# SharedEdge: GPS-Free Fine-Grained Travel Time Estimation in State-Level Highway Systems

YU YANG, Rutgers University, USA
FAN ZHANG, Shenzhen Institute of Advanced Technology, China
DESHENG ZHANG, Rutgers University, USA

Estimating travel time on the highway in real time is of great importance for transportation services. Previous work has been mainly focusing on the city scale for a particular transportation system, e.g., taxi, bus, and metro. Little research has been conducted to estimate fine-grained real-time travel time in state-level highway systems. This is because the traditional solutions based on probe vehicles or loop sensors cannot scale to state-level highway systems due to their large spatial coverage. Recently, the adoption of Electric Toll Collection (ETC) systems (e.g. EZ-pass) brings a new opportunity to estimate the real-time travel time in the highway systems with little marginal cost. However, the key challenge is that ETC data only record the coarse-grained total travel time between a pair of toll stations rather than fine-grained travel time in each individual highway edge. To address this challenge, we design SharedEdge to estimate the fine-grained edge travel time with large-scale streaming ETC data. The key novelty is that we estimate real-time fine-grained travel time (i.e., edge travel time) without using fine-grained data (i.e. GPS trajectories or loop sensor data), by a few techniques based on Bayesian Graphical models and Expectation Maximization. More importantly, we implement our SharedEdge in the Guangdong Province, China with an ETC system covering 69 highways and 773 toll stations with a length of 7, 000 km. Based on this implementation, we evaluate SharedEdge in details by comparing it with some baselines and the state-of-the-art models. The evaluation results show that SharedEdge outperforms other methods in terms of travel time estimation accuracy when compared with the ground truth obtained by 114 thousand GPS-equipped vehicles.

CCS Concepts: • **Networks** → **Sensor networks**; • **Information systems** → *Location based services*;

Additional Key Words and Phrases: Travel Time Estimation, Highway System, Cyber-physical System

**ACM Reference Format:**
Yu Yang, Fan Zhang, and Desheng Zhang. 2018. SharedEdge: GPS-Free Fine-Grained Travel Time Estimation in State-Level Highway Systems. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 1, Article 48 (March 2018), 26 pages. https://doi.org/10.1145/3191780

## 1 INTRODUCTION

Highway systems provide unhindered traffic flow with high throughput. Considering its great advantages in shortening travel time, highways are constructed both inner-city and intra-city [17]. For example, in the Guangdong province of China, there is 17 national highways and 52 provincial highways with a total length of 7000 km in 2015 [1]. The average daily trips on these highways are more than 4 million. Given the heavy demand, real-time estimations of travel time between entrances and exits are of great significance. Comparing

Authors' addresses: Yu Yang, Department of Computer Science, Rutgers University, Piscataway, NJ, 08854, USA, yy388@cs.rutgers.edu; Fan Zhang, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong, China, zhangfan@siat.ac.cn; Desheng Zhang, Department of Computer Science, Rutgers University, Piscataway, NJ, 08854, USA, desheng.zhang@cs.rutgers.edu.


2474-9567/2018/3-ART48 $15.00
https://doi.org/10.1145/3191780

with the typical estimation of the travel time between origins and destinations (OD), a fine-grained edge level estimation has more potential applications in many scenarios (Here we consider a road segment between two adjacent toll stations as an edge). For example, in the traffic anomaly detection, we need to pinpoint to road segments, i.e., edges, where the anomaly happens; in the context of navigation, we also need to know the travel time in individual edges to provide the fastest route.

Recently, with the updates of urban infrastructures, e.g., traffic cameras, traffic loop sensors, and GPS equipped probe vehicles, a lot of work has been done to estimate travel time within cities or across cities. These methods solve the problem from two aspects: one is to monitor vehicles on the roads with static sensors like loop sensors, the other is to use mobile sensors like probe vehicles. However, unlike the road network inside the cities, the highway system generally covers a larger scale area. This different context makes previous static-sensor methods difficult to apply since it is costly and unpractical to deploy sensors on all edges in a highway system, e.g. Guangdong has a highway system of 7000 km. For mobile sensors such as probe vehicles, it is also difficult to have enough vehicles in real time covering a highway system of 7000 km, which leads a low penetration rate with limited spatial-temporal coverage on the highway system. For the current services like Google Map [3], it requires collecting historical or real-time locations, which bring privacy and energy consumption issues using devices like smartphones [5] [35].

In this paper, we argue that the well-adopted Electronic Toll Collection (ETC) systems on highways bring a new opportunity to transparently estimate real-time travel time. The highway administrators have accumulated a huge amount of raw ETC data, including vehicle id, entrance, exit, and the corresponding times, which can be used to infer travel time with little marginal cost. Compared with static-sensor methods, an ETC system leverages existing infrastructures without any extra deployment of sensors, which reduces cost extensively. Compared with mobile-sensor methods, an ETC system captures all vehicles running on highways when they pass the toll stations. The travel time of each vehicle is obtained by the difference between the timestamps of passing an entrance station and a corresponding exit station. However, this coarse-grained travel time is for station-to-station trips, instead of for fine-grained edges. Even it is not the finest granularity compared with GPS tracking, however, the edge level is the finest result that can be achieved using ETC systems. Since it is impossible for vehicles to leave the highway until they arrive at the toll stations, the edge travel time can provide meaningful insights for drivers and administrators.

To address this granularity issue, we design SharedEdge, a probabilistic framework to estimate fine-grained edge travel time in highway systems. The key observation for SharedEdge is that vehicles may visit the same edges in their highway trips between different stations, which lead to much richer knowledge of travel time for edges shared by multiple vehicles. For example, if many vehicles passing the same edge have unexpected longer travel time, it is more likely that the shared edge has a poor traffic condition. Motivated by this, we consolidate ETC records on different paths together and represent the knowledge of the shared edges in a Bayesian graphical model to understand the correlations between ETC records for travel time estimations.

Even our paper focuses on the Origin-Destination (OD) based ETC system to infer the travel time, our framework goes beyond this context. Comparing with existing methods using GPS, our framework only requires origins, destinations and duration between them, which benefits many OD based systems, e.g. subway systems, bus systems, traffic camera systems, etc. More abstractly, we can consider these OD based systems as the sparest case of continuous tracking, which only have the starting location and the end location (i.e., two GPS points). This abstraction can enable more applications in the GPS based systems if the adjacent GPS locations are considered as OD or the sampling rate is low, e.g., real-time speed estimation with extremely sparse GPS sampling from large-scale vehicles, traffic jam detections, long-range travel time estimation, etc. To summarize, the contributions of this paper are as follows:

- Conceptually, we conduct the first study to infer fine-grained travel time in the highway system based on Electronic Toll Collection systems. Compared with previous works, we use existing infrastructures and achieve a full vehicular penetration rate without the GPS devices to avoid the privacy and energy consumption issues. Most importantly, our study is based on a large-scale real-world highway ETC system with 69 highways, 773 toll stations, and 4 million daily transaction records. Such a large-scale infrastructure and data access enable us to discover real-world highway mobility issues, which cannot be achieved by small-scale system or data. We will share such valuable data for the benefit of the research community.
- We design a novel system called SharedEdge to estimate the real-time fine-grained travel time in the highway systems. Specifically, we consolidate multiple ETC records to represent the constraints and correlations of edge travel time in a Bayesian Graphical Model. By formulating travel time estimation on the edge levels as a joint optimization, we apply Expectation Maximization (EM) to solve it in both batch and streaming manners for different situations.
- Based on the model, we implement SharedEdge with real-world ETC infrastructure and data access in the Guangdong province of China, which captures around 4 million daily toll records. To our knowledge, this is the first model implemented based on real-world large-scale ETC infrastructures, achieving a full vehicular penetration rate.
- With a one-month ETC dataset, we evaluate SharedEdge by comparing it to other baseline and state-of-the-art models. In particular, we obtain the ground truth of travel time on edge levels by a large-scale vehicular network with 114 thousand vehicles and 200 million daily GPS records. Compared with the empirical method we reduce error rates by 10% on average. Compared with the probe-vehicle based method, we achieve a similar accuracy with sufficient vehicles and a better accuracy with sparse vehicles.

The rest of the paper is organized as follows. Section 2 introduces our design motivation. Section 3 presents the architecture of SharedEdge. Section 4 describes the data collection infrastructure. Section 5 depicts the path inference component. Section 6 gives the travel time estimation component. Section 7 validates our design with datasets. Section 8 presents a real-world application on the top of SharedEdge modeling, followed by the related work and discussion in Sections 9 and 10. Finally, we conclude the paper in Section 11.

## 2 MOTIVATION

Before we get into the details of SharedEdge, we first make some useful observations that motivate our approach.

### 2.1 Limited Single-Edge Trips

Our goal is to estimate the edge travel time with ETC data. Therefore, the most straightforward method of obtaining accurate edge travel time is to use ETC records of adjacent toll stations that only visit single edges in the trip. Figure 1 shows the single-edge trip volume compared to the total volume in a day. We can see there are only less than 15% of trips traveling on single edges, which shows it is biased to use these records to represent the overall travel time in the edge. Figure 2 shows the edge coverage in the state with only single edge trips in the time slot 5 minutes. Even in the rush hours, the coverage is no more than 20%, which means single edge trips are not sufficient to cover the all the edges. And the single edge trips can only reflect the travel time of a particular vehicle, and it may lead to a bias if only limited single edge trips are used to estimate the generic travel time distribution. There are more vehicles passing the edge without leaving the highway, which leads to large number of trips from many vehicles. Therefore, using all these trips passing the edge gives better estimate of the travel time distribution. Based on these two observations, we argue that it is not feasible to use ETC records of adjacent toll stations to estimate the state scale edge travel time.

