IMPROVING ESTIMATION ACCURACY OF GPS-BASED ARTERIAL TRAVEL TIME USING K-NEAREST NEIGHBORS ALGORITHM

by

Zheng Li

A Thesis Submitted to the Faculty of the

DEPARTMENT OF CIVIL ENGINEERING AND ENGINEERING MECHANICS

In Partial Fulfillment of the Requirements

For the Degree of

MASTER OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

2017

STATEMENT BY AUTHOR

The thesis titled *Improving Estimation Accuracy of GPS-based Arterial Travel Time using k-Nearest Neighbors Algorithm* by *Zheng Li* has been submitted in partial fulfillment of requirements for a master's degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that an accurate acknowledgment of the source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: *Zheng Li*

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

________________________________	08/01/2017
Yao-Jan Wu	Date
Assistant Professor of Transportation	

ACKNOWLEDGEMENTS

I would like to thank my advisor Prof. Yao-Jan Wu for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Yi-Chang Chiu and Dr. Xianbiao Hu, for their insightful comments and encouragement. I also learned a lot under their guidance in both academic study and industry experience.

I would also like to thank Dr. Robert Kluger for contributing his time to revise my thesis. I could not have imagined having a high-quality thesis without his guidance on the organization of the study.

I appreciate many other students at the University of Arizona who have helped me during the study. I would like to thank Payton Cooke for reviewing my draft and providing practical suggestions.

Special thanks to Metropia, Inc. for providing the probe vehicle dataset and road network dataset. There was also support from Psomas, the Pima Association of Governments, and the City of Tucson. This project was supported with funding from the Regional Transportation Authority.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Link travel time plays a significant role in traffic planning, traffic management and Advanced Traveler Information Systems (ATIS). A public probe vehicle dataset is a probe vehicle dataset that is collected from public people or public transport. The appearance of public probe vehicle datasets can support travel time collection at a large temporal and spatial scale but at a relatively low cost. Traditionally, link travel time is the aggregation of travel time by different movements. A recent study proved that link travel time of different movements is significantly different from their aggregation. However, there is still not a complete framework for estimating movement-based link travel time. In addition, probe vehicle datasets usually have a low penetration rate but no previous study has solved this problem.

To solve the problems above, this study proposed a detailed framework to estimate movement-based link travel time using a high sampling rate public probe vehicle dataset. Our study proposed a k-Nearest Neighbors (k-NN) regression method to increase travel time samples using incomplete trajectory. An incomplete trajectory was compared with historical complete trajectories and the link travel time of the incomplete trajectory was represented by its similar complete trajectories. The result of our study showed that the method can significantly increase link travel time samples but there are still limitations. In addition, our study investigated the performance of k-NN regression under different parameters and input data. The sensitivity analysis of k-NN algorithm showed that the algorithm performed differently under different parameters and input data. Our study suggests optimal parameters should be selected using a historical dataset before real-world application.

1. INTRODUCTION

1.1 Background

Travel time plays a significant role in traffic planning, traffic management, and Advanced Traveler Information Systems (ATIS). While variables such as occupancy, speed and flow are popular in transportation engineering field, travel time is a more intuitive concept and can be easily understood by non-experts. Travel time is defined as the total time for a vehicle to travel from one point to another over a specified route, including stops and delay (Zhu et al., 2009). Link travel time is widely used to measure corridor performance and monitor traffic conditions. It also is used to provide accurate traffic information to travelers to enable them to make a better route choice and thereby contribute to road network balance. Traditionally, travel time is either collected from fixed sensors (loop detectors, microwave sensors, cameras, Bluetooth devices, etc.) or by mobile sensors and surveyors (manual collection, floating cars, Global Positioning System, etc.). However, in most of these cases the travel time estimation is relatively inaccurate because of the limited time and location information provided. In addition, travel time data can only be collected on some individual links or time periods due to the high cost of devices and labor.

Recently, there has been an increasing trend of using large public probe vehicle datasets (taxis, transits, navigation app data, etc.) for travel time estimation. One advantage of using public probe vehicle datasets is that they provide the possibility for real-time travel time estimation due to their large amount of data. In addition, because large public datasets can cover most of the links in a city, the travel time estimation could be expanded to a city level. Furthermore, using public probe vehicle datasets has a lower cost compared to traditional methods since expensive sensors and labor expenses are not required.

1.2 Problem Statement

While probe vehicle travel time estimation has plenty of advantages over traditional methods, it has several limitations. First, accurate travel time estimation requires a relatively high penetration rate and sampling rate. The penetration rate is defined as "the flow fraction of vehicles (unique devices) reporting to the probe data set as compared to the total flow of vehicles along a road and sampling rate is the average rate at which any device reports its position and velocity*" (Patire et al., 2015)*. Since probe vehicles are samples from all vehicles on the road, the travel time estimation result may not be statistically significant if the penetration rate is low. Also, a low sampling rate will lower the accuracy of travel time estimation. Most current probe vehicle datasets have a relatively low sampling rate and penetration rate, which limit their applications. Public probe vehicle datasets also suffer from uneven temporal-spatial sample distribution. For example, more data is collected on major arterials but less on low-grade sections; and more data is collected during peak hours but less collected during non-peak or even none may be collected in the late night.

As a result of the limitation of current probe vehicle datasets, most of the current studies can be classified into two research areas. The first research area aims to estimate travel time when the sampling rate or the penetration rate is low (Zheng et al., 2013; Wan et al., 2016; Jenelius et al., 2013; Partsinevelops et al., 2005; Bucknell et al., 2014; Argote-Cabañero et al., 2015; Zhan et al., 2013). Another research area focuses on improving the accuracy of travel time estimation (Zhang et al., 2015; Cao et al., 2014; Seo et al., 2015; Hellinga et al., 2002). Although there are already a large number of reports of research on travel time estimation using probe vehicle data, most of the previous studies focused on the individual link travel time estimation, only a few previous studies estimated travel time by different movements (e.g., go through the link, left turn, right turn). In addition, most of the previous studies were built on the scenario in which the probe vehicle data has a relatively high penetration rate but low sampling rate. Only a few studies focused on travel time estimation when probe vehicle data had a good sampling rate but poor penetration rate. Finally, most of the previous studies focused on how to utilize probe vehicle data, but only a few studies tried to increase the sample size of the study.

Accurate urban link travel time estimation can provide more accurate information to agencies, researchers and travelers. The methods in this study can be used for different scenarios, including short-term traffic management, long-term traffic planning, among other applications. Specifically, the travel time information by different movements can be used for signal timing system evaluation, corridor before-and-after study, navigation, etc. Furthermore, the implementation of k-nearest neighbors regression algorithm can increase the sample size of the travel time estimation as well as improve the accuracy of link travel time estimation.

1.3 Research Objectives

The objectives of this study include:

(1) To investigate the travel time estimation method when the dataset has a low penetration rate and high sampling rate;

(2) To develop a movement based travel time estimation method using probe vehicle data;

(3) To utilize algorithm to increase the sample size or penetration rate for probe vehicle based travel time estimation;

(4) To investigate the performance of the algorithm that increases sample size.

1.3 Thesis Organization

This thesis is organized as follows: The next chapter discusses related literature on urban arterial travel time estimation and k-nearest neighbors algorithm. Chapter 3 introduces the study corridor and the dataset of the study. Chapter 4 first discusses the movement based travel time estimation method, and introduces the implementation of the k-nearest neighbors algorithm in increasing travel time estimation sample size, then presents the sensitivity analysis method for the k-nearest neighbors algorithm. Chapter 5 describes the case study of various scenarios using the framework discussed in the Chapter 4. The case studies were conducted on Grant Road in Tucson, Arizona. In the first case study, the movement based travel time estimation result is described and the sample size of travel time estimation before and after using k-nearest neighbors (k-NN) algorithm is discussed. Then, the sensitivity analysis of the algorithm was conducted using leave-one-out cross validation method. Chapter 6 summarizes the research results, discusses the deficiency of the study, and proposes potential future work.

