# Link Travel Time Prediction from Large Scale Endpoint Data

Sankarshan Mridha, Niloy Ganguly, Sourangshu Bhattacharya
Indian Institute of Technology, Kharagpur
sankarshan@iitkgp.ac.in,[niloy,sourangshu]@cse.iitkgp.ernet.in

## ABSTRACT

Existing systems for travel time estimation either use data collected from loop detectors and probe vehicle locations, or from GPS traces from cellphones of "online" users. The former methods of data acquisition are expensive, while the latter turns out to be infeasible in connectivity-poor regions. However, many crowdsourced taxi trip datasets (from Boston, Beijing, Rome, etc.) are publicly available which, despite containing limited information, can be made useful for inferring meaningful insights by certain amount of data engineering. The datasets are both cheap to acquire (hence available in large volumes), and impose less heavy connectivity requirements on the end user. One such crowdsourced dataset is the NYC (New York City) Taxi dataset, which contains only the end-point information for each trip. In this paper, a link (road segment) travel time estimation algorithm named Least Square Estimation with Constraint (**LSEC**) has been developed from such end-point data, which estimates travel time 20% more accurately than existing algorithms. The key idea is to augment a subset of trips with unique paths using logged distance information, as opposed to fitting adhoc "route-choice" models.

## CCS CONCEPTS

•**Information systems** →**Data mining;** *Location based services;*

## KEYWORDS

Spatio-Temporal Data Mining, Travel Time Estimation, Path Certainty Estimation

**ACM Reference format:**
Sankarshan Mridha, Niloy Ganguly, Sourangshu Bhattacharya. 2017. Link Travel Time Prediction from Large Scale Endpoint Data. In *Proceedings of SIGSPATIAL'17, Los Angeles Area, CA, USA, November 7–10, 2017,* 4 pages.
DOI: 10.1145/3139958.3140006

## 1 INTRODUCTION

Estimation of link travel time [5] is a core problem in transportation engineering, with applications ranging from detection of traffic behaviour and anomalies at different places and times [3], assessing and improving overall efficiency of transportation systems [5], to the widespread application of route recommendation for individual users [1], etc. Google Maps[1] provides excellent route recommendation, but relies on users being *online* for collection of traffic data; this is not sustainable in many developing countries, since connectivity is expensive, and not as abundant as in developed countries. Therefore, a more ideal scenario for any recommendation algorithm would be to rely on data collected/released by the vehicles (and not individuals) plying in the city.

In recent times, several cities around the world have started online taxi hire companies, such as Uber[2], Ola[3], etc., which collect route information. Besides, in many cities, the city administration makes it mandatory for taxi hiring services to log their trip information. These taxi service organizations are increasingly making their taxi trip data public (e.g., Uber data[4]). However, understandably due to various constraints such as privacy concerns of the driver and the passenger, legal issues, business interests, etc., hardly any dataset contains complete trajectory information. For example, sometimes trip information is anonymized, or only pickup information is provided, or no distance information is shared, or exact pickup-dropoff information is masked. This trend is more or less common across taxi trip datasets released from different cities across the world like Porto, Boston, Chicago, New York, etc.

An important task, therefore, is to devise methods to meaningfully mine and utilize these increasingly available but incomplete trip datasets. In this paper, we specifically take up this challenge and develop a methodology to tackle the situation where entire trajectory information is obfuscated. New York City Taxi and Limousine Commission[5] have freely shared vast amounts of data for about all taxi trips made in the city recently. The data however does not contain routes for trips, but only starting and ending locations and timestamps along with the total distance travelled, fare, etc., for each taxi trip. We leverage on the huge amount of data provided to derive link level information of a large portion of links, and then use that for travel time estimation between any two given points. We believe the basic idea of *using the abundance of aggregate data to reconstruct fine-grained data (which is not available)* is a different outlook as none of the previous travel time estimation algorithms work on such incomplete data [1, 2, 4, 12].

In order to solve the problem on this dataset, the following steps are undertaken: (a) We reconstruct the route information from the (source, destination, distance) data, which is performed by enhancing the data with OSM[6] information; (b) From the route information, we estimate the travel time for each individual link[7]; (c) For any arbitrary route, we predict the travel time by stitching back the time taken to travel its constituent links

Our framework maps 65.6 million taxi trips in Manhattan in 2014 within 0.1 mile error threshold to unambiguous (unique) paths (section 2). The link mean travel time is accordingly estimated using our **LSEC** method (section 3), as the path travel time is spatially distributed over the constituent links of the paths. Results from experiments (section 4) with link travel time prediction algorithms, show that mean absolute percentage error (MAPE) for predicted link travel times is better by at least 20% than existing methods.