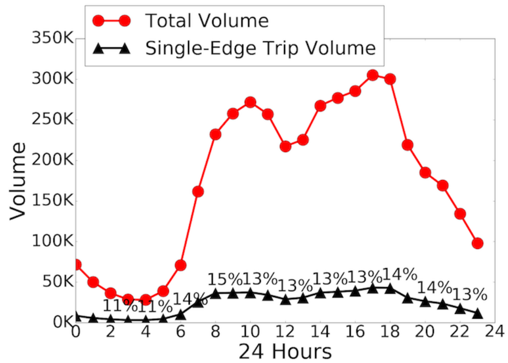


Fig. 1. Low Percentage of Single-Edge Trips

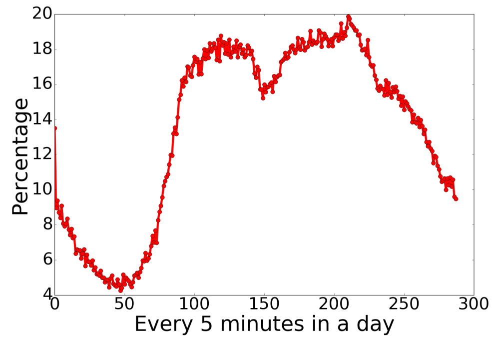


Fig. 2. Percentage of Edge Coverage

## 2.2 Multiple Path Choices

Considering the infeasibility of single-edge records, we leverage records visiting multiple edges. The problem is that there is generally more than one path between the origin and the destination. Figure 3 gives a subset of highway network in Guangzhou, the largest city in Guangdong province. We can see there are four possible paths between the origin and the destination (i.e. $P1$, $P2$, $P3$, $P4$). Among these paths, $P3$ is the shortest path that passes through the downtown area of Guangzhou while $P4$ is the longest path. People can select any of them for traveling. A naive method is to assume that people always choose the shortest path. Figure 4 shows a quantitative result of people's choices at the different time of a day. We can see people prefer to take $P3$ at the none-rush hour to save travel distances while choosing other paths in the rush hour to save travel time. It shows the shortest path assumption is too strong for real-world situations.

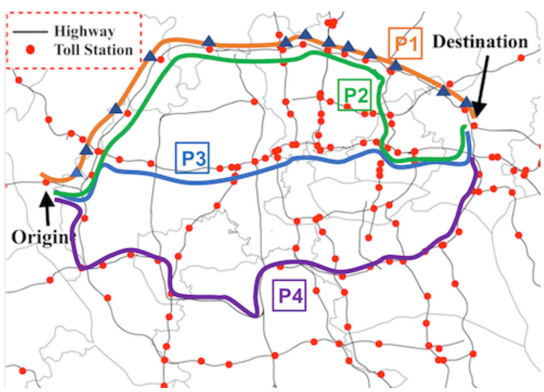


Fig. 3. Edge Travel Time Distribution

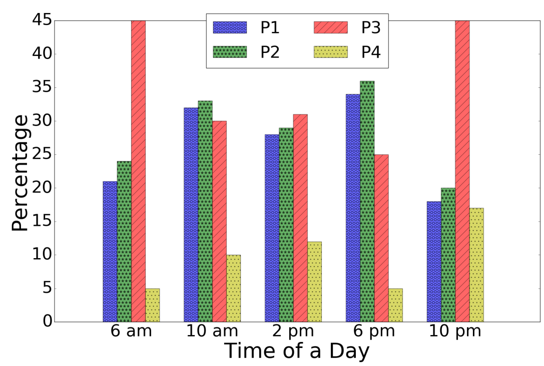


Fig. 4. Path Choices Percentage in Fig 3

## 2.3 Disproportionate Travel Time

Along with the multiple paths, another issue is that an ETC system only gives the total travel time between the origin and the destination rather than the travel time of the individual edges. For example, $P1$ has 14 edges, which are separated by triangle markers in Figure 3. We know that the total travel time over 14 edges rather than that of the single edge. An empirical method is to assume the travel time is proportional to the travel distance and split the time according to the edge length. In other words, it assumes that the average speed in each edge keeps the same. However, it is not true in reality because of the different traffic conditions (e.g. congestion, accidents) on the roads. To justify this, we randomly choose four edges with different lengths in the ETC system and plot the average speed at the different time of a day in Figure 5. We can see the average speed varies a lot at the different time of a day for each edge. At the same time, different edges also show different speeds. Overall, it validates the travel time is not proportional to the travel distance. Therefore, empirical methods are not feasible.

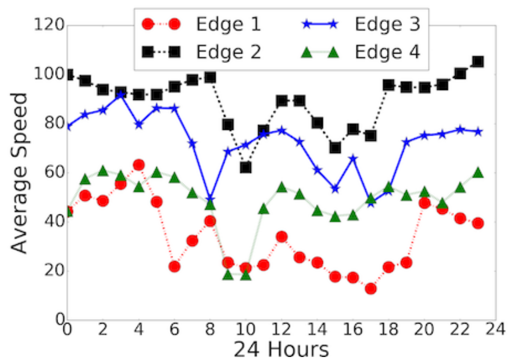


Fig. 5. Average Speed

## 2.4 Summary

These observations lead us to conclude that (i) single edge records are not enough to cover all the edges in the state-level highway systems, (ii) the path choice is not trivial and does not follow the shortest path assumption, and (iii) travel time of each edge is not proportional to the travel distance. These conclusions motivate us to estimate the fine-grained travel time by consolidating multiple ETC records in two steps, which are path inference and time estimation.

# 3 OVERVIEW

In the SharedEdge system, we utilize an existing ETC system as the infrastructure to estimate edge travel time on highways in real time. Without the introduction of any other sensors, our system achieves fine-grained resolutions to support real-world services. Figure 6 shows the three-layer architecture of SharedEdge.

- **Infrastructure Layer:** The physical infrastructure layer provides real-time data feeds from the ETC system as well as stores historical ETC data. The ETC system captures all the vehicles when entering or leaving the highways. It collects information including the origin, the destination, the duration, the vehicle type (i.e. Bus, Truck, Private vehicle), etc. It reflects the real-time traffic conditions on highways. The highway map data depict the highway road structure including the locations of each highway and toll stations. The details are given in section 4.
- **Model Layer:** Based on the ETC data and highway structure provided by the infrastructure layer, the model layer estimates fine-grained edge travel time in real time. There are two steps in modeling. First, we infer the actual travel path on the road map using real-time edge travel time estimation when we receive ETC records. The second step is to estimate the travel time in the visited edges based on the inferred path and travel duration. As a result, the estimated time feeds back to the path inference. These two steps iterate and improve each other for better optimization. The details are given in section 5, section 6 and section 7.
- **Application Layer:** The output of the model layer benefits many applications, which require fine-grained edge travel time such as route planning according to the real-time edge travel time and traffic anomaly detection by monitoring the travel time changes. These applications provide better strategies for drivers

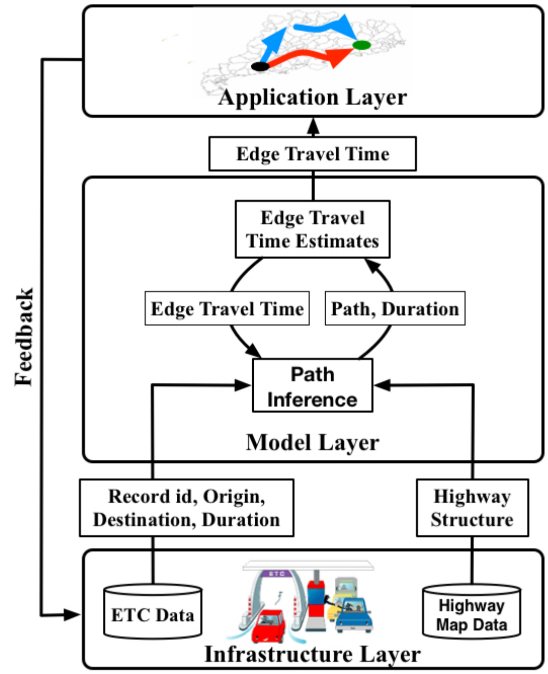


Fig. 6. System Architecture

and administrators, which feed back to the infrastructure layer for optimizing the layer and generating more data. The details are given in section 8.

Based on the three-layer architecture, we use the Guangdong province, one of the most populated and wealthy provinces in China with the total area of 179,800 $km^2$, as a testbed to implement our SharedEdge system. In summary, the infrastructure layer feeds real-time ETC data to the model layer. As each ETC record received, we utilize highway map data to infer the traveling path. Combining multiple ETC records, we assign travel time to each edge and then estimate the edge travel time. The output of the model layer is the real-time edge travel time, which is then fed to the application layer such route planning service. These services benefit both drivers and administration management to improve the efficiency of the infrastructure layer. And our system can also be extended to other contexts (i.e real-time speed estimation, traffic jam event detection). Since our model estimates the travel time on the edge, we can obtain the speed by using the edge length from the map data. For traffic jam event detection, through comparing the estimated travel speed and normal travel speed, we can know which edge is under the condition of congestion, which implies the traffic jam event.

## 4 INFRASTRUCTURE LAYER

In this section, we give a brief description of the ETC system in Guangdong Province, China, which provides real-time highway traffic data feeds. Figure 7 gives an overview of the ETC system in Guangdong province, a network of 69 highways and 773 ETC toll stations covering an area of 179,800 $km^2$. The overall highway road structure is a radial map. The central cities have denser stations, which are connected with surrounding cities by several major highways. Real-time data from an ETC system contain all the enter/leave records from toll stations in the Guangdong province, no matter whether a vehicle pays in cash or uses an electric device. There are totally

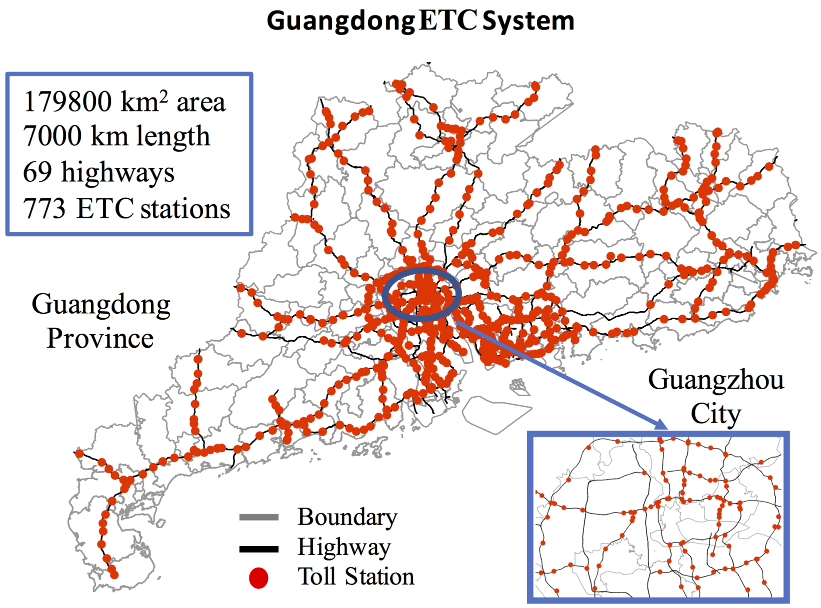


Fig. 7. ETC System in Guangdong Province

more than 4 million transactions every day. Each transaction includes the time, road id, toll station id, vehicle type (i.e. Bus, Truck, Private vehicle), etc. Details are shown in table 1 and used fields are highlighted.