2. LITERATURE REVIEW

There are many travel time estimation frameworks according to the varieties of data source and travel time estimation methods. This section systematically reviews prior studies of travel time estimation from three aspects, including the data source, probe vehicle travel time estimation methods, and k-nearest neighbors algorithm.

2.1 Travel Time Estimation Data Source

Traditionally, travel time estimation for an urban area is relied mainly on fixed sensors, including: loop detectors (Coifman, 2002; Robinson and Polak, 2005; Wu et al. 2004), automated vehicle identification (AVI) (Park and Rilett, 1998; Li and Rose, 2011; Sherali et al., 2006), Bluetooth devices (Wang et al., 2011; Haghani et al. 2010; Park et al. 2016), microwave sensors (Yeon et al., 2008) and so on. All the above-mentioned data collection methods require corresponding sensors installed to retrieve data. Once the sensor is installed, it can continuously record data on the monitored road section. However, the cost of installing and maintaining fixed sensors is relatively high because a large number of sensors are needed to achieve the appropriate accuracy level or cover a large research area.

An alternative approach is to measure travel times by mobile traffic sensors, e.g., floating cars (Byon et al., 2006), probe vehicles (Boyce et al., 1994), cellular data and so on. Vehicles equipped with tracking devices (GPS or mobile phone) can be used for collecting travel times at any location without roadside equipment. However, mobile sensors are still costly because stabilized data collection needs operational vehicles running on the study area all the time. Hence, they can only cover a limited number of routes for a limited duration of time (Jenelius and Koutsopoulos, 2013). Due to the cost consideration, there are only a small number of traffic studies using mobile sensors.

Recently, many public vehicles (e.g., taxis, transit, etc.) are equipped with GPS devices. These public vehicles, to some extent, are probe vehicles and they can collect travel time on most of the network links during their service time with a low cost. In addition, with the popularity of mobile phones, trajectory data that is collected from mobile phones can also be used for travel time estimation. The appearance of the new data sources provides the possibility for a large-scale and long-term travel time estimation. Along with the growth and availability of probe vehicle dataset, numerous studies have been conducted on travel time estimation using public datasets. Zhan et. al (2013) successfully estimated hourly travel time using NYC taxicab origin and destination (OD) trip data. Jenelius et. al (2013) discussed a statistical model for urban road network travel time estimation using vehicle trajectories obtained from low frequency GPS probes. A case study was conducted on an arterial network in Stockholm, Sweden using taxi fleets data.

Data fusion technique provides an approach to combine different data together to increase the accuracy of the travel time estimation result. Nantes et. al developed a framework to estimate travel time by a combination of heterogeneous data sources, especially loop detectors, probe vehicles and Bluetooth sensors (Nantes et al., 2016). Mehran et al. reconstructed the trajectory of sparse probe vehicle data in order to get travel time information by a data fusion of probe vehicle, fixed sensor and signal timing.

According to the location where travel time is collected, travel time can be classified into travel time on freeways or travel time on arterials. While vehicular flow on freeways is often treated as uninterrupted flow, flow on arterials is much more complicated since it can be affected by signal delay, queue delay, pedestrians and entry vehicles. There is plenty of travel time estimation research on freeways (Moorthy and Ratcliffe, 1988; Lee and Fambro, 1999; Lin, 2001; Abdulhai et al., 2002), but the travel time estimation research on urban areas is very limited. In any case, on highway or urban environments, since travel time depends on the origin and destination, ATIS normally use methods that calculate travel time at a link or section level, which change the research object from trips to road sections (Cheu et al. 2002). Feng et al. proposed that the distribution of link travel time in an urban area can be approximated using mixtures of normal distributions. While historical travel time data is available, probe vehicle data can be used to identify current traffic statement based on Bayes Theorem (Feng et al., 2014).

2.2 Probe Vehicle Travel Time Estimation Methods

As explained in the previous section, travel time estimation models strongly depend on the data. Since each type of traffic sensor provides different traffic information, only probe vehicle based estimation models are reviewed.

Probe vehicles equipped with GPS systems can collect position, speed and time stamp data every few seconds (Li and McDonald, 2002). Theoretically, probe vehicles can provide all the information needed to calculate travel time on any area at any time. However, due to the shortcoming of current probe vehicle datasets, this approach still has many limitations with respect to applications of probe vehicle travel time estimations.

The limitations mainly come from two aspects: low sampling rate and/or low penetration rate.

Low sampling rate has made it difficult to measure travel time directly because few information is known between every two continuous data points. Since most current datasets have a low sampling rate, there are many papers that seek to calculate accurate travel time using a sparse probe vehicle dataset. Wan et al. proposed a method to reconstruct maximum likelihood trajectory of probe vehicles between sparse updates based on Expectation Maximization algorithm (Wan et al., 2016). Another method is to use models to estimate travel time (neural networks, etc.). Zheng et al. built a three-layer neural network model to estimate complete link travel time for individual probe vehicle traversing the link and both simulation data, and real-world data were used to verify the result of the model (Zheng et al., 2017).

When penetration rate is low, probe vehicle samples cannot represent the entire population and the estimation is not accurate. There is much research on the relationship between sample size and estimation error. Patire et al. analyzed the estimation error when sampling rate and penetration rate are different by a data fusion approach (Patire et al., 2015). Bucknell et al. analyzed estimation error of different combinations of penetration rate and sampling rate on highways using NGSIM dataset (Bucknell et al., 2014). However, until now, there is no research investigation on how to increase the sample size of probe vehicle datasets. To some extent, the appearance of public probe vehicle datasets can increase the penetration rate, which is important for the application of probe vehicle data. However, the problem of low penetration rate is still very common, and this means a way to increase probe vehicle sample size based on existing datasets is required.

Finally, link travel time estimation in most of the previous studies is the time difference from upstream to downstream and the movement of vehicles are not considered. However, travel time is highly related with the movement of vehicle. For example, typically, left turn vehicles experience a longer travel time than through movement vehicles. Travel time estimation by movements is essential because it can provide more accurate information for agencies and travelers. Additionally, there are several studies on the penetration rate requirement for the probe vehicle travel time estimation (Patire et al., 2015; Srinivasan and Jovanis, 1996; Argote-Cabañero et al., 2015; Bucknell et al., 2014), but only very few studies focus on increasing penetration rate (Liu et al., 2009). Increasing the penetration rate can effectively reduce the cost of data collection and increase the accuracy of travel time estimation. In addition, there is few valid methods to increase probe vehicle samples without adding new data source.

2.3 K-Nearest Neighbors Regression Algorithm

The K-Nearest Neighbors regression algorithm (k-NN) is a non-parametric technique and it has been widely used in travel time estimation. Handley et al. (1998) used flow, occupancy and other variables as inputs of k-NN algorithm to estimate travel time on freeways. Robinson and Polak (2005) successfully used single loop detector data as inputs of k-NN to estimate travel time within an urban area. They compared different parameters of k-NN algorithm and the result of k-NN algorithm with other algorithms such as Neural Network. They also inferred that there is a high potential to use the probe vehicle GPS data as the input of the k-NN algorithm. Zhou et al. (2016) applied sparse probe vehicle data as the input of k-NN algorithm to estimate link travel time in an urban area. The study suggested that the k-NN algorithm performed better than the Neural Network model.

k-NN has many advantages over other regression algorithms, which makes it suitable for probe vehicle data. The assumption under k-NN regression is that target value is represented by k closest samples. Compared with parametric techniques like linear regression, k-NN has no target function, which is more suitable to probe vehicle data concerning an urban area. Probe vehicles are greatly affected by surrounding environment (road geometry, signal timing, time of day, and other vehicles, etc.) so that a fixed target function may not be able to fit the data well. Compared with other non-parametric techniques like Neural Network, k-NN is not only simple but also contains transportation engineering theory. The training process of k-NN can be explained but other models, like neural network, are hard to explain.

