, in this paper, we estimated TTDs with the E2E measurements detected by a limited number of traffic detectors deployed at or near the intersections. With the proposed KDE-E2E method, traffic administrators can deploy traffic detectors (e.g., traffic cameras) or dispatch probe vehicles to collect traffic data at some critical positions. Thus, a lot of resources can be saved. Furthermore, through observing and analyzing the distribution of travel times in the links, traffic administrators can carry out effective measures to avoid the occurrence of congestion. As the number of traffic detectors is related to financial costs, it is part of our future work to explore the optimal strategy for traffic detector placement, such as the minimum number of traffic detectors requiring to be deployed.

APPENDIX A
M-STEP IN EM ALGORITHM

We use $\mathbb{T}_{r_j}$ to denote the E2E measurements collected on $r_j \subseteq \mathbb{R}$. Then $\mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}})$ can be rewritten as

$$\mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}}) = \sum_{r_j \subseteq \mathbb{R}} \sum_{t_{r_j} \in \mathbb{T}_{r_j}} \sum_{k=1}^{d_{r_j}} \ln \frac{1}{h_{e_k}} + \sum_{r_j \subseteq \mathbb{R}} \sum_{t_{r_j} \in \mathbb{T}_{r_j}} \ln \sum_{z=1}^{\mathcal{Z}_{r_j}} \mathcal{N}(t_{r_j}|\mu_{r_j,z}, h_{r_j}^2). \tag{24}$$

$$\frac{\partial \mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}})}{\partial \mu_{e_k,i}} = \sum_{r_j \subseteq \mathbb{R}_{e_k}} \sum_{t_{r_j} \in \mathbb{T}_{r_j}} \left(\frac{\partial \ln \sum_{z=1}^{\mathcal{Z}_{r_j}(\mu_{e_k,i})} \mathcal{N}(t_{r_j}|\mu_{r_j,z}, h_{r_j}^2)}{\partial \mu_{r_j,z}} \cdot \frac{\partial \mu_{r_j,z}}{\partial \mu_{e_k,i}} \right)$$
$$= \sum_{r_j \subseteq \mathbb{R}_{e_k}} \frac{\sum_{t_{r_j} \in \mathbb{T}_{r_j}} \sum_{z=1}^{\mathcal{Z}_{r_j}(\mu_{e_k,i})} \gamma_{t_{r_j}}(y_z)(\mu_{r_j,z} - t_{r_j})}{2h_{r_j}^2}, \tag{25}$$

Further, we use $\mathbb{R}_{e_k}$ to represent the set of paths which cover e_k. Then we take the derivatives of $\mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}})$ with respect to $\mu_{e_k,i} \in \mu_{e_k}$in $\Theta_{e_k} \subseteq \mathbf{\Theta}_{\mathbb{R}}$. As $\mu_{r,z} = \sum_{k=1}^{d_r} u_{e_k,i}, \forall i \in n_{e_k}$, $\frac{\partial \mu_{r_j,z}}{\partial \mu_{e_k,i}} = 1$.Thus, $\frac{\partial \mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}})}{\partial \mu_{e_k,i}}$ is formulated as (25), shown at the top of this page, where $\gamma_{t_{r_j}}(y_z)$ is the responsibility.

Setting $\frac{\partial \mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}})}{\partial \mu_{e_k,i}}$ to zero, we find that it is also difficult to calculate $\mu_{e_k,i}$ since h_{r_j}s for $\forall r_j \subseteq \mathbb{R}_{e_k}$ are different. To simplify the calculation, we assume that h_{e_k}s on $\forall e_k \in E$ are the same (*Assumption 1*). With N_{r_j} defined in (20), we obtain $u_{e_k,i}$ as follows:

$$u_{e_k,i} = \frac{1}{N_{r_j}} \sum_{r_j \subseteq \mathbb{R}_{e_k}} \sum_{t_{r_j} \in \mathbb{T}_{r_j}} \sum_{z \in \mathcal{Z}_{r_j}(\mu_{e_k,i})} \gamma_{t_{r_j}}(y_z) t_{r_j}. \tag{26}$$