Table 1. ETC Data Description

| Fields | Value |
|---|---|
| **Entrance/Exit Station** | Futian Station |
| **Entering/Exit Time** | 2016-06-01 00:00:51 |
| Plate | F85IS1B4GU |
| Vehicle Type | Bus/Truck/Others |
| Axis Count | 2 |
| Weight | 1500kg |
| Length of Shortest Path | 35km |

To obtain deep into the infrastructure layer, we explore the ETC data from following aspects:

- Edge length: Edge length is an important feature in the system. Figure 8 shows the distribution of edge lengths on highways. More than half of the edges have lengths less than $10km$. Considering that the speed limit is $120km/h$ on highways, the travel time on the $10km$ edge may vary between $5-10$ minutes depending on the actual traffic condition.
- Temporal feature: We choose a typical station and explore the traffic pattern on weekday and weekend. Figure 9 shows the traffic volume changing in 24 hours of a day. Compared with weekend, we can see the traffic on weekday is heavier at 8 am, at noon and at 5 pm because it is the typical beginning and end of the office hour. The weekend traffic is heavier in the mid-night because the nightlife is generally longer on weekend. Considering the difference, we evaluate both time periods in Section 7.

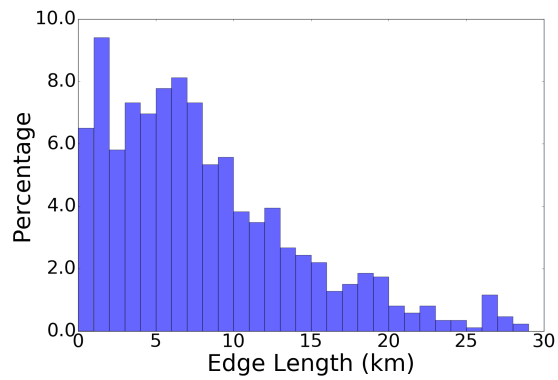


Fig. 8. Edge Length

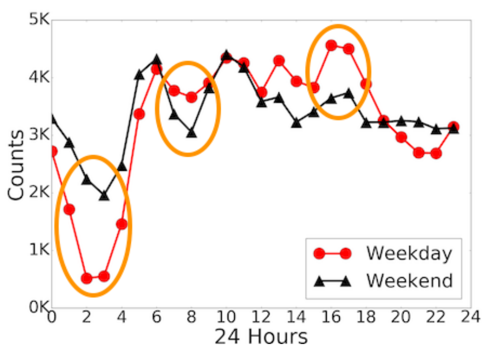


Fig. 9. Traffic Flow on Weekday and Weekend

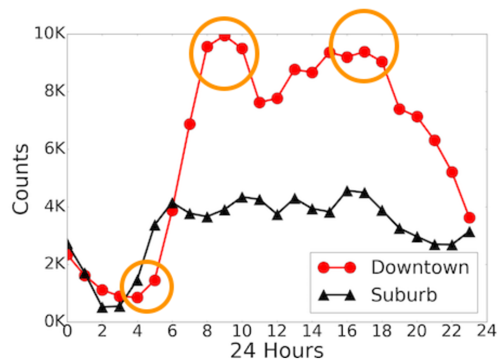


Fig. 10. Traffic Flow in Downtown and Suburb

- Spatial feature: We show the spatial difference by choose two typical stations located at downtown and suburb respectively in Figure 10. We can see the downtown has apparently heavier traffic than that of the suburb. Two peak hours are at 9 am and 6 pm. It also shows the suburb traffic volume starts to increase from 4 am while the downtown traffic volume starts from 5 am, which depict different work and life patterns. Considering the different station density in downtown and suburb, we evaluate how the station density affects the performance in Section 7.
- Travel time: We also study the travel time feature. Shown in Figure 11, the most common trips are less than 30 minutes because of the heavy demands of intra-city commuting (e.g. beltways). Most of the trips are less than 2 hours. Figure 12 shows the travel time proportion in different time of a day. We group the travel time into four clusters (i.e. 0-30 minutes, 30-60 minutes, 60-90 minutes, longer than 90 minutes). We can see short trips dominant the majority in all time. Longer trips (i.e. longer than 90 minutes) increase at midnight.

One of the major strengths of SharedEdge is the rich ETC data in the large scale highway system. With these data, our physical infrastructure layer can achieve large-scale real-time travel time estimation, which is unprecedented in both coverage and immediacy.

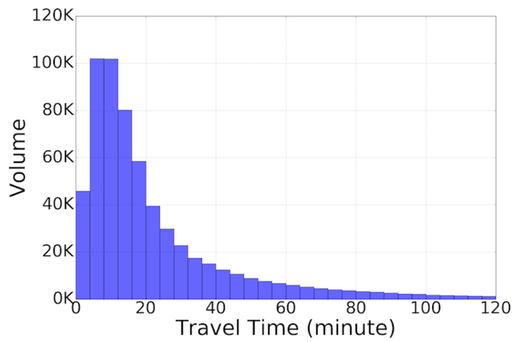


Fig. 11. Travel Time Distribution

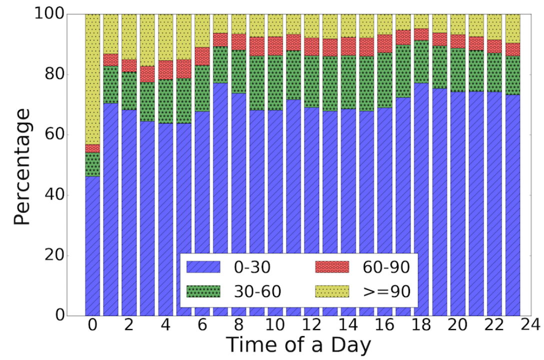


Fig. 12. Travel time proportion in different time of a day

## 5 MODEL LAYER: PATH INFERENCE

As we mentioned, there are generally multiple paths between the origin and the destination. In this section, we infer the path of each ETC record using initialized or estimated real-time travel time.

### 5.1 Preliminary

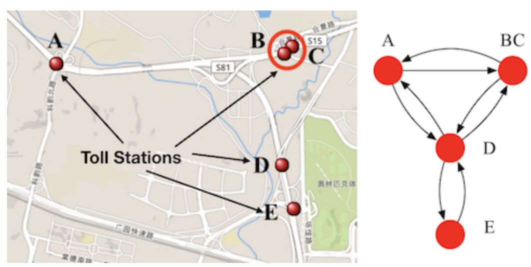


Fig. 13. Highway Network to Graph

To better illustrate our model, we first introduce how we convert the highway network into a graph and give some formal definitions.

- *Highway Network.* A highway network is modeled as a directed graph $G(V, E)$, where $V$ refers to the set of vertices(i.e., toll stations) and $E$ refers to the set of edges(i.e., road segments between adjacent toll stations). We consider two stations as one vertex if they are the different gates of the same station. And the edge exists if there is a road connection between two stations without any other intermediate stations. For example in Figure 13, $B$ and $C$ are closed stations so we merge them into one vertex $BC$. The direct road connections between stations are converted into edges. We will use *edge* to represent the road segment in the following parts.
- *Path.* A *path* $P$ is a list of adjacent edges, where each two consecutive edges are connected in $G$. We use $P_O$ and $P_D$ to denote the origin and the destination station respectively.
- *Trajectory. Trajectory* is a list of pairs, which consist of locations and time stamps.

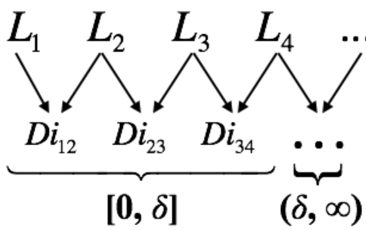


Fig. 14. Path Space Pruning

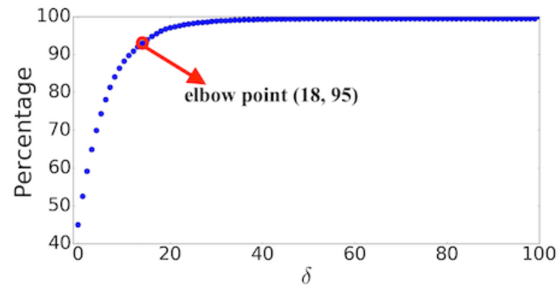


Fig. 15. Coverage changing with $\delta$

### 5.2 Pruning

To infer the path, we first need to figure out all the path candidates. Theoretically, there are infinite paths between the origin and the destination considering the complex highway graph. It would be computationally expensive if we take all the possible paths into consideration. To reduce the search space, we assume that all paths are *simple paths* that one edge can only appear at most once and people only choose the shortest path or paths that are reasonably longer than the shortest. Figure 14 gives an example of the method. We first search *simple paths* between the origin and the destination using searching algorithms (i.e. Depth First Searching Algorithm) and sort them by the length in an ascending order $(L_1, L_2, ...)$. For adjacent lengths, we compute the increasing percentage by $Di_{ij} = (L_j - L_i)/L_i$ where $j = i + 1$. Then we define a threshold $\delta$ to prune the paths that we only keep paths with lengths $L_i$ whose $Di_{ij} < \delta$. We can see the higher value of $\delta$, the higher coverage of possible paths. As $\delta$ increases to the extremely large, all the paths can be covered. However, the higher coverage brings the issue of computing cost for exploring multiple paths. It is a tradeoff between accuracy and performance. To find a reasonable $\delta$, we perform a survey in Guangdong on origins and destinations with multiple paths. Figure 15 shows how the coverage changes with the value of $\delta$. we can see as the value of $\delta$ becomes greater than the elbow point, the coverage percentage does not gain much additional coverage. Therefore, we choose the elbow point with the $\delta$ value 18%.