3. DATA

Typically, probe vehicle based travel time estimation requires two types of data: probe vehicle trajectory data and road network data. In order to better illustrate the method, the public probe vehicle dataset used in this study is introduced first. Then, detailed data fields and the study area are discussed. After that, the detailed process of data cleaning and processing is presented. Finally, processed data, which is the input of travel time estimation and k-nearest neighbor algorithm, is introduced.

3.1 Data Source

Two main types of data were used in this study. The first type was second-by-second probe vehicle data collected from a smartphone navigation app[1]. The probe vehicle data included location, speed and acceleration information. The GPS module in the phone begins to collect data when a user starts navigation and send collected data back to the server in real time. When users reach to their destination, navigation and data collection will finish automatically.

The other type of data was road network data. Road network data was consisted of links, defined as a straight one-direction road segment from one point to another point. Road network data contained the road geometry information (location, length, direction, etc.) and topological information (i.e., the topological relationship between different links).

[1] The smartphone app named "Metropia" (http://www.metropia.com/)

Probe vehicle data that is collected from the navigation app has many advantages over the data collected from taxis or public transportation. Probe vehicle trajectory from navigation app is typically not affected by passengers. For example, the trajectory of taxis includes traveling time and dwell time of passengers, which introduce error to travel time estimation.

3.2 Data Description

3.2.1 Study Corridor

The study corridor mainly focused on Grant Road between I-10 and Swan Road in Tucson, Arizona, and data collection was conducted in both directions. Grant Road is a major east-west direction arterial with annual average daily traffic (AADT) of 36,000 per day (Pima Association of Governments, 2014). The study corridor is shown in Figure 1 with primary cross-streets labeled. Most of the road was five-lane in total, with two lanes in each direction and one lane in the center for left turns. At the time of data collection, the only six-lane sections extended from Fairview Ave. to Stone Ave. and starting at Swan Ave. heading eastward. All study links have a speed limit of 40 mph (64 km/h). The links in the study refer to the one direction segment between each contiguous primary cross-streets on Grant Road. For example, the eastbound Oracle-Stone link refers to the road on Grant Road between Oracle Road to Stone Road in the east direction.

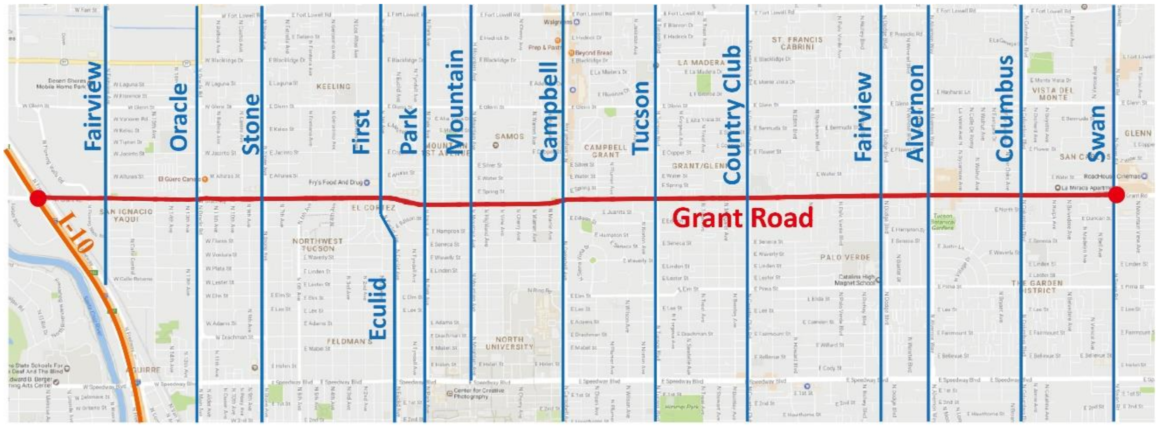


Figure 1 Study Corridor

3.2.2 Probe Vehicle Trajectory

Vehicle trajectory data were used to extract travel time information and further to build a historical database for the K-NN algorithm. The data was collected by the smartphone app when a user starts a trip using the app, the internal GPS module built into the smartphone is activated and starts to record the second-by-second data. These data, including detailed position such as latitude, longitude, heading, timestamp, velocity and corresponding link in the roadway network are collected at a fine time interval and sent back to the cloud server, where they are stored and will be used for further analysis.

The original data was collected from January 1st to December 31th, 2015. There were 57,645,478 GPS points collected from 1837 users and 43,315 trips in Tucson. The example of probe vehicle trajectory data is shown in Table 1 and visualization of partial data is shown in Figure 2.

Table 1 Example of Probe Vehicle Data

MetropianID	TrajectoryID	ReservationID	LinkID	Longitude	Latitude	Altitude	direction	unixtime	speed	accuracy
3593	4707	73	6839	-110.940369	32.250275	2420	90	1437554419363	36	16
3593	4708	73	6839	-110.940193	32.250275	2421	90	1437554420045	36	16
3593	4709	73	6839	-110.940018	32.250275	2422	89	1437554421067	37	16
3593	4710	73	6839	-110.939842	32.250271	2421	90	1437554422045	36	16
3593	4711	73	6839	-110.939674	32.250271	2420	89	1437554423052	37	16
3593	4712	73	6839	-110.939491	32.250271	2420	89	1437554424039	37	16
3593	4713	73	6839	-110.939308	32.250267	2418	89	1437554425042	38	16
3593	4714	73	13262	-110.939117	32.250263	2418	89	1437554426053	39	16
3593	4715	73	13262	-110.938934	32.250267	2416	89	1437554427049	39	16
3593	4716	73	13262	-110.938751	32.250267	2416	89	1437554428064	39	16

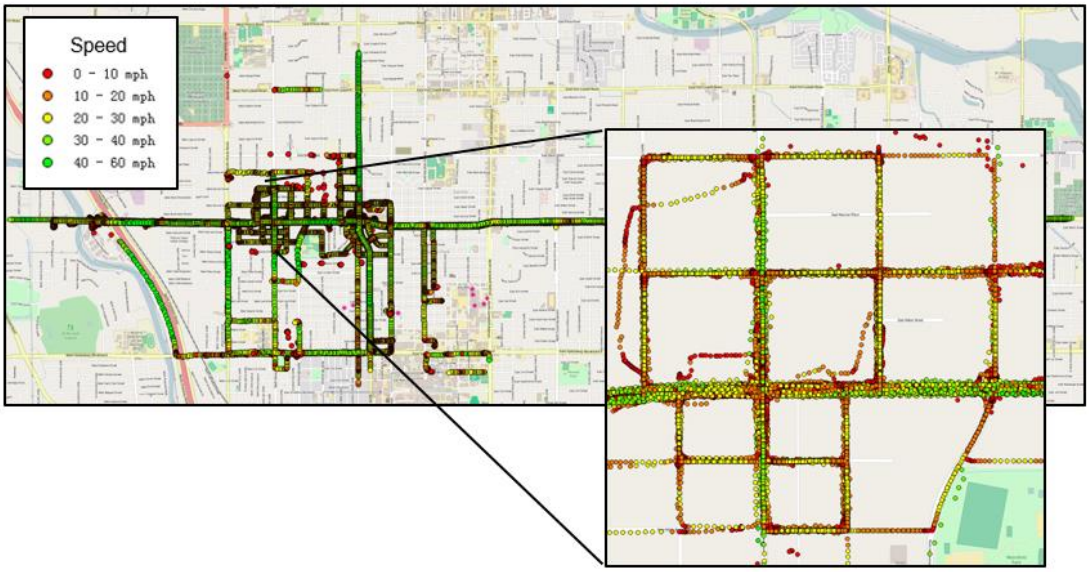


Figure 2 Original Probe Vehicle Data Visualization

The detailed description of data fields is shown in Table 1. Each record or each row is one GPS point.