[1] http://maps.google.com

[2] https://www.uber.com/
[3] https://www.olacabs.com/
[4] https://movement.uber.com/cities
[5] http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml
[6] http://www.openstreetmap.org
[7] A link is a single segment of a road


*SIGSPATIAL'17, Los Angeles Area, CA, USA*
 978-1-4503-5490-5/17/11…$15.00
DOI: 10.1145/3139958.3140006

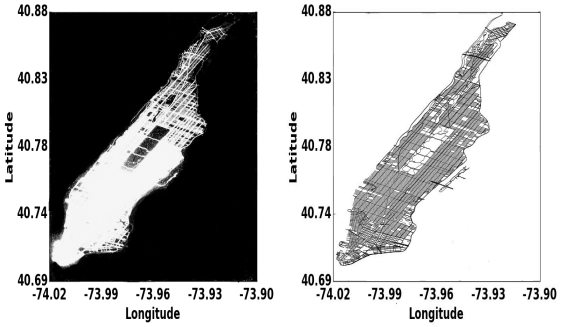


**Figure 1: (a) Original pickup, dropoff distribution (in *white*) and (b) The constructed full road network for Manhattan.**

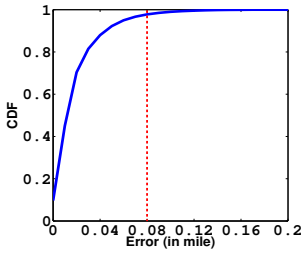


**Figure 2: *CDF* shows 99% taxi OD pairs are mapped to the closest node within an error threshold of 0.08 mile.**

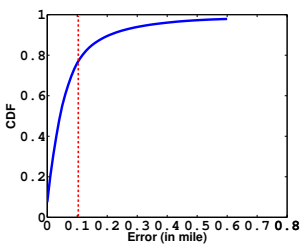


**Figure 3: *CDF* shows 76.1% trips are attributed to unique paths within 0.1 mile error threshold.**

**Related Work:** The two main factors determining systems for estimation of link travel time are: data sources and the underlying model. Li [2] uses a linear model to predict travel times from flow. Asghari et al. [1] discusses the intrinsic variability in link travel times and predicts the most reliable route. Both the above works use data from loop detectors, ANPR, etc. Another set of data sources are low-frequency GPS probe vehicle data [4], [12]. Zheng et al. [12], tackle the problem of redistributing travel times to intermediate links using a black-box neural network model. However, they need the actual travel times at the training stage which is hard to get for a large road network. Yuan et al. [9] uses time dependent landmark based model for estimating time between two landmarks. Rahmani et al. [4], predict travel times for routes, rather than links, thereby skipping the intricacies of path time models. [10, 11] try to estimate distributions of link travel times from the taxi trip data. They use the fare field logged in the data to infer the probability of path taken by a given trip. These methods are computationally expensive and unlikely to be scalable to citywide data. They also use EM algorithm, which only gives a local optimum and hence is prone to differences in initial points. Finally, they [10] estimate negative values for expected link travel times, which we have verified and compared their method with our **LSEC** method (section 4.2). This is is one of the reasons behind its inferior performance.

# 2 DATASET CONSTRUCTION

## 2.1 Data Sources

**NYC taxi dataset:** We obtained this data from the New York City Taxi and Limousine Commission. For this work we have focused only on Manhattan City and consider only those trips which start and end inside the Manhattan City itself. After initial data preprocessing, cleaning and filtering, it has **86.1** million valid trips made in Manhattan for the year 2014. However the trip data doesn't provide intermediate route information for any trip. It provides only the pickup and dropoff coordinates, timestamp along with total trip distance and duration. We have used only the following features: *pickup and dropoff timestamps, coordinates, trip distance and duration* only.