### 5.3 Initialization

Our final goal is to estimate the travel time distribution on each edge. So for the first time, we do not have the real-time traffic information on each edge. To complete the path inference, we first initialize the travel time on all the edges using historical ETC data. Specifically, for all the edges in the highway system, we assign different weights to each edge according to their length. For each historical ETC record, we assume they travel on the shortest path and split the travel time into a set of samples by the weights. The samples from multiple ETC records are shuffled and those of the same edge are grouped together as samples. Then we compute the mean and standard deviation of the samples to approximate those of the distribution. Even there are some inaccuracies compared with the true distribution, our method can iteratively update the distribution as more real-time ETC records are received. The details of updating process are given in section 6.

### 5.4 Inference

Since the total travel time is the sum of the travel time over each edge, we introduce a new variable $Z$ defined as

$$Z = \sum_{i=1}^{n} X_i \tag{1}$$

where $X_i$ is the travel time on the edge $e_i$. For each path candidate, we have a random variable $Z$. Therefore, our goal is to obtain the $Z$, which is mostly probably to produce the total travel time $D$. Since $X_i$ are concatenated and affect each other, we cannot treat them independently to obtain the exact distribution of $Z$. Here, we choose sampling to simulate the distribution of Z. For a path with $n$ edges, we sampling each $X_i$ in the path for $m$ times so there are $m$ samples that each one consists of $n$ values. For each of the sample, we compute the sum by equation(1). To show which path has higher probability, we compute the Mean Absolute Percentage Error (MAPE) of each path by

$$MAPE = \frac{100}{m} \sum_{t=1}^{m} \frac{|Z_t - D|}{D} \tag{2}$$

where $Z_t$ is the sum of the sample $t$. After the above process, we can obtain an MAPE for each of the path and the one with the minimum error is our inferred path.

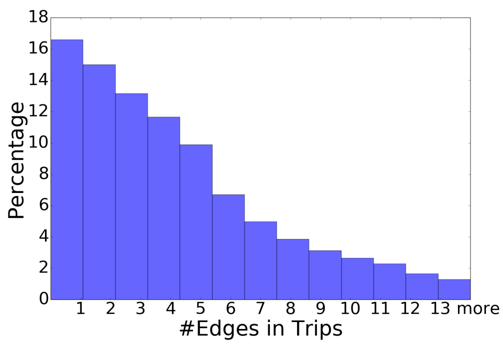


Fig. 16. Path Length (#edges)

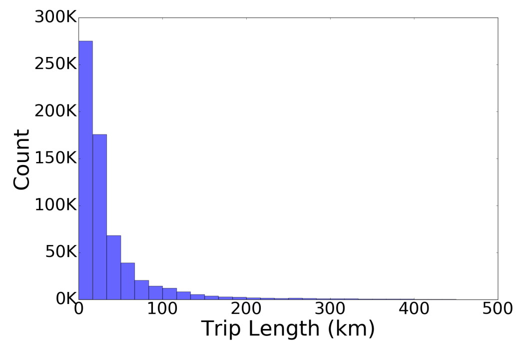


Fig. 17. Trip Distance (km)

Fig. 16 gives the statistics of inferred path length and Fig. 17 gives the distance of the trips. We can see the multiple-edge trips are 84%, which is much larger than the single-edge trips.

## 5.5 Summary

In summary, we utilize the real-time edge travel time information and the highway map structure to infer the travel path on highways. In short, we first search for all the possible simple paths in the highway graph between the origin and the destination. Based on the assumption of the possible paths, we prune the path candidates by a survey result, which define the threshold $\delta$. Then we compute the MAPE of each candidate, which is used as the criterion to measure the probability of each path. The one with the minimum MAPE is chosen as our inferred path.

# 6 MODEL LAYER: EDGE TRAVEL TIME ESTIMATION

After the path inference, we need to figure out the travel time on each edge of the path. A single ETC record cannot accurately determine the travel time of a single edge because there are multiple ways to split and assign time to the edge. In the following section, we present a travel time estimation approach by consolidating multiple ETC records.

## 6.1 Model Representation

Estimating the travel time distribution of edges is difficult by the factor that we cannot observe the travel time of individual edges. Instead, we observe the sum of travel time over edges in a path. We first study those ETC records between adjacent toll stations to observe the statistic features of the travel time in each edge. In the previous studies [10] [24], people assume a normal distribution for the travel time. We check the normality in Figure 18 and Figure 19. It shows the edge travel time distribution and corresponding Q-Q plot of a typical edge in one hour. We can see that the edge travel time follows a normal distribution. This observation motivates us to model the edge travel time as a normal distribution.

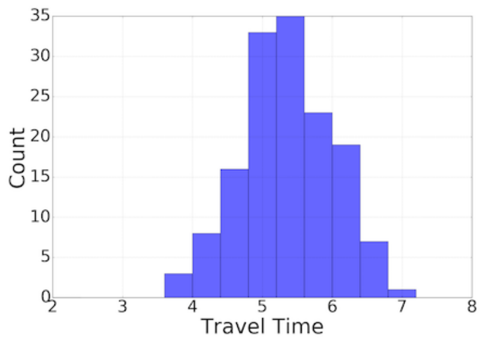


Fig. 18. Edge Travel Time Distribution

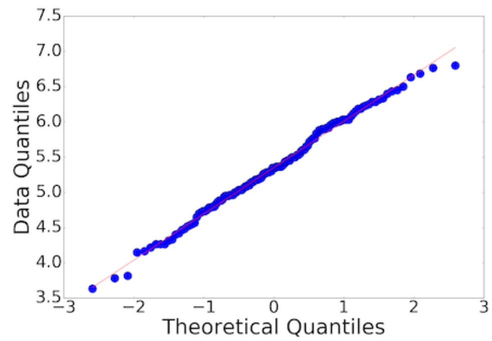


Fig. 19. Q-Q Plot of Edge Travel Time

Considering one ETC record $r$ with the inferred path $P$, the observed travel time can be represented as

$$D = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}^T \times \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} \tag{3}$$

where $D$ is the total travel time, $X_i$ is the travel time on the edge $e_i$ and $\alpha_i$ is an indicator function defined as

$$\alpha_i = \begin{cases} 1 & \text{if edge } e_i \text{ is visited} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

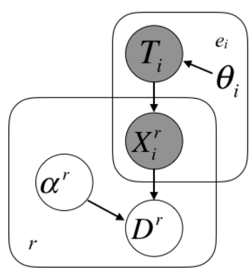


Fig. 20. A subset of the Bayesian Graphical Model. Grey nodes are hidden variables and white nodes are observations. Two boxes are the domain of the ETC record and the edge perceptively.

In our setting, $D$ is obtained from real-time ETC records and $\alpha_i$ can be obtained with inferred path in section 5. These two variables are accessible and regarded as observations. For each path, there are a set of hidden variables $X_i$ and their distribution $T_i$, which can be parametrized as some parameter vector $\theta_i$. We model the dependencies between the observations and hidden variables as a Bayesian Graphical Model shown in Figure 20. The problem can be formulated as a learning problem of estimating $T_i$. The set of parameters that maximize the likelihood of these observations is the solution to the maximum likelihood problem

$$\max_\theta \sum_r \log \pi(D^r|\alpha^r;\theta)$$

where $D^r$ is the travel time of the ETC record $r$. $\pi(D^r|\alpha^r;\theta)$ is the probability of observing the travel time $D^r$ given the inferred path represented as $\alpha^r$ under the parameters $\theta$. Combining with equation(3) and (4), we have

$$\begin{aligned} &\pi(D^r|\alpha^r;\theta) \\ &= \int_X \pi(D^r|X,\alpha^r)\pi(X;\theta)dX \\ &= \int_X \pi(D^r|X,\alpha^r)\pi(X;\theta)\times(\prod_{i:\alpha_i=1}\pi(X_i;\theta)dX) \end{aligned} \tag{5}$$

To solve this maximum likelihood problem, we apply EM algorithm [9] to solve it. Figure 21 shows the workflow of the EM algorithm.

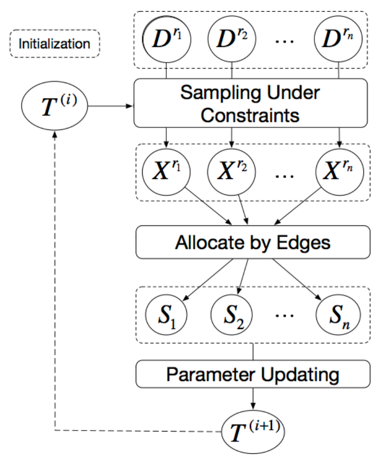


Fig. 21. EM Algorithm Workflow

- In the initialization step, we initialize the travel time distribution $T^{(0)}$ on all the edges using historical ETC data. Specifically, for each ETC record, we assign weights to each visited edge according to the edge length then split the travel time $D^r$ into a set of samples by the weights. The samples from multiple ETC records are shuffled and those of the same edge $e_i$ are grouped together into $S_i$. Then we compute the mean and standard deviation of the samples to approximate those of the distribution.
- In the E-step, we generate samples $X^r$ from each edge on the path of each ETC record according to the current edge travel time distribution. Each sample contains two elements, the travel time($x_i^r$) and the corresponding probability($w_i^r$) in the distribution. The reason we use weighted samples rather than plain samples is that it provides smoothing updating between iterations. The sampling process obeys the constraint described in equation (3). This involves a statistic problem of sampling from Multinormal distribution under constraints, which we solve in section 6.2.
- In the M-step, we group the samples of the same edge together into $S_i$ and update the current distribution. Individual samples may be insufficient to estimate the distribution so we use the overlapping of multiple

trips to estimate the real distributions. Here we introduce two different manners described in section 6.3. Then the newly updated parameters are passed to the next iteration as an online fashion.

### 6.2 Sampling Under Constraints

For each ETC record, we can only know the total travel time and the inferred path without information on each edge. Theoretically, there are infinite ways to split the time and assign to each edge. As we mention in section 2, the travel time is not proportional to the travel distance. Therefore, we utilize the statistic sampling technique to solve the problem. The goal is to sample from the visited edges with the current distribution under the constraint that the sum of samples is equal to $D^r$. Mathematically, it can be presented as

$$\begin{aligned} &Y(\mu, \Sigma) = \begin{bmatrix} X_1 & X_2 & \dots & X_n \end{bmatrix}^T \\ &\text{subject to } \alpha x = D \end{aligned} \tag{6}$$

where $x$ are the samples from Y with mean $\mu$ and covariance matrix $\Sigma$, $D$ is the travel time and $\alpha$ is a vector of elements defined in equation(3).