Table 2 Original Probe Vehicle Data Fields

Attributes	Description
MetropianID	ID of app users (unique)
ReservationID	ID of trips (unique)
TrajectoryID	ID of GPS points (unique)

LinkID	ID of the link that GPS point was on. The value was given when GPS data was collected.
Longitude	Longitude
Latitude	Latitude
Altitude	Altitude
Direction	Moving direction of the vehicle. The value increases clockwise from north direction and range from 0 to 360
Unixtime	Unixtime timestamp of the GPS points
Speed	Speed of the vehicle (MPH, mile per hour)
Accuracy	Spatial accuracy (feet), means error that reported coordination is within.

The distribution of accuracy is shown in Figure 3. The accuracy of the most of original data was lower than 32 feet.

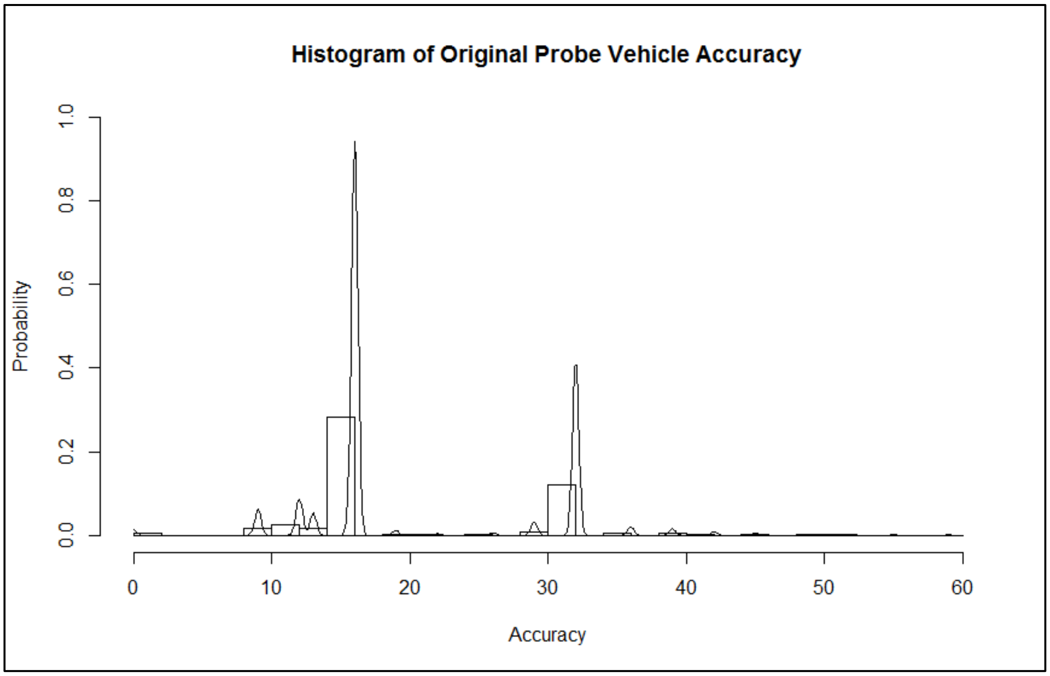


Figure 3 Histogram of Original Probe Vehicle Accuracy

3.2.3 Road Geometry Data

Roadway geometry data included the location of the intersections and links, type of road segment (e.g., freeway, highway, arterial), speed limit, link length, turn connections between links (e.g., left turn, right turn, etc.) and so on. This roadway geometry data is originally used for routing purposes. As a result, links are relatively short (sometimes 200 feet or less) and they cannot be used directly for travel time estimation. This occurs mainly because the length of the links in the data is relatively short and it will cause low accuracy of travel time estimation if they combined with the GPS data with a sampling rate of one point per second.

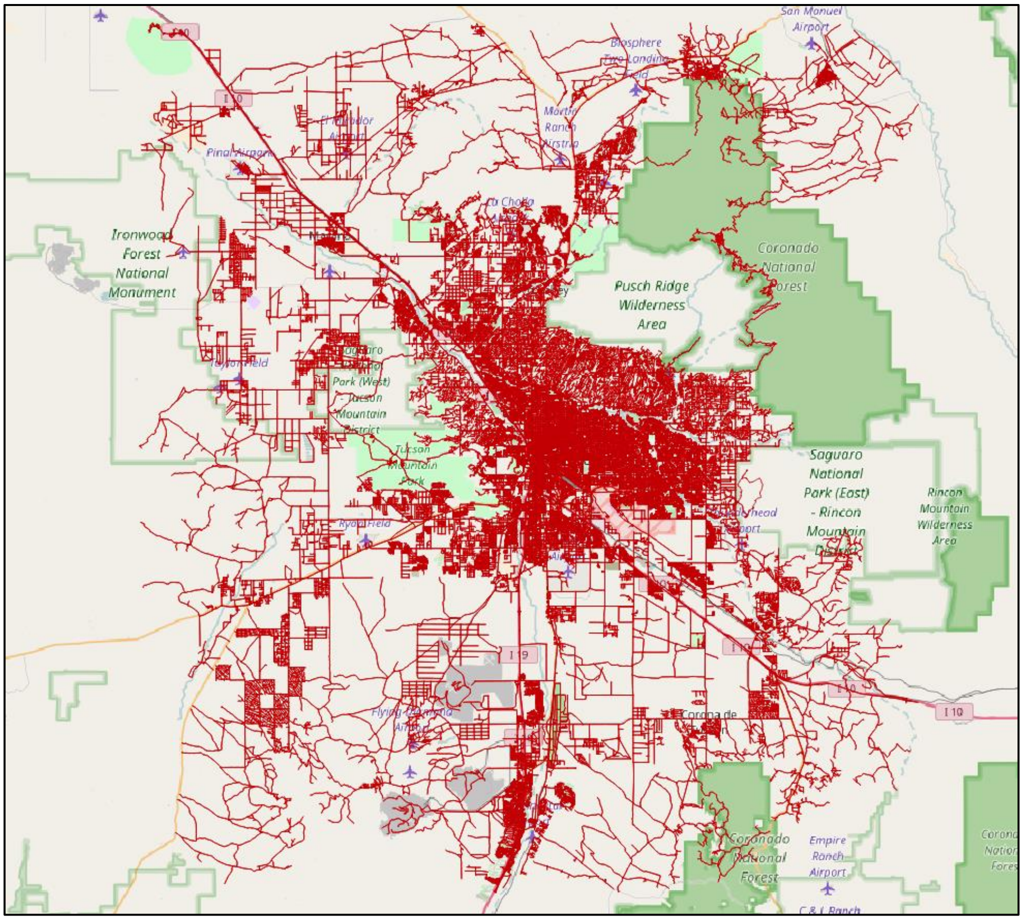
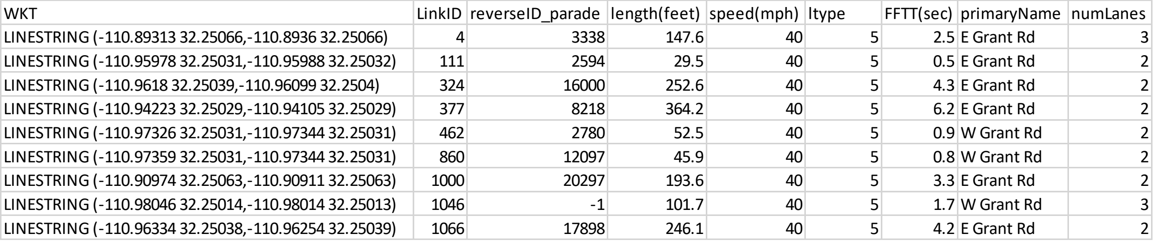

San Manuel Airport
Biosphere Two Landing Field
Pinal Airpark
Ironwood Forest National Monument
Coronado National Forest
Pusch Ridge Wilderness Area
Saguaro National Park (East) - Rincon Mountain District
Rincon Mountain Wilderness Area
Tucson Mountain Park
Ryan Field
I 10
I 19
Empire Ranch Airport
Coronado National Forest

Figure 4 Road Geometry Data, Tucson

The example of road geometry data is shown in Table 3 and detailed data description is shown in Table 4.