**OSM dataset:** While road networks are available for many cities, from the city corporation or state transportation authorities, very often these networks are incomplete or have missing properties [7]. Following [7], we use data from OpenStreetMap (OSM), which is an open source collaborative project dedicated to create free and editable map of the world. We download the OSM data for Manhattan city (longitude range: (−74.02,−73.90) and latitude range: (40.69, 40.88)) and have created the corresponding road network. Figure 1 shows the road network for Manhattan city, which has **14886** nodes and **23884** links.

## 2.2 Mapping Taxi Trip data to Road Network

We use *proximity based location mapping* to map all the original location to the closest node (according to Euclidean distance) in the road network. In the subsequent analysis, the original locations in the trips data, are replaced with the closest node in the road network. A total of 7028049 locations (resulting from taxi (pickup,dropoff) pairs) were mapped to **14886** nodes in the road network. Figure 2 plots the euclidean distance between the original (from NYC taxi data) and the mapped locations (from OSM data) (x-axis) against the cumulative fraction of points (y-axis). This distance measures the *error* in the mapping process. We can see that nearly **99%** of 7028049 locations are mapped with an error of only **0.08 mile** or less. The superficial (extra) nodes in the road network which capture the curvature of road segments contributed significantly in producing such highly accurate results.

## 2.3 Distance based Path Attribution

The biggest challenge here is that the path taken through the road network between origin-destination (OD) pairs is not provided. Recent works [10, 11] use probabilistic models to estimate the probability of alternative paths being taken. However as mentioned above, these approaches suffer from many drawbacks, including arriving at negative link travel time estimates. In this paper, we take an orthogonal approach of *distance based path attribution*. Our main idea is to consider trips which matches closely (with very low error) to a unique path. This leaves us with a dataset of trips which can be attributed to unique paths with low error.

We generate $k$-shortest paths between each $OD$ pair using Yen's k-shortest path algorithm [8], taking $k = 50$. For each path, the distance as measured as sum of distances of the links from road network. The distances of all paths are then clustered into *distance buckets*. If multiple paths fall inside same distance bucket for an $OD$ pair then we call them *ambiguous*. Next we extract the non-ambiguous path set for each OD pair. Finally the algorithm returns the path from non-ambiguous path set which has the closest distance to the trip distance for the OD pair.

We applied the above algorithm for attributing paths to the taxi-trip data filtered to Manhattan area. Note that the algorithm attributes paths to all trips, unless the corresponding $OD$ pair has

no non-ambiguous path. Figure 3 plots the *cdf* for errors (in mile) computed in the attributed paths. We can see that a large fraction of trips are attributed paths within an error of 0.1 mile, after which the rise in fraction of trips tapers off and the error in attribution increases. We choose 0.1 mile as the error threshold. **65.6 million** trips, out of a total of 86.1 million trips in the Manhattan area, lie within an error threshold of 0.1 miles (retention of 76.1%). We consider this filtered and path attributed set of trip as our *route annotated data*. We use this data for link travel time estimation in sections 3.

## 3 LINK TRAVEL TIME ESTIMATION

Let $G = (V, E)$ denote the road network, where $V$ is the set of all *nodes* which are the origin and destination points of various trips, $E \subseteq V \times V$ is the set of all road *links* in the network. We assume a directed graph with $e = (v_1, v_2)$ implying that traffic can flow from $v_1$ to $v_2$. A path $p$ is an ordered list of nodes ($V_p$) such that $(v_{i-1}, v_i) \in E, \forall i$. Equivalently, it can also be thought of as a list of links ($E_p$) such that if $e_{i-1} = (u_{i-1}, v_{i-1})$ and $e_i = (u_i, v_i)$, then $u_i = v_{i-1}, \forall i = 1, \ldots, M$, where $M$ is the total number of links in the road network. Our processed NYC trips dataset $\mathcal{D}$ obtained in section 2, can be described as a collection of paths $p_i$ along with trip travel time $T_i, \forall i = 1, \ldots, N$, where $N$ is the total number of of trips. Note that a given path $p_i$ may have multiple trips. Hence, we can compute the mean travel time $\bar{T}_p$ for a path $p$ as $\bar{T}_p = \frac{1}{N_p} \sum_{i:p_i=p} T_i$, where $N_p$ is the number of trips through path $p$.