**ALGORITHM 1:** Sampler for Multinormal distributions under constraints on a hyperplane

- Sample $y \sim Y(\mu, \Sigma)$
- Return $x = y + \Sigma\alpha^T(\alpha\Sigma\alpha^T)^{-1}(D - \alpha y)$, which can be realized using
  - Solve $\varphi$ such that $(\alpha\Sigma\alpha^T)\varphi = D - \alpha y$;
  - Return $x = y + \Sigma\alpha^T\varphi$.

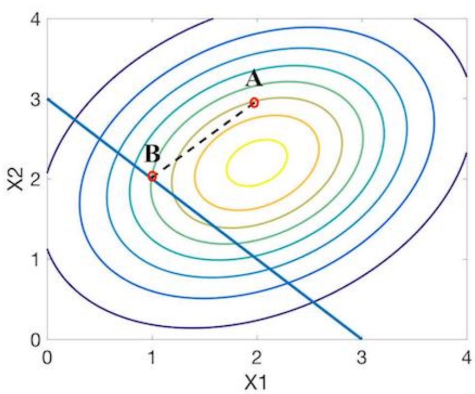


Fig. 22. A Two Dimensional Sampling Example

Statistically, we can consider the formula (6) as a n-dimensional Multinormal distribution $Y$ truncated on the intersection of the hyperplane. Figure 22 takes $n = 2$ as a simple example. The line is the hyperplane constraint. Therefore the target is to draw $x$ from unconstrained $Y$ and map it to the hyperplane. To achieve this, we apply the simulation algorithm proposed in [7]. The simulation process is presented in Algorithm 1 and the detailed proofs can be found from [7]. The only unknown parameter here is $\Sigma$ since it is difficult to directly observe the covariance matrix. To solve this, we approximate it as the sample covariance. We sample simultaneously from each edge for $m$ times and then compute each item $q_{jk}$ of the covariance matrix by

$$q_{jk} = \frac{1}{m}\sum_{i=1}^{m}(x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k) \tag{7}$$

where $x_{ij}$ is the $ith$ sample from $X_j$ and $\bar{x}_j$ is the mean of samples.

## 6.3 Parameter Updating

We introduce two manners to update the travel time distribution for the needs of different real-time situations.

*6.3.1 Batch Manner.* In the batch manner, the updating happens when all the data in a time slot are received. The advantage of the batch manner is that the updating is based on the data in a longer period, which is more representative than a single record. Here we model this problem as a maximum likelihood problem. Given samples on one edge $e_i$ from $n$ ETC records in the format $(x_i^r, w_i^r)$, we have the log likelihood function

$$\begin{aligned} &ll(S;\mu,\sigma^2) \\ &= \ln \prod_{j=1}^{n} w_i^{r_j} \pi(x_i^{r_j};\mu,\sigma^2) \\ &= \ln(\prod_{j=1}^{n} w_i^{r_j} (2\pi\sigma^2)^{-1/2} exp(-\frac{(x_i^{r_j}-\mu)^2}{2\sigma^2})) \\ &= \sum_{j=1}^{n} \ln w_i^{r_j} - \frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i^{r_j}-\mu)^2 \end{aligned} \tag{8}$$

where $S$ is the set of samples on the same edge. The first order conditions for the function are

$$\frac{\partial}{\partial\mu} ll(S;\mu,\sigma^2) = 0 \qquad \frac{\partial}{\partial\sigma^2} ll(S;\mu,\sigma^2) = 0$$

Then we have the estimators for $\mu$ and $\sigma^2$

$$\hat{\mu} = \frac{1}{n}\sum_{j=1}^{n} x_i^{r_j} \qquad \hat{\sigma}^2 = \frac{1}{n}\sum_{j=1}^{n}(x_i^{r_j} - \hat{\mu})^2$$

The updated parameters are then used as new parameters for the next batch.

*6.3.2 Streaming Manner.* In the streaming manner, we consider data arriving one by one. The parameters are updated when a new observation is received. The advantage of the streaming manner is that it can give the real-time feedback from the data and does not need extra space to store all the data. The key of the streaming manner is to leverage the new data while maintaining the numerical stability. To achieve this, we apply a sliding window method. For each edge, we maintain a queue with length $k$ (e.g. 100) to hold incoming samples. When the new sample comes and the queue is full, we remove the earliest sample to make room for the new sample. To update the mean and variance, we apply the method proposed in [29] on the samples in the queue by

$$\hat{\mu} = \frac{\sum_{i=1}^{n} w_e^i x_e^i}{\sum_{i=1}^{n} w_e^i} \qquad \hat{\sigma}^2 = \frac{\sum_{i=1}^{n} w_e^i (x_e^i - \hat{\mu})^2}{\frac{n-1}{n}\sum_{i=1}^{n} w_e^i}$$

This method can efficiently update the mean and variance of weighted sampled data when an additional data value is included in the set. Then the updated parameters are used as initialized value for the next record.

### 6.4 A Special Edge

Besides the regular edges on highways, there is a special case to be considered while a vehicle passes toll stations. Unlike the general road structure in cities shown in [13], there is a short length of road called ramp, which connects the station to the main road of highways. ~~Figure 19 shows an example of the ramp.~~ The key difference is that vehicles on the near main road do not use ramps unless they enter or leave the highway from the corresponding station. In this case, estimating the travel time on ramps is another story.

In order to deal with the ramp travel time, we take the ramp travel time out as an individual estimation part. To simplify the model, we estimate the ramp travel time based on two factors, the length of the ramp and the length of the waiting queue in the toll station. For the ramp length, we can obtain it from the highway map data. For the length of the waiting queue, we approximate it as the number of vehicles passing the station, which is accessible from the ETC data. Especially, the waiting queue is considered to be empty on the ramp of the origin station since the travel times only cover the period after passing the stations. Then we apply a linear regression method to model the travel time on the ramps based on these two factors.

### 6.5 Summary

Based on the constraints of multiple ETC records on the shared single edge, we estimate the edge travel time in real time. In short, with constraint sampling on ETC records, we obtain the travel time samples of each edge and then group them together to estimate the travel time distribution. With parameter updating in two manners, we can balance the trade-off between accuracy and immediacy.

## 7 MODEL LAYER: IMPLEMENTATION AND EVALUATION

In this section, we first introduce the implementation of SharedEdge based on the data from Guangdong Province. Then we present the evaluation based on the probe vehicles data and a field study.

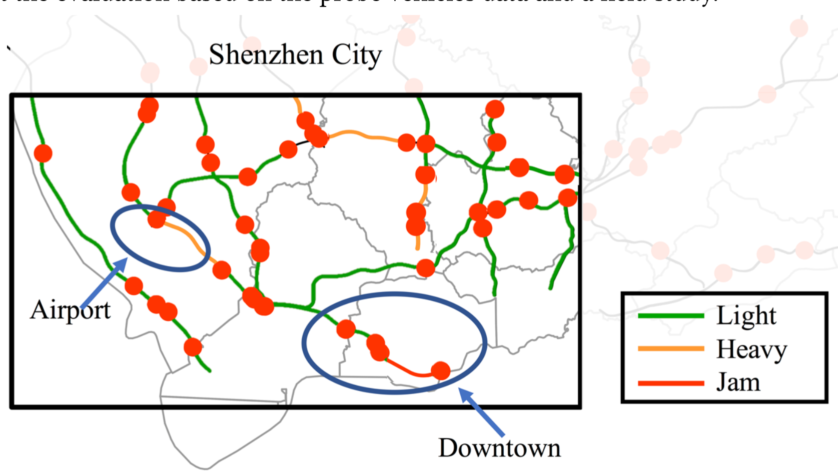


Fig. 23. Traffic map of Shenzhen city at 6 pm

### 7.1 SharedEdge Implementation

We implement our SharedEdge based on the data corollected from the ETC system of Guangdong province. Specifically, we utilize ETC transactions in January 2016, where 3-week data are used as historical data and 1-week data are used as real-time data. For the offline work such as statistic analyses, we utilize a Spark cluster consisting of 5 nodes, and each of them is equipped with 16 cores and 128 GB RAM. For the online estimation, we utilize a single core of a server with 3.00GHz Xeon CPU and 32GB RAM. For a real-time system, computing feasibility is always a important concern. In our system, the most expensive process happens in the stage of offline (i.e. initializing the distributions of all the edges), which costs totally 1.3 minutes mainly because of loading the large historical dataset. Since it is offline behavior, it does not affect the online performance of the system. In the online stage, the average processing speed is 400 transactions every second. From Fig. 1 we know the highest volume of a day is 300 thousand transactions in one hour, which is around 83 transactions every second. Therefore, our system is capable of satisfying real-time computing feasibility.

Considering the extremely large size of the data, we found several kinds of errant data including duplicated data, missing value and outliers. In order to address these issues, we conduct a detailed cleaning process to filter out errant data. For the map information, we use OpenStreetMap[4] to obtain the highway road structure. Because of some missing highways in OpenStreetMap, we incorporate highway information from Gaode Map [2] as complementary.

The output of our SharedEdge is the edge travel time. We present the output as a traffic map, where different colors of highways represent different levels of congestion. Figure 23 shows a part of the traffic map located in the main area of Shenzhen city, the second largest city in Guangdong Province. We can see the airport and downtown area have obviously congested traffic.

## 7.2 SharedEdge Evaluation

*7.2.1 Dataset.* To evaluate the performance of our model, we introduce another dataset of probe vehicles (shown in Table 2). We leverage commercial vehicles and private vehicles to collect real-time trajectories from both types of vehicles: 14 thousand commercial vehicles and 100 thousand private vehicles. These vehicles upload their real-time locations every 10-15 seconds.