Table 3 Example of Road Geometry Data



WKT	LinkID	reverseID_parade	length(feet)	speed(mph)	ltype	FFTT(sec)	primaryName	numLanes
LINESTRING (-110.89313 32.25066,-110.8936 32.25066)	4	3338	147.6	40	5	2.5	E Grant Rd	3
LINESTRING (-110.95978 32.25031,-110.95988 32.25032)	111	2594	29.5	40	5	0.5	E Grant Rd	2
LINESTRING (-110.9618 32.25039,-110.96099 32.2504)	324	16000	252.6	40	5	4.3	E Grant Rd	2
LINESTRING (-110.94223 32.25029,-110.94105 32.25029)	377	8218	364.2	40	5	6.2	E Grant Rd	2
LINESTRING (-110.97326 32.25031,-110.97344 32.25031)	462	2780	52.5	40	5	0.9	W Grant Rd	2
LINESTRING (-110.97359 32.25031,-110.97344 32.25031)	860	12097	45.9	40	5	0.8	W Grant Rd	2
LINESTRING (-110.90974 32.25063,-110.90911 32.25063)	1000	20297	193.6	40	5	3.3	E Grant Rd	2
LINESTRING (-110.98046 32.25014,-110.98014 32.25013)	1046	-1	101.7	40	5	1.7	W Grant Rd	3
LINESTRING (-110.96334 32.25038,-110.96254 32.25039)	1066	17898	246.1	40	5	4.2	E Grant Rd	2

Table 4 Data Description of Road Geometry Data

Attributes	Description
WKT	Shape of the link
LinkID	ID of link (unique)
ReverseID_parade	ID of link that is in the reverse direction (unique)
Length(feet)	Length of the link in feet
Speed(mph)	Speed limit of the link in miles per hour
Ltype	The road type of the link, e.g., Arterial
FFTT(sec)	Free flow travel time to pass the link
PrimaryName	The name of the road that the link is on
numLanes	The number of lanes of the link

3.3 Data Preprocessing

The raw data was processed in several steps to be used in travel time estimation. First, since our research area focused only on Grant Road from I-10 to Swan Road, original data was selected only in the research area. In addition, low accuracy data and some data that have missing values was cleaned in order to increase the accuracy of the study. In addition, since our research area focused on Grant Road only from I-10 to Swan Road, the original data needed to be filtered by spatial location. Then, probe vehicle data was linked to road geometry by map matching. After that, road links were combined to corridor level. Finally, the movement table was built to estimate corridor travel time in different directions.

3.3.1 Data Selection

The original data covered the range of city. however, only data within the research corridor is needed. Data selection is a process that extract data that is related to our research from original dataset. There are two kinds of data were selected: road geometry data and probe vehicle data. Road geometry data was selected manually in QGIS[2] and only those links within the research area were selected from original data. Probe vehicle data was selected by trips. Only the trip that passed one or more links within the research area was selected. Selected road geometry data is shown in Figure 5, where the different colors represent different links. After data selection, there were 247 links within the research area.

As Figure 5 shows, the length of links is relatively short. In contrast, the research objective of travel time study is the corridor between adjacent intersections. The corridor travel time can be calculated by adding the travel time of all the links. However, since the sampling rate is about one point per second, the accuracy will be low. Thus, links need to be converted to corridors and the logical relations of the corridors needs to be rebuilt.

[2] QGIS is an open-source GIS toolkit. (http://www.qgis.org/en/site/)

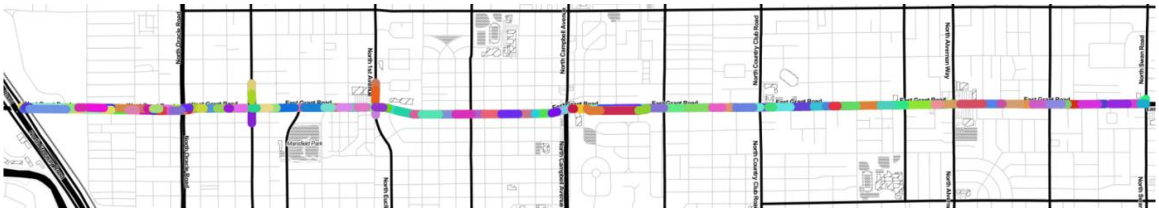


Figure 5 Selected Road Geometry Data

3.3.2 Data Cleaning

The GPS data should satisfy the following requirements in order to be used in our study:

(1) High sampling rate. Sampling rate means the GPS data collection frequency. This study required a high sampling rate to maintain the accuracy of the corridor travel time estimation. The recommended sampling rate is one GPS point per second.

(2) High spatial accuracy. The GPS accuracy here refers mainly to the accuracy with respect to the location of the vehicles because corridor travel time is estimated mainly by location and timestamp. The longer the study corridors, the lower the accuracy of GPS data required. When collecting data, the accuracy of GPS data is not guaranteed and the reasons can be varied. Low data accuracy can be caused by the effect of tall buildings, imperfections of the GPS module, satellite positions, etc. Since GPS data is collected by apps, there are built-in functions to know the accuracy of that GPS data, e.g., location.getAccuracy() function in the Android system and similar functions in the iOS system. The accuracy value returned by those functions means the accuracy is guaranteed to be within X distance with a 68% confidence level. In this study, only data having an accuracy of 32 feet or lower was used.

(3) Large historical dataset. A large historical dataset is required for the K-NN algorithm to find the trajectories that are most similar to an incomplete trajectory. If there

is insufficient historical data, the accuracy of the k-NN algorithm may be affected or the algorithm may not be conducted at all. The details are discussed in the following chapters.

According to the requirements, low accuracy GPS points (accuracy larger than 32) and GPS points that have missing value and abnormal values were cleaned. After that, the sampling rate was checked at the trip level. The accuracy distribution is shown in Figure 6.

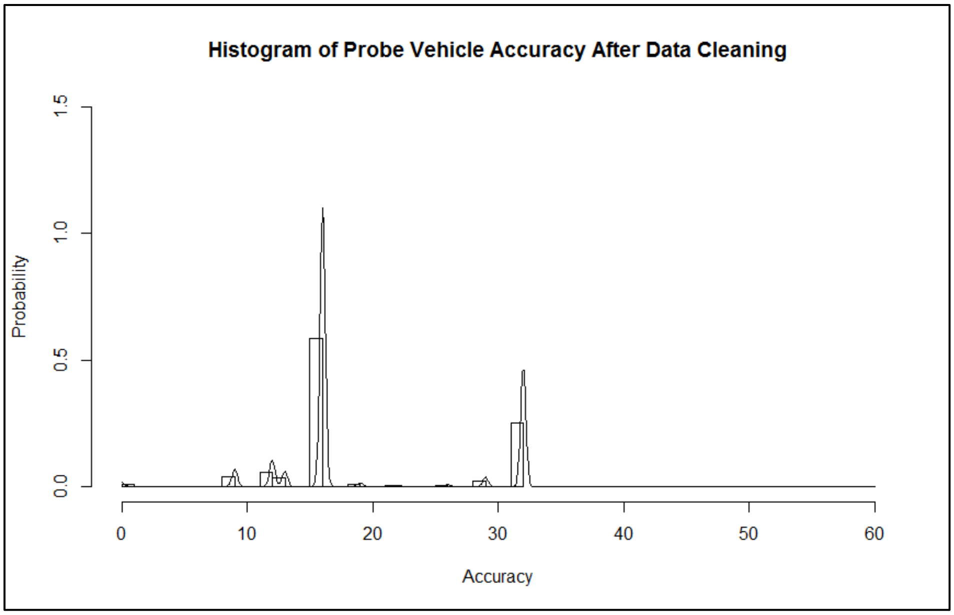


Figure 6 Histogram of Probe Vehicle Accuracy After Data Cleaning

3.3.3 Map Matching

Map Matching is a process that pairs probe vehicle data with the road geometry data. Although probe vehicle data and road geometry data were matched by the data collection application during the data collection process, issues arose due to the real-time nature of data collection. Only location was considered in the original map matching process but speed and direction are also critical to the accuracy of the map matching process. The accuracy of the map matching process directly relates to the accuracy of travel time estimation since corridor travel time is the time difference between when the vehicle entered the corridor and when it left the corridor.