In order to estimate link travel times, we formulate a probabilistic model for trip travel times. Let $X_k, k = 1, \ldots, M$ denote the random variables capturing travel times for the $k^{th}$ link. Note that this model is for weekday/weekend and an hour of the day. Also, let $T_p$ be the random variable denoting travel time for path $p$. We formulate the model as $\sum_{k \in p} X_k = T_p + \epsilon$, where $\epsilon$ is a zero mean Gaussian noise. This is a simplified model, which ignores waiting time at the nodes. Taking expectation w.r.t. all random variables $X_k$ and noting that $x_k = \mathbb{E}[X_k]$ and $\bar{T}_p = \mathbb{E}_X[T_p]$ we write $\sum_{k \in p} x_k = \bar{T}_p + \epsilon$. The assumption here is that the noise corrupting the observation of average path travel time $\bar{T}_p$ is independent of the path $p$ and zero mean Gaussian. We also assume the various trip travel times $T_i$ are due to variation in link travel times $X_k$. Hence, the average of trip travel times $\bar{T}_p$ for a path $p$ can be used as a plug-in estimate for $\mathbb{E}_X[T_p]$.

With this model, and the data for path travel times $\{\bar{T}_1, \ldots, \bar{T}_N\}$, we can write the above equations in a compact form as $A\boldsymbol{x} = \bar{\boldsymbol{T}} + \epsilon \mathbf{1}$, where $\boldsymbol{x} = [x_1, \ldots, x_M]^T$, $\bar{\boldsymbol{T}} = \left[\bar{T}_1, \ldots, \bar{T}_N\right]^T$, and $\mathbf{1}$ is the vector of all ones. $A_{M \times N}$ is a coefficient matrix, where $A_{ik} = 1$ if $i^{th}$ link is a part of $k^{th}$ path, $A_{ik} = 0$ otherwise. We use maximum likelihood paradigm to estimate the mean link travel times $x_k$. This leads to the least squares problem:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \|A\mathbf{x} - \bar{\mathbf{T}}\|_2 \tag{1}$$

We call this the least-square estimate (LSE) for link travel times.

A problem with this formulation is that the optimal mean link travel time estimates $x_k^*$ can be negative. This is because $\sum_{k \in p} x_k$ can be positive, even though the individual $x_k$ are not all positive. Hence, we impose the further constraint that $x_k \geq 0, \forall k = 1, \ldots, M$. The final estimation problem becomes:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} \parallel A\hat{\boldsymbol{x}} - \bar{\boldsymbol{T}} \parallel_2 \quad \text{s.t} \quad \boldsymbol{x} \geq 0 \tag{2}$$

We call this the least-squares estimation with constraints (**LSEC**). Note that this is an instance of non-negative least squares problem [6], which is a convex optimization problem, and can be solved using many efficient algorithms.

## 4 EXPERIMENTAL RESULTS

### 4.1 Setup

**Baseline methods:** PPE [10], assumes a linear model for fare in terms of distance and time, and tries to fit the coefficients using least squares regression. They assume a multinomial logistic regression model for route probabilities in terms of distance and time, using the fitted coefficients. Using the path probability, they compute expected travel time, which is matched to actual travel time for a non-linear least squares fitting. Our second baseline is basically the naive least squares estimation based algorithm LSE without positivity constraints on estimated links as described in Section 3, equation (1). While PPE needs the trip duration, distance and fare information, LSE and LSEC need trip duration and distance along with attributed paths.

**One week dataset:** Since optimization suggested in [10], uses the Jacobian matrix which is $O(|E|^2)$, PPE cannot be run on the entire road network. Hence, following [10], we select a smaller region of Manhattan City with 1829 nodes and 2334 road links for baseline comparison. Next we construct a smaller taxi trip dataset for running their algorithm. While the authors have used single week's data for the year 2010 we have used it for year 2014. They used the trip data from 3/15/2010 to 3/21/2010 where as we use our path attributed taxi trip data from 3/15/2014 to 3/21/2014 whose pickup and dropoff fall into test region and got total 155601 trips for these seven days period.

**Experimental procedure:** We compared with baseline using one week dataset (section 4.2) and validated the performance of our method on the entire 2014 dataset separately (section 4.3). We randomly select (80%) training and (20%) test set data and evaluation was done on test set data. All the work has been performed on machine having 48 CPU (Intel(R) Xeon(R) CPU, $E5-2697$ v2 @ 2.70GHz), RAM 256 GB. Distance based path matching took roughly $\sim 10^{-3}$ seconds per trip. For the road network with 23884 link, our LSEC takes on an average $\sim 0.60$ seconds per link for link mean and variance in travel time estimation.