Table 2. Evaluation Dataset Description

| Commercial Vehicles | | Private Vehicles | |
|---|---|---|---|
| Time | January, 2016 | Time | January, 2016 |
| vehicles | 14,000 | vehicles | 100,000 |
| records | 410 million | records | 430 million |
| Format<br>Plate, Date&Time, GPS | | | |

*7.2.2 Ground Truth:* To extract the ground truth of the travel time, we first utilize the map matching algorithm [18] to obtain the route in the form of a sequence of edges. Then the travel time in each edge can be approximated by $t_e - t_s$ where $t_s$ and $t_e$ are the start time and the end time of the trajectories on the edge. The reason that we called it an approximation is that the vehicle may not be exactly at the ends of the edge at time $t_s$ and $t_e$ because of the uploading interval around 10 to 15 seconds. Considering the edge travel time is generally more than 10 minutes, it is reasonable to make the approximation. With different travel time on the edge, we use the average travel time as our ground truth. We choose 6 am to midnight as our evaluation period since we can see from Figure 1 that other time periods do not have high traffic volumes so people can drive as fast as possible under the speed limitation. Since the probe vehicles cannot provide full spatial-temporal coverage, we choose those edges and time slots with enough data as test cases.

*7.2.3 Metrics:* Considering the output of our model is the distribution of the edge travel time, we conduct both the statistic test and quantitative test.

- For the statistic test, we conduct Kolmogorov-Smirnov Goodness-of-Fit test [25]. We use ground-truth in each edge as samples to test the null hypothesis:

$$H_0 : \text{samples are sampled from the estimated distribution.}$$

Then at each time, we have the percentage of edges that fail to reject the null hypothesis $H_0$, which shows the overall fitness of our model. For parameters setting, we choose 0.05 as significance level, which is commonly used in tests.

- For the quantitative test, we study the Mean Absolute Error (MAE) (in minutes) and Mean Relative Error (MRE) of the estimated mean with ground truth, which are defined as

$$MAE = \frac{\sum_{i=1}^{m} |y_{e_i} - \bar{y}_{e_i}|}{m}$$
$$MRE = \frac{\sum_{i=1}^{m} |y_{e_i} - \bar{y}_{e_i}|}{\sum_{i=1}^{m} \bar{y}_{e_i}}$$

where $y_{e_i}$ is the estimated mean of the edge $e_i$ and $\bar{y}_{e_i}$ is the ground truth.

*7.2.4 Baselines:* For quantitative comparison, we choose the following baselines:

- Edge-Length-Weighted (ELW) based method: We assume all the vehicles use the shortest path and assign weights to each edge according to their length. For each ETC record, we split the total time into pieces according to the weights of visited edges. We use the mean and standard deviation of the samples to approximate those of the edge distribution.
- Probe-Vehicle (PV) based method: We choose [8] as another baseline. Since we also use probe vehicle as ground truth, we split the probe vehicle data into two parts, 2/3 as training data and 1/3 as test data.

*7.2.5 Results.*

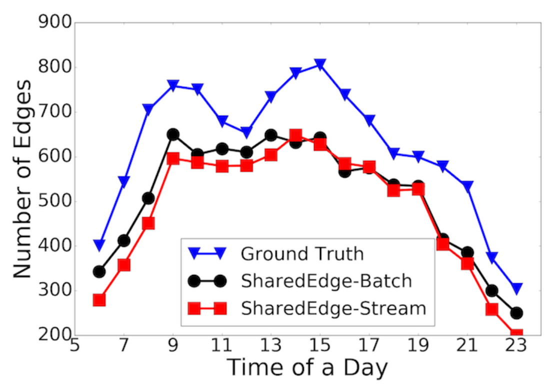


Fig. 24. Goodness of Fit

- **Goodness-of-Fit:** For Goodness-of-Fit test, we apply Kolmogorov-Smirnov test to validate if our estimated distributions are consistent with the ground truth. The ground truth is the number of edges that are visited by probe vehicles. We test our hypothesis $H_0$ on all the visited edges in both the streaming manner and batch manner to see how many edges accept $H_0$.
Figure 24 shows the overall fitness of our model at the different time of a day. The SharedEdge-Batch and SharedEdge-Stream are the numbers of edges that accept our hypothesis, which means the estimated distributions are statistically correct. The lowest percentage of correctly estimated edges is around 71%, which happens in the early morning and late night. The high percentage is around 92%, which happens at

11 am and 7 pm. This result is self-explanatory since the low-percentage time corresponds to the none-rush hour and the high-percentage time corresponds to the rush hour. It shows high-volume ETC records can improve the performance compared to the low volume.
Figure 24 also gives the comparison between the streaming manner and batch manner. It shows these two manners are close in the different time of a day. It means we can utilize less computing resource using the streaming manner with limited loss of accuracy. We can see the performances are almost the same in the afternoon rush hour. It shows frequent updating of the parameters achieves a similar performance as the batch manner.

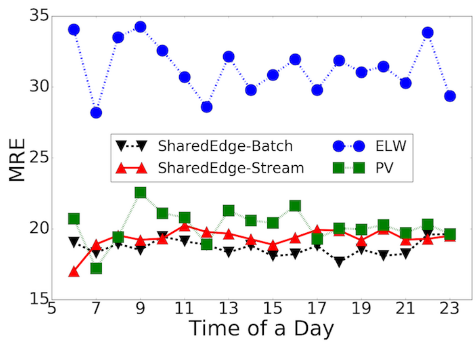


Fig. 25. MRE changing over time of day (Weekday)

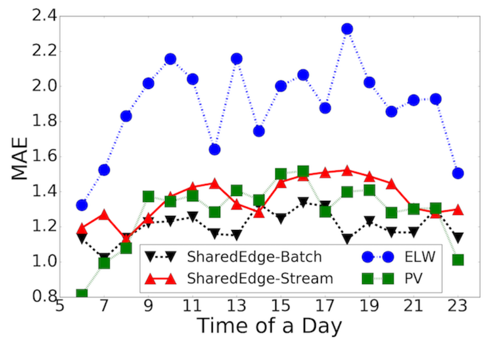


Fig. 26. MAE changing over time of day (Weekday)

- **Accuracy:** For the accuracy test, we compare the performance of our method with other baselines in terms of MRE and MAE. We choose several parameters including weekday, weekend, path length, station density and data scale to evaluate the performance under different contexts.
  **Weekday:** Figure 25 and Figure 26 show the MRE and MAE changing over time of a day on the weekday. From Figure 25 we can see both SharedEdge and PV outperform ELW by 10% on average, which proves the unreliability of the empirical method. Comparing SharedEdge with PV, we can see SharedEdge has similar performance in most of the time. It shows that the travel time estimation can be achieved with the same level of accuracy without using any probe vehicles. Because of the high volume and coverage of the highway traffic flow, SharedEdge has a higher coverage that cannot be achieved with sparse probe vehicles. From Figure 26 we can see the absolute value of the error is around 1.2 minutes, which is acceptable on highways.

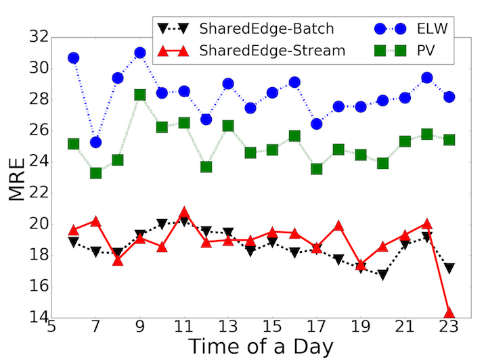


Fig. 27. MRE changing over time of day (Weekend)

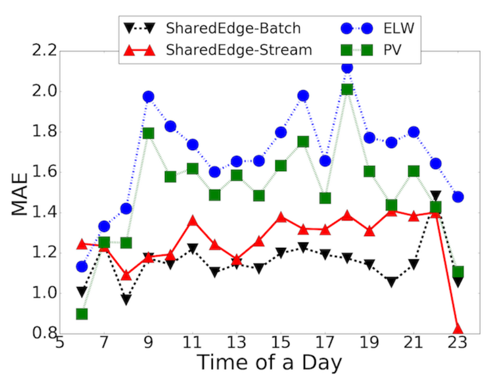


Fig. 28. MAE changing over time of day (Weekend)

**Weekend:** Figure 27 and Figure 28 show the MRE and MAE changing over time of a day on weekend. Compared with ELW, our method outperforms it by 8% on average. As most of the people do not drive to work on weekends, we observe a different pattern compared of the PV method. Different from weekday, the PV method has a significant loss of accuracy, which is 6% less compared with our method. The estimated result of the PV method is biased because the decreased number of probe vehicles cannot represent the whole population. Our method keeps stable since we cover all the vehicles running on highways, which are representative of the population.

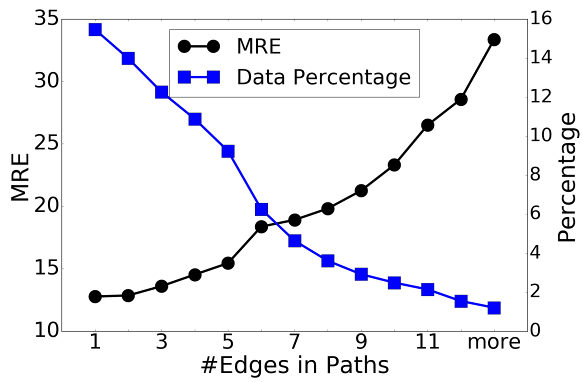


Fig. 29. Performance changing over path length

**Path Length:** Path length (number of edges on the path) is also an important factor that affect the performance of the model since paths with less edges produce better estimation. Figure 29 also validates the hypothesis by showing how the performance changes with the number of edges on the path. The right y-axis is the data percentage of the specific path length over the whole data. We can see that when the number of edges is less than eight, the MRE is below 20%. One interesting observation is that the percentage of the data also affects the accuracy. From the number of edges of 5 to 6, with the rapid decrease of the percentage, the error also has rapid increase. And the accuracy is not good enough if we only use long-hop trips. This is mainly because of the limited of ETC data, which only capture the end to end travel time. However, the travel time for most edges is estimated by both short-hop trips and long-hop trips, and short-hop trips cover a high percentage among all the trips and enable highly accurate travel time estimation. Therefore, for travel time estimated by both long-hop and short-hop trips, the performance is promising.

**Station Density:** We also evaluate how the density of stations affect the model performance. Figure 30 gives the performance changing over the station density. We randomly sample toll stations from 10% to 100% and use transactions only from sampled stations to estimate the travel time. We can see the model can achieve a stable performance after the station density reaches 70%. This is because the majority of traffic concentrates on a few stations, which affects the penetration of the vehicles.