A hidden markov chain algorithm (Zheng et al., 2011) was used, and location, speed, the direction of movement of probe vehicle data was used in the map matching process. The final map matching accuracy was over 95%. The map matching accuracy was calibrated by partial selecting map matched probe vehicle data points and manually judged.

3.3.4 Link to Corridor

Since the purpose of the study is to estimate travel time from an upstream intersection to a downstream intersection, a.k.a. corridor travel time, the road geometry data was converted to corridor level. The relationship of links and corridors was manually created in QGIS (Quantum GIS Development Team, 2017). The converted road geometry data is shown in Table 5.

Table 5 Example of Corridor Data

Corridor ID	Corridor Name	Road Name	Link ID
1	Oracle Rd to Stone Ave	Grant Road	462 860 1490 2780 4270 6311 6721 7545 10938 12097 12652 13480 14290 20072 22050 22695 23305 25112 25286 26600 26781 27046
2	Stone Ave to First Ave	Grant Road	324 1066 2399 4607 4997 7567 8000 9171 10995 11863 12537 13149 13462 14863 16000 16200 16869 17099 17309 17898 18464 20974 21443 21458 24223 27491
3	Grant to Alturas	Stone Ave	5177 7395 9864 10310 17729 19677 24504 27788
4	Grant to Sahuaro	Stone Ave	11009 12538 15171 25826

5	Grant to Alturas	1st Ave	2098 3243 4465 6136 14591 24472

3.3.5 Corridor to Movement

In order to estimate travel time in different directions, a movement table was built. The movement table is shown in Table 6; a movement is the sequence of corridors to justify, or explain, the movements of the probe vehicles. If the upstream and downstream corridors are known, the movement of probe vehicle can be justified. For example, if a vehicle passed corridor 1, corridor 8 and corridor 9 continuously, we know the vehicle is moving westbound.

Table 6 Example of Movement Data

Movement ID	Corridor Name	Direction	From Corridor ID	To Corridor ID	Self-Corridor ID
1	Fairview Ave to Oracle	WB	1	8	9
2	Oracle Rd to Stone Ave	WB	2	9	1
3	Stone Ave to First Ave	WB	6	1	2
4	First Ave to Park	WB	10	2	6
5	Park to Mountain	WB	11	6	10

3.4 Processed data

After data preprocessing, there are 10,054,193 GPS points collected from 9849 trips that were used in the case study. All the GPS points are within the research area and at least have an accuracy of 32 feet. Probe Vehicle data was matched with links, which finally converted to corridors. Processed Data is shown in Figure 7 with GPS data categorized by corridors and each color in the figure means different corridors.

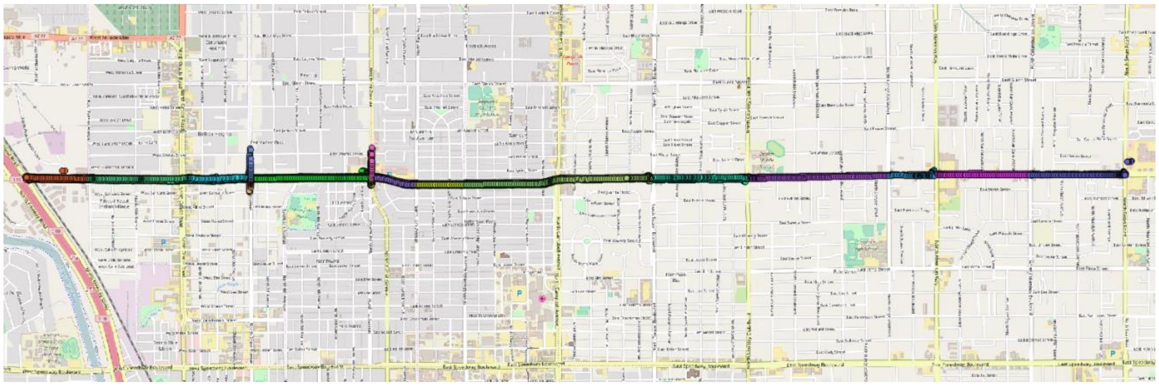


Figure 7 Processed Probe Vehicle Data, Categorized by Corridors

An example of processed data is shown in Table 7 and detailed data description is shown in Table 8.

Table 7 Example of Processed Data

MetropianID	ReservationID	TrajectoryID	LinkID	Longitude	Latitude	Altitude	direction	unixtime	speed	accuracy	MMLinkID	CorridorID
3593	73	4707	6839	-110.940369	32.250275	2420	90	1437554419363	36	16	6839	12
3593	73	4708	6839	-110.940193	32.250275	2421	90	1437554420045	36	16	6839	12
3593	73	4709	6839	-110.940018	32.250275	2422	89	1437554421067	37	16	6839	12
3593	73	4710	6839	-110.939842	32.250271	2421	90	1437554422045	36	16	6839	12
3593	73	4711	6839	-110.939674	32.250271	2420	89	1437554423052	37	16	6839	12
3593	73	4712	6839	-110.939491	32.250271	2420	89	1437554424039	37	16	13262	12
3593	73	4713	6839	-110.939308	32.250267	2418	89	1437554425042	38	16	13262	12
3593	73	4714	13262	-110.939117	32.250263	2418	89	1437554426053	39	16	13262	12
3593	73	4715	13262	-110.938934	32.250267	2416	89	1437554427049	39	16	13262	12
3593	73	4716	13262	-110.938751	32.250267	2416	89	1437554428064	39	16	13262	12

Table 8 Data Description of Processed Data

Attributes	Description
MetropianID	ID of app users (unique)
ReservationID	ID of trips (unique)
TrajectoryID	ID of GPS points (unique)
LinkID	ID of the link that the GPS point was on. The value was given when GPS data was collected.
Longitude	Longitude
Latitude	Latitude
Altitude	Altitude

Direction	Moving direction of the vehicle. The value increases clockwise from the north direction and ranges from 0 to 360
Unixtime	Unixtime timestamp of the GPS points
Speed	Speed of the vehicle (MPH, mile per hour)
Accuracy	Spatial accuracy (feet), means error that reported coordination is within.
MMlinkID	The link ID that the GPS point was matched
CorridorID	The corridor ID that GPS point was matched

4. METHODOLOGY

The framework of link travel time estimation is shown in Figure 8. Raw probe vehicle trajectory data was pre-processed before any further analysis. As mentioned in the section of “Data Description” (section 3.2), the data pre-processing consists of data selection, data cleaning, map matching, link to corridor and corridor to movement.