### 4.2 Baseline Comparison on 1-week Data

Table 1 shows root mean square error (RMSE) and mean absolute percentage error (MAPE) for link travel time estimated by **LSEC**, LSE and PPE for four different time of the day 6:00 am (6), 10:00 am (10), 4:00 pm (16), and 8:00 pm (20). We note that LSEC performs nearly similar to LSE while it performs much better than PPE, both in terms of RMSE and MAPE. We observe, RMSE is higher for rush hours 10 and 16 when the average trip duration per mile (TDM) is higher. While for hour 6 and 20 the error is comparatively low. The approach proposed in PPE suffers from several problems. First, linear relationship between fare distance and time could not be verified for the full one year dataset (they verify it with trip data of seven days spanned over a very small part of Manhattan). Second, it assumes that probability of following a path is related to the fare, which is not verified for larger city wide dataset. Third, the approach is computationally very expensive as it maintains probabilities for top-$k$ paths for all trips, and hence could not be applied to the entire dataset.

| Hour | PPE | | LSE | | LSEC | |
|---|---|---|---|---|---|---|
| | RMSE (minute) | MAPE (%) | RMSE (minute) | MAPE (%) | RMSE (minute) | MAPE (%) |
| *6* | 1.56 | 25.68 | 1.13 | 20.14 | **1.12** | **19.15** |
| *10* | 3.09 | 29.25 | 2.53 | 22.21 | **2.26** | **21.57** |
| *16* | 2.71 | 30.16 | 2.29 | 23.34 | **2.12** | **22.26** |
| *20* | 1.76 | 26.28 | 1.33 | 20.43 | **1.29** | **20.12** |

**Table 1: Comparison of method LSEC with baseline PPE and LSE. LSEC is showing good progress over PPE in terms of both RMSE and MAPE.**

| Hour | | 6 | 10 | 16 | 20 |
|---|---|---|---|---|---|
| *(%) of -ve* | *PPE* | 23.73 | 24.25 | 27.97 | 24.29 |
| *Link time* | *LSE* | 12.12 | 15.81 | 14.65 | 13.24 |

**Table 2: Comparison of percentage (%) negative link between PPE and LSE. For each of the hours CDF of LSE is always lower than that of PPE.**

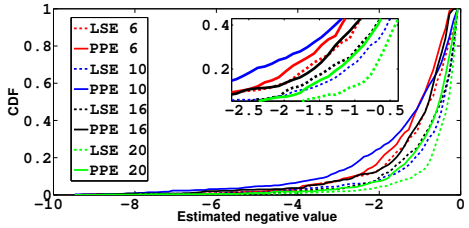


**Figure 4: Comparison of frequency distribution of estimated negative links for PPE and LSE for different hours of the day.**

| Hour | Day Type | Path Mean Time Estimation | | | | TC |
|---|---|---|---|---|---|---|
| | | MAPE (%) | RMSE (min) | PC (TC>0) | LC (%) | |
| *6* | *W* | 19.10 | 1.57 | 120640 | 77.06 | 1434065 |
| *10* | *W* | 23.00 | 3.36 | 372090 | 79.57 | 3166472 |
| *16* | *W* | 23.34 | 3.37 | 316246 | 80.96 | 2503511 |
| *20* | *W* | 19.62 | 2.27 | 534513 | 84.61 | 3639485 |
| *6* | *NW* | 21.23 | 1.47 | 57457 | 73.56 | 184571 |
| *10* | *NW* | 20.10 | 2.14 | 221421 | 78.57 | 1017649 |
| *16* | *NW* | 23.68 | 3.00 | 229880 | 80.06 | 1054417 |
| *20* | *NW* | 22.51 | 2.60 | 228005 | 81.26 | 1210975 |

**Table 3: Measuring error in estimation of path mean travel time for *weekdays* (*W*) and *weekends* (*NW*). It also shows hourly details about Trip Count (TC), Path Count (PC) for (TC > 0) during Weekdays (W) and Weekends (NW) and the corresponding Link Coverage (LC).**

**Effect of negative links:** A major drawback of both the baselines is that they have no mechanism to stop estimating negative link travel times. Table 2 reports the percentage of links with estimated negative link travel times. We note that both LSE and PPE estimate a significant percentage of negative links. A further investigation also shows the correlation between percentage of negative links and the accuracy of travel time prediction. In figure 4 we have plotted the CDF of negative links frequency for different hours of the day for LSE and PPE. We see that the frequency of negative values estimated by PPE (given by solid lines) is more than those of LSE (given by dotted lines), which correlates with the inferior performance of PPE compared to LSE. Hence, we conclude that one of the reasons behind the inferior performance of PPE over LSE (and also LSE over LSEC) is the spurious computation of negative link travel times.