Similarly, setting the derivative of $\mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}})$ with respect to $h_{e_k}$to zero, we have

$$h_{e_k}^2 = \frac{1}{N_{r_j}} \sum_{r_j \subseteq \mathbb{R}_{e_k}} \sum_{t_{r_j} \in \mathbb{T}_{r_j}} \sum_{z \in \mathcal{Z}_{r_j}(\mu_{e_k,i})} \gamma_{t_{r_j}}(y_z)(t_{r_j} - u_{e_k,i})^2. \tag{27}$$

In the q-th iteration of M-step, we update $u_{e_k,i}^{(q)}$ and $(h_{e_k}^{(q)})^2$ with (26) and (27) using the responsibilities evaluated with the parameters $\mathbf{\Theta}_{\mathbb{R}}^{(q-1)}$.

Appendix B
Estimation of Space Mean Speed

We denote the instantaneous speed of a vehicle i traveling on a link by $v_{i,ins}$. Furthermore, we use v_{tms} and t_{sms} to denote time mean speed (TMS) and space mean speed (SMS) of vehicles traveling on the same link respectively. We regard t_{sms} as an approximate value of real SMS for the i-th vehicle. More precisely, based on [12], [39], we have the relationship between TMS and SMS as follows:

$$v_{tms} = v_{sms} + \frac{\sigma^2}{v_{sms}}, \tag{28}$$

where $\sigma^2 = E((v_{i,ins} - v_{sms})^2)$ and $E(v_{i,ins}) = v_{tms}$. Then, the solution to (28), v_{sms}, can be obtained as follows:

$$v_{sms} = \frac{3v_{tms} + \sqrt{9v_{tms}^2 - 8E(v_{i,ins}^2)}}{4} \tag{29}$$

Han *et al.* [40] assumed a quadratic relationship between $E(v_{i,ins}^2)$ and $E(v_{i,ins})$: $E[v_{i,ins}^2] = aE(v_{i,ins})^2 + bE(v_{i,ins}) + c$ where the parameters $\{a, b, c\}$ were estimated using 9304 samples as $\{a, b, c\} = \{1.22, -15.21, 207.95\}$. We estimate $E(v_{i,ins})$ by

$$E(v_{i,ins}) = \sum_{i=1}^{n} v_{i,ins}/n, \tag{30}$$

where n is the number of GPS trajectories collected on the link. Substituting (30) into (29), we have v_{sms}.

References

[1] K. Tang, S. Chen, and Z. Liu, "Citywide spatial-temporal travel time estimation using big and sparse trajectories," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 4023–4034, Dec. 2018.

[2] K. Kwong, R. Kavaler, R. Rajagopal, and P. Varaiya, "Real-time measurement of link vehicle count and travel time in a road network," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 4, pp. 814–825, Dec. 2010.

[3] M. Rahmani, E. Jenelius, and H. N. Koutsopoulos, "Non-parametric estimation of route travel time distributions from low-frequency floating car data," *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 343–362, Sep. 2015.

[4] A. Prokhorchuk, V. P. Payyada, J. Dauwels, and P. Jaillet, "Estimating travel time distributions using copula graphical lasso," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–6.

[5] S. Susilawati, M. A. Taylor, and S. V. C. Somenahalli, "Distributions of travel time variability on urban roads," *J. Adv. Transp.*, vol. 47, no. 8, pp. 720–736, Dec. 2013.

[6] E. K. M. Moylan and T. H. Rashidi, "Latent-segmentation, hazard-based models of travel time," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 2174–2180, Aug. 2017.

[7] R. Zhang, S. Newman, M. Ortolani, and S. Silvestri, "A network tomography approach for traffic monitoring in smart cities," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2268–2278, Jul. 2018.