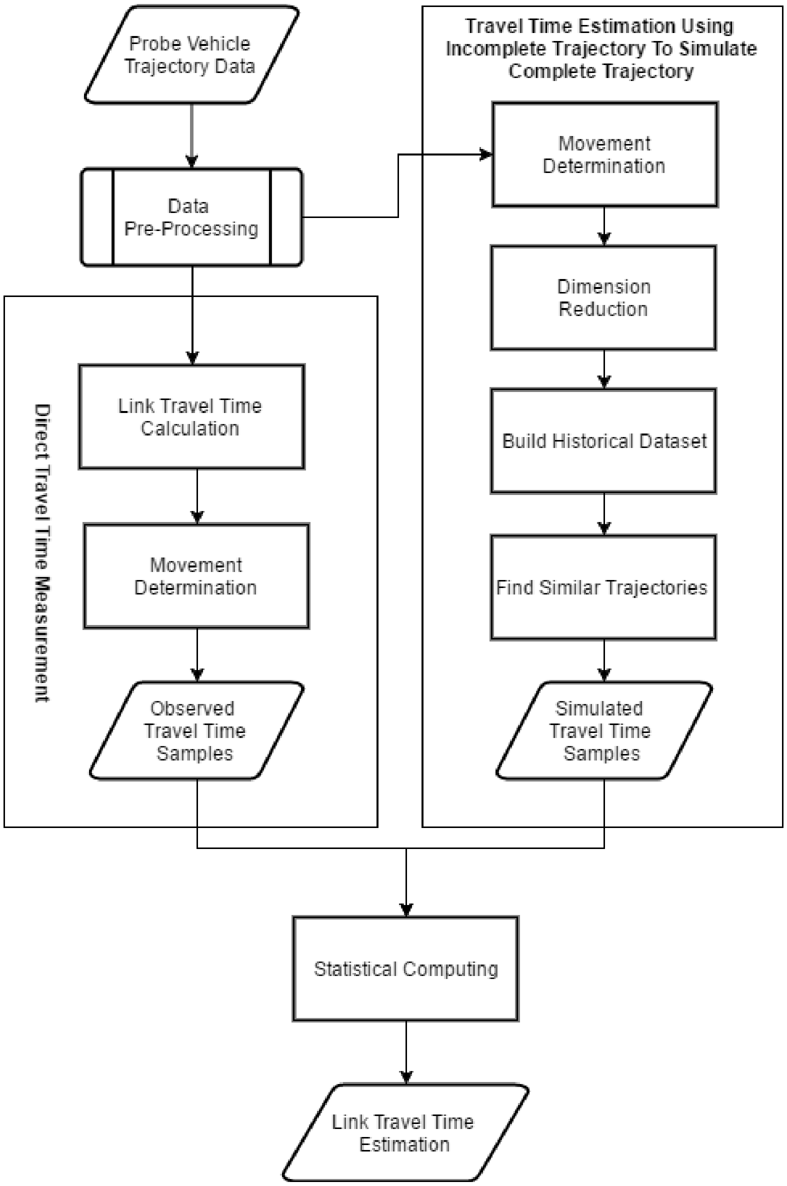

Probe Vehicle Trajectory Data
Data Pre-Processing
Direct Travel Time Measurement
Link Travel Time Calculation
Movement Determination
Observed Travel Time Samples
Travel Time Estimation Using Incomplete Trajectory To Simulate Complete Trajectory
Movement Determination
Dimension Reduction
Build Historical Dataset
Find Similar Trajectories
Simulated Travel Time Samples
Statistical Computing
Link Travel Time Estimation

Figure 8 Study Framework

The travel time estimation framework was mainly consisted of two modules: direct travel time measurement and travel time estimation using incomplete trajectories to simulate complete trajectories. Link travel time was estimated by combining directly measured travel time samples and travel time samples that estimated from incomplete trajectories. Finally, statistical indicators (mean, average, and confidence interval) can be calculated based on these samples. Note that the framework can work with use of only the direct travel time measurement module. The travel time estimation using incomplete trajectories to simulate complete trajectories module can increase the sample size but the module is not required.

A probe vehicle trajectory stems from a series of time-stamped points, each of which contains the position information. The trajectory that has passed through the whole study link is a “complete trajectory” for that link. However, sometimes a probe vehicle only passes part of the link, in which case travel time cannot be directly calculated because there is no data at the downstream location or upstream location. The trajectory in this condition is called “incomplete trajectory.” The comparison of a complete trajectory and an incomplete trajectory is shown in Figure 9. For better visualization, probe vehicle trajectories are shown by lines but they are actually series of GPS points. The red line that passed both the downstream and upstream locations is a complete trajectory. There are three types of incomplete trajectories: a trajectory that enters the link from the upstream location and leaves before downstream location (type 1); a trajectory that enters the link after upstream location and leaves before downstream location (type 2); and a trajectory that enters the link after the upstream location and leaves after passing the downstream location (type 3). In urban travel time estimation, a study link usually starts from an intersection and ends by the next intersection so that a vehicle typically experiences an accelerating process and a smooth traveling process and may or may not be affected by the downstream intersection. Since different types of incomplete trajectory travel different parts of the link, their traffic information contained in the data is different.

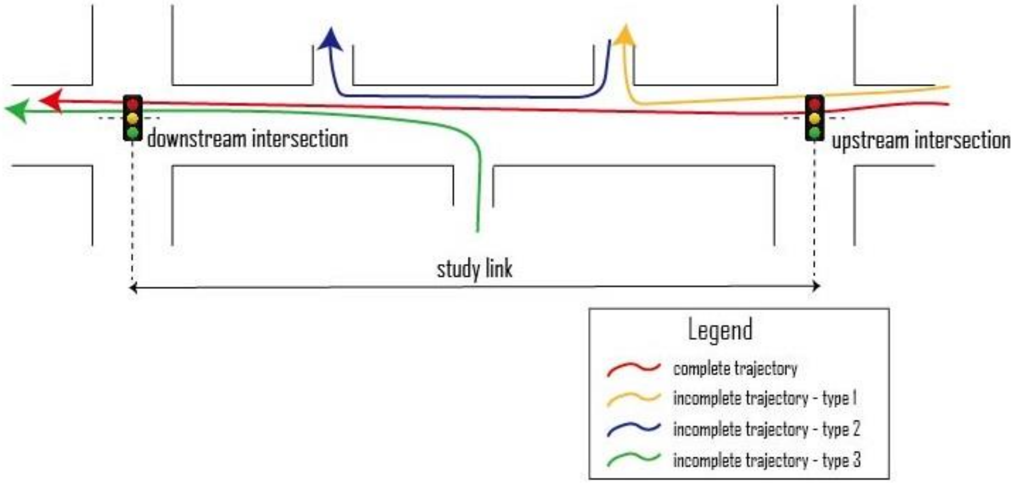

downstream intersection
upstream intersection
study link
Legend
complete trajectory
incomplete trajectory - type 1
incomplete trajectory - type 2
incomplete trajectory - type 3

Figure 9 Complete Trajectory and Incomplete Trajectory

An incomplete trajectory cannot be used directly to calculate link travel time, but it does contain traffic information. To increase the sample size of travel time estimation, there is another module to calculate travel time using incomplete trajectories. An incomplete trajectory was compared with historical complete trajectories to find several of the most similar complete trajectories. The simulated travel time of incomplete trajectory was represented by its similar complete trajectories, whose travel time can be directly calculated. The assumption here was the incomplete trajectory would experience similar traffic as that experienced by its similar complete trajectory.

The framework in Figure 8 mainly addressed the following technical issues:

(1) The study proposed a complete framework of travel time estimation using probe vehicle data. The framework can analyze link travel time by movements. Comparing with traditional link travel time, in which link travel time is the aggregation of all the movements, link travel time obtained using the proposed framework can provide more precise evaluation of the link and provide more accurate travel time information for travelers.

(2) The method proposed in the framework can increase link travel time samples size by a large extent. The framework discussed the potential and implementation of travel time calculation using incomplete trajectories, which have never been used before. Since incomplete trajectories can constitute a huge amount of a public probe vehicle dataset, using the incomplete trajectories can greatly increase the information available for travel time estimation.

To better illustrate the framework used in this study, the direct link travel time measurement module is introduced in the following section of this chapter. Then, in the next section, the principle and module of travel time estimation using incomplete trajectories to simulate complete trajectories is explained. The sensitivity analysis of the travel time estimation using incomplete trajectories to simulate complete trajectories module is presented in the last section of this chapter. In the sensitivity analysis, different parameters and input data are tested to verify the applicability and the accuracy of the algorithm. The pseudo code of direct travel time measurement and travel time estimation using incomplete trajectory is in Appendix B.