**Data Scale:** Since our system benefits from the high volume of the traffic flow on highways, we also test how the traffic volume affects our system. We randomly sample ETC data at each time from 10% to 100% and compute the average MRE and MAE of a day. Figure 31 shows the result changes as the data size increasing. We can see the MRE is 35% with 10% of the data. As the data size increases, the accuracy also increases. When the data size is greater than 60%, the accuracy approaches the plateau around 19%. This means our system can still achieve high accuracy without the complete data.

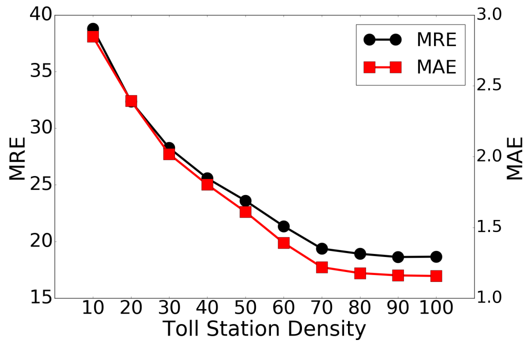


Fig. 30. Performance changing over density of stations

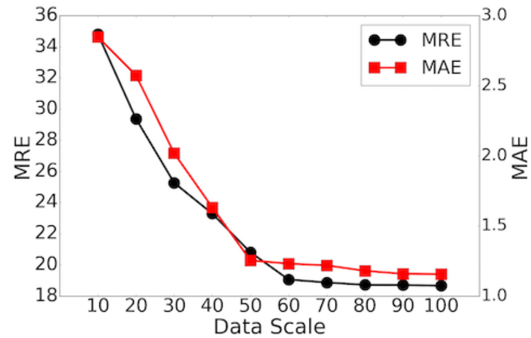


Fig. 31. Performance changing over data scale

- **Estimation Visualization:** We also qualitatively show our result with estimation in Shenzhen City comparing with the ground truth in different time of a day in Fig. 32. It shows that our system can detect all the jam traffic conditions. For heavy traffic, it can detect most of them with some shifting between adjacent edges sometimes (Fig. 32 (a)). The overall performance is good enough for the daily guide of traffic conditions.

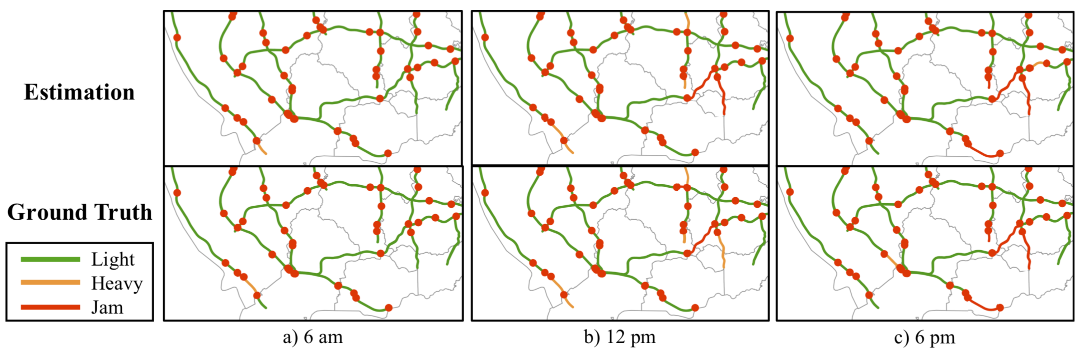


Fig. 32. Estimation Visualization

*7.2.6 Summary.* In the evaluation section, we test our method from both statistic fitness aspect and quantitative aspect. For the goodness-of-fit, our method can statistically fit 90% of edges on average. Compared with the empirical method ELW, our method outperforms it by 10% on both weekday and weekend. Compared with the probe-vehicle based method PV, our method can achieve similar accuracy when the vehicles are sufficient and outperform it by 6% when the vehicles are sparse. Another observation is that the batch manner and streaming manner can achieve similar accuracy, which validates the feasibility of streaming manner in the real-world situations.

## 8 APPLICATION LAYER: ROUTE SUGGESTION

### 8.1 Background

The highway system is one of the most important services for transportations because of its high-speed limitation, no traffic signals and high throughput. It brings significant improvement in shortening travel time and millions of vehicles using highways every day. Considering the complex highway network and unexpected traffic conditions, planning an optimal route is essential for efficient transportation. As we discuss in section 2, the real travel time

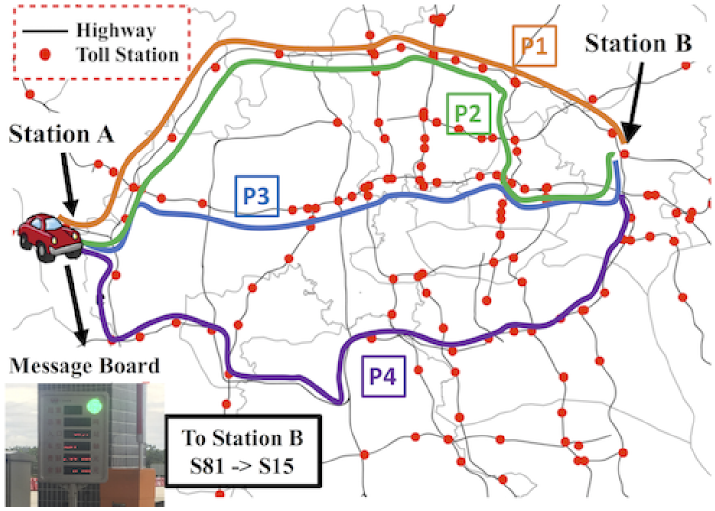


Fig. 33. Route Suggestion System

is not proportional to the travel distance. It leads to the fact that the shortest path is not always the fastest one. Therefore, an optimal route planning depends on the real-time traffic conditions.

Current route planning services like Google Map [3] crowdsourcing all the user locations to estimate the traffic condition on the road. These services require collecting GPS locations continuously, which leaks users privacy and affects the battery life of smartphones. With our system SharedEdge, we can estimate the travel time of each edge in real time without collecting real-time GPS locations. It brings a great flexibility to infer the travel time of any path by concatenating the edges together. One factor is that we cannot get real time updates if we only use the vehicles whose destinations are far away from the traffic jam road segments. However, considering the heavy volume of vehicles on the highways, there are plenty of the vehicles that exit the highways near the traffic jam segments. With those vehicles, we can get immediate updates of the traffic conditions. Those vehicles that exit the highways far away from the jam segments would be given only limited weights in our model compared to those exits the highways nearby. Based on SharedEdge, we demonstrate one case study with a Route Suggestion system (RS) that suggests fastest routes for highway users with only ETC data. Figure 33 shows a demo of our application. As a drive arrives at the entering toll station, our system firstly retrieves its historical records to infer the destination. The reference [27] shows individual vehicles' mobility in highway system is far from random and highly dependent on its historical trajectory. Based the origin and the destination, we use Dijkstra's algorithm to find the fastest path based on the inferred edge travel time. Then we present the recommended route on the message board in the toll station for users (e.g. first take S81 highway then go to S15 highway).

The key advantage of the RS system is that we do not need to collect vehicle real-time locations. Compared with other navigation systems like Google Map [3], which need crowdsourcing user real-time locations to estimate the traffic condition, our application protects users privacy and avoid possible issues with smartphones (i.e. dead battery, weak signal, etc.).

## 8.2 Evaluation

We evaluate the application in terms of the efficiency of the route suggestion result. In the common scenario, if there is no route suggestion application, the vehicle's actual travel time (ATT) is captured by our existing dataset. This time is defined as the baseline. We are interested in how much time the drivers can save using our RS system compared to the baseline. To test this, we assume users follow the system suggestion and synthesize the travel

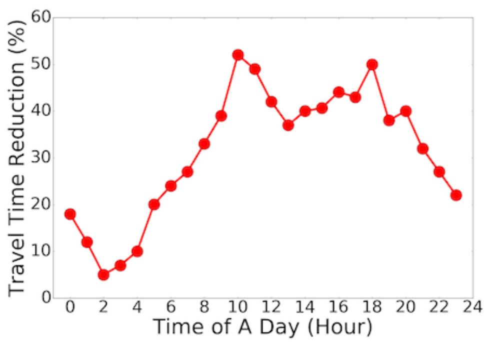


Fig. 34. TTR in A Day


Fig. 35. Example of a test scenario. (a) is an ETC station; (b) is a rest area; (c) is a sample trajectory

time (STT) when the RS system works. We defined the evaluation metric travel time reduction (TTR) as follows:

$$TTR = \frac{\sum_{i=1}^{m} ATT_i - STT_i}{\sum_{i=1}^{m} ATT_i}$$

to represent the ratio of reduced travel time if users follow the suggestion of our RS system.

Figure 34 shows the changing of TTR in different time of a day. It shows with the introduction of the RS system, the travel time is reduced. Especially, the reduction rate can be around 50% in the morning and evening rush hour. Figure 34 is also consistent with the traffic flow changing in a day shown in Figure 10. In the heavy traffic time period, RS system can achieve better result since it avoids the congested edge. In the none-rush period, the shortest path generally has shortest travel time.

Besides using GPS trajectories from the dataset, we also send our own drivers carrying a GPS logger to test the accuracy of our application. The trips mainly focus on commuting among three cities including Guangzhou, Shenzhen and Shanwei. The trip lengths range from 150km to 300km. The corresponding travel time ranges from 2 hours to 5 hours. Figure 35 shows an example of the test scenario where (a) is an ETC toll station and (b) is a rest area in the highway. Figure 35 (c) is the screen shot of a logging app where the detailed trajectory, travel time and travel distance are recorded. Compared with trips without using our RS system, drivers can save 30% on average of the travel time in the busy hours.

## 9 RELATED WORK

Many existing works have been done to estimate the real-time travel time or traffic. Generally, they solve this problem from two directions depending on the data they use, one use End-To-End data (e.g. ETC, smart cards) and the other use Continuous data (e.g. GPS). Table 3 gives the summary of related works. Compared with other works, our work leverages existing infrastructures to estimate travel time on highways. Moreover, the system provides fine-grained estimation without using fine-grained data, which fills the blank of studies.