### 4.3 Performance of LSEC on 1-year Data

We see in table 3, for estimating path time by our **LSEC** method, *MAPE* value is within 24% - intrinsic variability in the travel time data is partially responsible for the error. The undershoot and overshoot of estimation are roughly in similar proportion. Notice that although the MAPE is slightly worse in the weekend than the weekdays, the RMSE shows opposite behaviour because the car speed is higher in weekend (people reach their destination faster). Hence although the percentage error is high, the absolute mismatch with respect to time is low. The table also shows the number of estimated links over Manhattan city - the experiment covers. It is nearly 80% and 78% for weekdays and weekends respectively, clearly signifying the scale of our experiment.

## 5 CONCLUSION

The primary contribution of this paper is to demonstrate a simple yet effective framework, to calculate mean travel time for each link in a map, using **LSEC**. This is achieved on a large historical taxi trip dataset, which contains only the *end point information* for each trip. The paper shows that using a road network, and a simple concept of non-ambiguous shortest paths, we can reconstruct routes and estimate travel time on a city-wide scale, which none of the existing research systems do. The percentage of links whose mean could be calculated, is around 80%. We believe the coverage would increase further if we work with more data. A simple way to enhance data would be to consider data from all seven years (2009 - 2015) together, which is a future work. Finally, we plan to make the annotated (with derived route information) NYC taxi data public for future research endeavours.

## REFERENCES

[1] Mohammad Asghari, Tobias Emrich, Ugur Demiryurek, and Cyrus Shahabi. 2015. Probabilistic estimation of link travel times in dynamic road networks. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 47.

[2] Yanying Li. 2008. Short-term prediction of motorway travel time using ANPR and loop data. *Journal of Forecasting* 27, 6 (2008), 507–517.

[3] Bei Pan, Ugur Demiryurek, and Cyrus Shahabi. 2012. Utilizing real-world transportation data for accurate traffic prediction. In *2012 IEEE 12th International Conference on Data Mining*. IEEE, 595–604.

[4] Mahmood Rahmani, Erik Jenelius, and Haris N Koutsopoulos. 2015. Non-parametric estimation of route travel time distributions from low-frequency floating car data. *Transportation Research Part C: Emerging Technologies* 58 (2015), 343–362.

[5] Irum Sanaullah. 2013. *Real-time estimation of travel time using low frequency GPS data from moving sensors*. Ph.D. Dissertation. © Irum Sanaullah.

[6] Philip B Stark and Robert L Parker. 1995. Bounded-variable least-squares: an algorithm and applications. *Computational Statistics* 10 (1995), 129–129.

[7] Jameson L Toole, Serdar Colak, Bradley Sturt, Lauren P Alexander, Alexandre Evsukoff, and Marta C González. 2015. The path most traveled: Travel demand estimation using big data resources. *Transportation Research Part C: Emerging Technologies* 58 (2015), 162–177.

[8] Jin Y Yen. 1971. Finding the k shortest loopless paths in a network. *management Science* 17, 11 (1971), 712–716.

[9] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. 2010. T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*. ACM, 99–108.

[10] Xianyuan Zhan, Samiul Hasan, Satish V Ukkusuri, and Camille Kamga. 2013. Urban link travel time estimation using large-scale taxi data with partial information. *Transportation Research Part C: Emerging Technologies* 33 (2013), 37–49.

[11] Xianyuan Zhan, Satish V Ukkusuri, and Chao Yang. 2015. A Bayesian mixture model for short-term average link travel time estimation using large-scale limited information trip-based data. *Automation in Construction* (2015).

[12] Fangfang Zheng and Henk Van Zuylen. 2013. Urban link travel time estimation based on sparse probe vehicle data. *Transportation Research Part C: Emerging Technologies* 31 (2013), 145–157.