[8] Q. Yang, G. Wu, K. Boriboonsomsin, and M. Barth, "A novel arterial travel time distribution estimation model and its application to energy/emissions estimation," *J. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 325–337, Nov. 2017.

[9] K. Wan, "Estimation of travel time distribution and travel time derivatives," Ph.D. dissertation, School Oper. Res. Financial Eng., Princeton Univ., Princeton, NJ, USA, 2014.

[10] D. Woodard, G. Nogin, P. Koch, D. Racz, M. Goldszmidt, and E. Horvitz, "Predicting travel time reliability using mobile phone GPS data," *Transp. Res. C, Emerg. Technol.*, vol. 75, pp. 30–44, Feb. 2017.

[11] M. Ramezani and N. Geroliminis, "On the estimation of arterial route travel time distribution with Markov chains," *Transp. Res. B, Methodol.*, vol. 46, no. 10, pp. 1576–1590, Dec. 2012.

[12] P. Duan, G. Mao, C. Zhang, and S. Wang, "Starima-based traffic prediction with time-varying lags," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1610–1615.

[13] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 25–34.

[14] N. G. Duffield and F. L. Presti, "Network tomography from measured end-to-end delay covariance," *IEEE/ACM Trans. Netw.*, vol. 12, no. 6, pp. 978–998, Dec. 2004.

[15] I. Sanaullah, M. Quddus, and M. Enoch, " Developing travel time estimation methods using sparse GPS data," *J. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 532–544, Apr. 2016.

[16] A. Bhaskar, M. Qu, and E. Chung, "Bluetooth vehicle trajectory by fusing Bluetooth and loops: Motorway travel time statistics," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 113–122, Feb. 2015.

[17] J. J. V. Díaz, A. B. R. González, and M. R. Wilby, "Bluetooth traffic monitoring systems for travel time estimation on freeways," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 123–132, Jan. 2016.

[18] L. Li, X. Chen, Z. Li, and L. Zhang, " Freeway travel-time estimation based on temporal–spatial queueing model," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1536–1541, Sep. 2013.
[19] T. Yi and B. M. Williams, "Dynamic traffic flow model for travel time estimation," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2526, pp. 70–78, Jan. 2015.
[20] J. Sochor, A. Herout, and J. Havel, "BoxCars: 3D boxes as CNN input for improved fine-grained vehicle recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3006–3015.
[21] S. Sivaraman and M. M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 267–276, Feb. 2010.
[22] J. Yeon, L. Elefteriadou, and S. Lawphongpanich, "Travel time estimation on a freeway using discrete time Markov chains," *Transp. Res. B, Methodol.*, vol. 42, no. 4, pp. 325–338, May 2008.
[23] M. Rahmani, E. Jenelius, and H. N. Koutsopoulos, "Floating car and camera data fusion for non-parametric route travel time estimation," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1286–1291.
[24] R. Li, G. Rose, and M. Sarvi, "Using automatic vehicle identification data to gain insight into travel time variability and its causes," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1945, pp. 24–32, Jan. 2006.
[25] Y. Guessous, M. Aron, N. Bhouri, and S. Cohen, "Estimating travel time distribution under different traffic conditions," *Transp. Res. Procedia*, vol. 3, pp. 339–348, Jan. 2014.
[26] W. Pu, "Analytic relationships between travel time reliability measures," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2254, pp. 122–130, Jan. 2011.
[27] N. G. Duffield, J. Horowitz, F. L. Presti, and D. Towsley, "Network delay tomography from end-to-end unicast measurements," in *Proc. Thyrrhenian Int. Workshop Digit. Commun.* Taormina, Italy: Springer, 2001, pp. 576–595.
[28] Y. Tsang, M. Coates, and R. D. Nowak, "Network delay tomography," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2125–2136, Aug. 2003.
[29] Y. Xia and D. Tse, "Inference of link delay in communication networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 12, pp. 2235–2248, Dec. 2006.
[30] B. W. Silverman, *Density Estimation for Statistics and Dataanalysis*. Evanston, IL, USA: Routledge, 2018.
[31] H. Nguyen, W. Liu, and F. Chen, "Discovering congestion propagation patterns in spatio-temporal traffic data," *IEEE Trans. Big Data*, vol. 3, no. 2, pp. 169–180, Jun. 2017.
[32] D. W. Scott and G. R. Terrell, "Biased and unbiased cross-validation in density estimation," *J. Amer. Stat. Assoc.*, vol. 82, no. 400, pp. 1131–1146, 1987.
[33] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Manage. Sci.*, vol. 17, no. 11, pp. 712–716, Jul. 1971.
[34] C. Biernacki, G. Celeux, and G. Govaertc, "Choosing starting values for the em algorithm for getting the highest likelihood in multivariate Gaussian mixture models," *Comput. Statist. Data Anal.*, vol. 41, nos. 3–4, pp. 561–575, Jan. 2003.
[35] D. Pelleg *et al.*, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Proc. ICML*, Jun. 2000, vol. 1, pp. 727–734.
[36] X. Meng, "Scalable simple random sampling and stratified sampling," in *Proc. Int. Conf. Mach. Learn.*, Feb. 2013, pp. 531–539.
[37] N. R. Velaga, M. A. Quddus, and A. L. Bristow, "Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems," *Transp. Res. C: Emerg. Technol.*, vol. 17, no. 6, pp. 672–683, Dec. 2009.
[38] H. Zou, B. Huang, X. Lu, H. Jiang, and L. Xie, "A robust indoor positioning system based on the Procrustes analysis and weighted extreme learning machine," *IEEE Trans. Wireless Commun.*, vol. 15, no. 2, pp. 1252–1266, Feb. 2016.
[39] B. Huang, L. Xie, and Z. Yang, "TDOA-based source localization with distance-dependent noises," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 468–480, Jan. 2014.
[40] J. Han, J. W. Polak, J. Barria, and R. Krishnan, "On the estimation of space-mean-speed from inductive loop detector data," *Transp. Planning Technol.*, vol. 33, no. 1, pp. 91–104, Dec. 2009.