### 9.1 Continuous Perspective

The character of continuous data is that the intermediate status of objects can be captured at near real time. For example, taxis can upload their locations every minute [6]. With continuous data, both higher and lower granularity traffic conditions can be estimated. [22] estimates the traffic condition between regions in the city. [6]

Table 3. Traffic Estimation Survey

| Scenarios | | Continuous | End-To-End |
|---|---|---|---|
| City | Coarse-grained | [22] | [23] |
| | Fine-grained | [6] [11] [13] [12] [33] [28] [26] [32] [34] | [16] |
| Highway | Coarse-grained | [31] | [15] |
| | Fine-grained | [19] [21] | SharedEdge(Our work) |

estimates the city-scale traffic volume using taxis. [33] monitors the urban traffic with the help of bus riders. [26] estimates the road traffic delay using smartphones. [28] [34] [11] [13] [12] use sparse probe vehicle GPS locations to estimate the passed road traffic.

All these methods have a common problem of the low penetration rate with limited spatial-temporal coverage on the roads. It also brings issues like privacy and energy consumption when using devices like smartphones. In our method, ETC data provide high spatial-temporal coverage to avoid the low penetration rate issue considering its high volume. Private information such as identity is not recorded since we treat them anonymously to capture only the time and location when passing toll stations.

### 9.2 End-To-End Perspective

The character of End-To-End data is that only the information at end points can be captured (e.g. location, timestamp). The intermediate information is unknown or missing. [14][20][23][15] use various models to estimate the travel speed on an individual road segments based on the speed reading from loop sensors, then convert the speed into a travel time. [16] estimates the travel time in metro networks using smart card logs.

However, these works cannot satisfy our case. For [16], the key difference is that there is no congestion on the metro way so the riding time of the metro line is stable. The major variance of the travel time is because of the waiting time in the station. In our case, the driving time on the way is the dominating time. For loop sensor based methods, it is not practical to deploy loop sensors in all the road segments. Instead, our work takes one more step to estimate the travel time of individual edges.

## 10 DISCUSSION

In this section, we provide some discussions about SharedEdge as follows:

- **Extensibility:** Our paper only focuses on the OD based ETC system and aims to infer the edge-level travel time. However, based on our framework, many other applications can be implemented in the OD based systems like subway systems, bus systems, and traffic camera. In all these similar systems, we can obtain the information including origins, destinations, and travel duration from origin to destination. Based on the travel time estimation, real-time traffic speed can also be estimated if we know the road length. Similarly, other events, e.g., traffic jam, can be inferred if the travel time is longer than the normal situations based on anomaly detection techniques. Even in the GPS based systems, if we consider the OD-based detection as the sparest sampling of the GPS locations, then our framework can also be used for many GPS-tracking applications that do not require high GPS sampling rates.
- **Historical data:** SharedEdge leverages the historical data to initialize the edge travel time distribution. It may bring a cold start problem when we do not have enough historical data. To address the code start problem, we start our system from the early morning when there is less traffic on highways. In this case, people can drive as fast as they can under the speed limitation. Therefore, the travel time can be initialized as the edge length divided by the speed limitation, which is close to the travel time in real-world situations without heavy traffic.

- **Staying Time at Rest Areas:** In the highway system, there are many rest areas for people to rest or refuel. Depending on their individual situations, people decide if they would go to the rest areas and how long to stay there. In our system, we do not estimate the time cost at the rest areas. We assume that there are only small proportion of vehicles stopping in the rest areas compared to the whole traffic flow in the highway stations. If the vehicle spends too much time in the rest area, then its total travel time would be longer than others. We can consider this ETC record as outliers to avoid possible bias. If the vehicle only spends little time at the rest area, its total travel time would be close to others where the bias can be ignored.
- **Travel Time on Ramps:** In our work, we use a linear regression method to estimate the travel time in ramps in order to simplify the model. In fact, other complex models Support Vector Regression proposed in paper [30] can also be applied to improve the accuracy.

## 11 CONCLUSION

In this paper, we focus on the large scale highway systems and design a novel system called SharedEdge to estimate the fine-grained travel time on highways. Based on the constraints of multiple ETC records, we present a Bayesian graphical model to iteratively estimate the edge travel time in both batch and streaming manner. The key innovation of SharedEdge is that we estimate fine-grained travel time without using GPS trajectories. We evaluate the system with two datasets, one is the existing probe vehicle trajectories and the other is trajectories with our own drivers. Both experiments show our system can achieve even better accuracy than other methods. With the estimated travel time, we demonstrate a case study of the route suggestion system, which shows the system can indeed improve the travel efficiency in real life.

## ACKNOWLEDGMENTS

The authors would like to thank anonymous reviewers for their valuable comments. This work is partially supported by the by Rutgers Global Center, China 973 Program (2015CB352400), National Natural Science Foundation of China (41401470), Research Program of Shenzhen under Grants JSGG20150512145714248, KQCX 2015040111035011, CYZZ 20150403111012661 and KJYY 20160331162313860.

## REFERENCES

[1] China ministry of transport. http://www.mot.gov.cn/.
[2] Gaode map. http://www.gaode.com.
[3] Google map. https://www.google.com/maps.
[4] Open street map. http://www.openstreetmap.org.
[5] Y. Agarwal and M. Hall. Protectmyprivacy: detecting and mitigating privacy leaks on ios devices using crowdsourcing. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 97–110. ACM, 2013.
[6] J. Aslam, S. Lim, X. Pan, and D. Rus. City-scale traffic estimation from a roving sensor network. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 141–154. ACM, 2012.
[7] Y. Cong, B. Chen, M. Zhou, et al. Fast simulation of hyperplane-truncated multivariate normal distributions. *Bayesian Analysis*, 2017.
[8] C. De Fabritiis, R. Ragona, and G. Valenti. Traffic estimation and prediction based on real time floating car data. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 197–203. IEEE, 2008.
[9] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
[10] R. Herman and T. Lam. Trip time characteristics of journeys to and from work. *Transportation and traffic theory*, 6:57–86, 1974.
[11] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen. Learning the dynamics of arterial traffic from probe data using a dynamic bayesian network. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1679–1693, 2012.
[12] H. Hu, G. Li, Z. Bao, Y. Cui, and J. Feng. Crowdsourcing-based real-time urban traffic speed estimation: From trends to speeds. In *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*, pages 883–894. IEEE, 2016.
[13] T. Hunter, T. Das, M. Zaharia, P. Abbeel, and A. M. Bayen. Large-scale estimation in cyberphysical systems using streaming data: a case study with arterial traffic estimation. *IEEE Transactions on Automation Science and Engineering*, 10(4):884–898, 2013.

[14] Z. Jia, C. Chen, B. Coifman, and P. Varaiya. The pems algorithms for accurate, real-time estimates of g-factors and speeds from single-loop detectors. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pages 536–541. IEEE, 2001.
[15] J. Kwon, B. Coifman, and P. Bickel. Day-to-day travel-time trends and travel-time prediction from loop-detector data. *Transportation Research Record: Journal of the Transportation Research Board*, (1717):120–129, 2000.
[16] H. Lee, D. Zhang, T. He, and S. H. Son. Metrotime: Travel time decomposition under stochastic time table for metro networks. In *Smart Computing (SMARTCOMP), 2017 IEEE International Conference on*, pages 1–8. IEEE, 2017.
[17] S.-m. Li and Y.-m. Shum. Impacts of the national trunk highway system on accessibility in china. *Journal of Transport Geography*, 9(1):39–48, 2001.
[18] F. Marchal, J. Hackney, and K. Axhausen. Efficient map matching of large global positioning system data sets: Tests on speed-monitoring experiment in zürich. *Transportation Research Record: Journal of the Transportation Research Board*, (1935):93–100, 2005.
[19] C. Nanthawichit, T. Nakatsuji, and H. Suzuki. Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway. *Transportation Research Record: Journal of the Transportation Research Board*, (1855):49–59, 2003.
[20] K. F. Petty, P. Bickel, M. Ostland, J. Rice, F. Schoenberg, J. Jiang, and Y. Ritov. Accurate estimation of travel times from single-loop detectors. *Transportation Research Part A: Policy and Practice*, 32(1):1–17, 1998.
[21] F. Rempe, P. Franeck, U. Fastenrath, and K. Bogenberger. Online freeway traffic estimation with real floating car data. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pages 1838–1843. IEEE, 2016.
[22] F. Rempe, G. Huber, and K. Bogenberger. Travel time prediction in partitioned road networks based on floating car data. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pages 1982–1987. IEEE, 2016.
[23] J. Rice and E. Van Zwet. A simple and effective method for predicting travel times on freeways. *IEEE Transactions on Intelligent Transportation Systems*, 5(3):200–207, 2004.
[24] A. Richardson and M. Taylor. Travel time variability on commuter journeys. *High Speed Ground Transportation Journal*, 12(1), 1978.
[25] M. A. Stephens. Edf statistics for goodness of fit and some comparisons. *Journal of the American statistical Association*, 69(347):730–737, 1974.
[26] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 85–98. ACM, 2009.
[27] S. Wan, J. Meng, S. Fang, X. Xing, K. Xie, and K. Bian. Predictability analysis on expressway vehicle mobility using electronic toll collection data. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pages 2589–2594. IEEE, 2016.
[28] Y. Wang, Y. Zheng, and Y. Xue. Travel time estimation of a path using sparse trajectories. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 25–34. ACM, 2014.
[29] D. West. Updating mean and variance estimates: An improved method. *Communications of the ACM*, 22(9):532–535, 1979.
[30] C.-H. Wu, J.-M. Ho, and D.-T. Lee. Travel-time prediction with support vector regression. *IEEE transactions on intelligent transportation systems*, 5(4):276–281, 2004.
[31] S. V. Wunnava, K. Yen, T. Babij, R. Zavaleta, R. Romero, and C. Archilla. Travel time estimation using cell phones (ttecp) for highways and roadways. 2007.
[32] P. Zhou, Z. Chen, and M. Li. Smart traffic monitoring with participatory sensing. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 26. ACM, 2013.
[33] P. Zhou, S. Jiang, and M. Li. Urban traffic monitoring with the help of bus riders. In *Distributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference on*, pages 21–30. IEEE, 2015.
[34] H. Zhu, Y. Zhu, M. Li, and L. M. Ni. Seer: metropolitan-scale traffic perception based on lossy sensory data. In *INFOCOM 2009, IEEE*, pages 217–225. IEEE, 2009.
[35] Z. Zhuang, K.-H. Kim, and J. P. Singh. Improving energy efficiency of location sensing on smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 315–330. ACM, 2010.

Received August 2017; revised November 2017; accepted January 2018