Peibo Duan (S'16–M'18) received the B.S. and M.S. degrees from Northeastern University, Shenyang, China, in 2011 and 2013, respectively. He is currently pursuing the Ph.D. degree with the School of Computing and Communication, University of Technology Sydney (UTS) under the supervision of Prof. G. Mao. His current research interests include intelligent transportation system and distributed constraint optimization problem.

Guoqiang Mao (S'98–M'02–SM'08–F'18) joined the University of Technology Sydney as a Professor of wireless networking and the Director of Center for Real-time Information Networks in 2014. He has published about 200 papers in international conferences and journals, which have been cited more than 5000 times. His research interest include intelligent transport systems, applied graph theory and its applications in telecommunications, the Internet of Things, wireless sensor networks, wireless localization techniques, and network performance analysis. He is a fellow of IET.

Jun Kang received the B.E. degree in automatic control, the M.S. degree in pattern recognition and intelligent system, and the Ph.D. degree in control science and engineering from Northwestern Polytechnical University (NPU), Xi'an, China, in 1998, 2005, and 2009, respectively. Since 2009, he has been an Associate Professor with the School of Information Engineering, Chang'an University, Xi'an. His research interests include machine learning, big data analysis, intelligent transportation and information system, and networked control system.

Baoqi Huang (S'08–M'12) received the B.E. degree in computer science from Inner Mongolia University (IMU), Hohhot, China, in 2002, the M.S. degree in computer science from Peking University, Beijing, China, in 2005, and the Ph.D. degree in information engineering from Australian National University, Canberra, ACT, Australia, in 2012. From May 2013 to April 2014, he was a Research Fellow with Nanyang Technological University, Singapore. He is currently with the College of Computer Science, IMU, where he is also a Professor. His research interests include wireless sensor networks and mobile computing. He was a recipient of the Chinese Government Award for Outstanding Chinese Students Abroad in 2011.