4.1 Direct Link Travel Time Measurement

4.1.1 Link Travel Time measurement

Since different definitions of links will lead to different travel time results, the definition of a link in this study was explained first. As shown in Figure 10, there are two intersections: intersection A and intersection B. Intersection A is the upstream intersection and intersection B is the downstream intersection. The link is defined as the road segment from the center of the upstream intersection (intersection A) to the center of the downstream intersection (intersection B) going in one direction. There are three trajectories shown and all of them traveled the whole link. The first GPS points after the vehicle enters the link (hereinafter, the **F**irst point) is defined as point F for each trajectory i. The last GPS points before the vehicle leaves the link (hereinafter, the **L**ast point) is defined as point L for each trajectory i. The movement-based travel time for each trajectory TT_i is defined as:

$$TT_i = t_{i,L} - t_{i,F} \tag{1}$$

where t is the timestamp of GPS point, i represents different trajectories, the first GPS point after the vehicle enters the link is defined as point F and the last point before the vehicle leaves the link is defined as Point L.

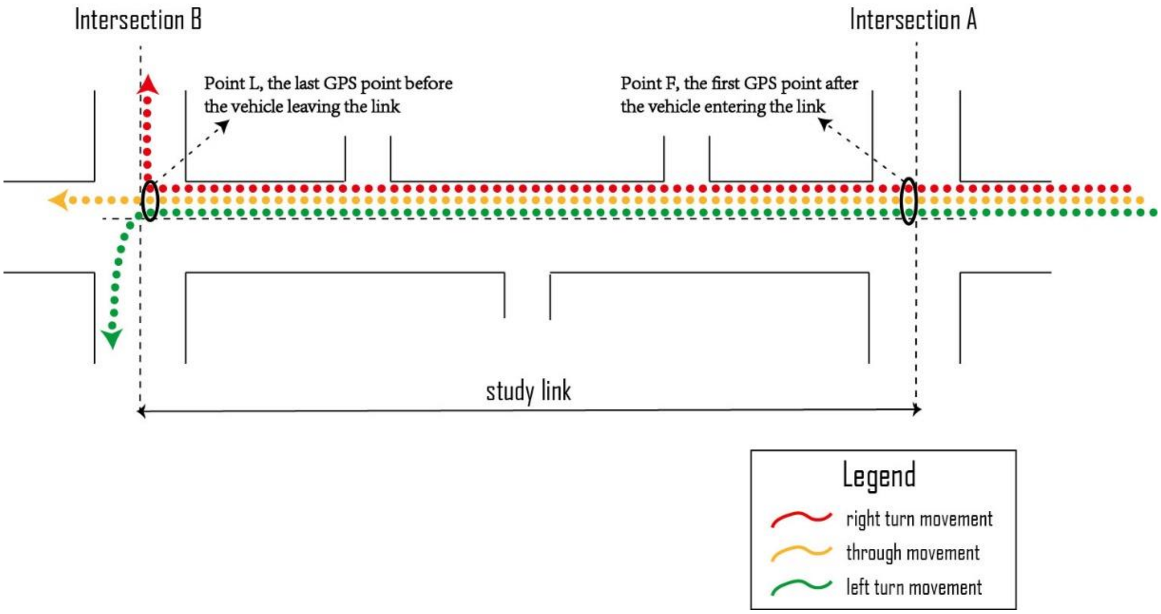

Intersection B
Intersection A
Point L, the last GPS point before the vehicle leaving the link
Point F, the first GPS point after the vehicle entering the link
study link
Legend
right turn movement
through movement
left turn movement

Figure 10 Complete Trajectory of Different Movements

As link travel time is the time difference between the point F and the point L, then it is necessary to know which points are the first and the last point. Therefore, the relationship between the GPS points and the link needs to be built. In the map matching part of data pre-processing, each point was matched with the link it belongs to. Until now, each trajectory has a sequence of link IDs that represent the links that the probe vehicle has traversed. However, even the most accurate map matching algorithm is not entirely accurate and the error of map matching needs to be eliminated to prevent error in travel time calculation. In addition, as mentioned before, direct travel time measurement can use only the trajectory that passed the whole link; thus, the completeness of trajectory needs to be verified. To remove the map matching error and verify the completeness of the trajectory in a link, the following process was taken:

(1) Map matching errors were eliminated. Map matching error usually happens where links intersect and a point is matched with another link that is close to the correct

link. Map matching error can be eliminated by checking map matching continuity since map matching error only happens occasionally. For a map matched trajectory, if an unfamiliar link is matched with only very few points, it is highly possible that it is a map matching error and erroneous points are assigned with a correct value.

(2) A complete probe vehicle trajectory must enter a link from its upstream successive links and exit through the downstream successive links; otherwise the trajectory is not complete and it was removed.

(3) The distance between the upstream point of the link and the first point should be within a threshold value (e.g., 10 feet), so it goes with the last point. This is to prevent error that may be generated from a low sampling rate. The trajectory was abandoned if it cannot satisfy this criterion.

After eliminating the map matching error and checking the completeness of the trajectory, an accurate link travel time sample can be calculated by Equation (1). The last step was to determine the movement of this travel time sample. The trajectory was compared with a predefined movement table. A movement was consisted of a series of link IDs. For example, as shown in the Figure 11, if a vehicle pass link 1, link 4 and link 6 consecutively and it pass a full link 4, then the vehicle makes a left turn. Similarly, if the vehicle pass link 1, link 4 and link 7 consecutively, then the calculated link travel time belongs to through movement.

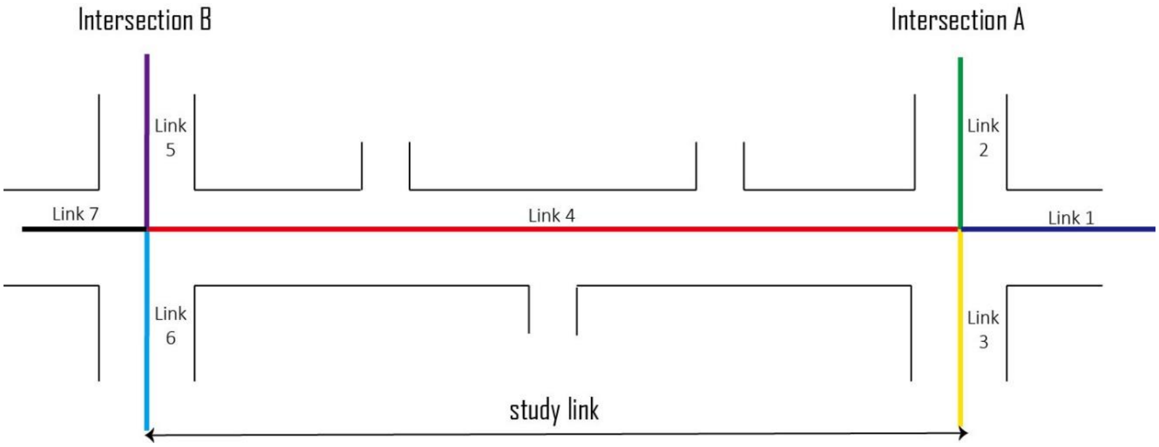


Figure 11 Movement Determination

4.2 Travel Time Estimation Using Incomplete Trajectory to Simulate Complete Trajectory

A widespread problem in most of the current probe vehicle datasets is a low penetration rate (in other words, small sample size). A public vehicle dataset is mostly contributed by application users or specific groups of people (e.g., taxi drivers) and the penetration rate of data fluctuates largely based on data contributors' spatial and temporal characters. There are two ways to increase the sample size of link travel time estimation: collecting more data or better utilizing the current data. However, as it is not economical efficient to collect more data over a long period or in a large region, extracting more information from current dataset becomes a better choice.

In the travel time estimation using incomplete trajectories to simulate complete trajectories module, the incomplete trajectories, which were usually discarded before, were instead utilized to generate additional link travel time samples. The main idea behind the module is that probe vehicle trajectories under the same traffic condition have similar characters. Assume a virtual probe vehicle is traveling the whole link at the same time as the probe vehicle of an incomplete trajectory. The trajectory of the virtual probe vehicle has characteristics similar to those of the incomplete trajectory since they are subject to the same traffic. An incomplete trajectory is compared with historical dataset to find the most similar complete trajectories. Thus, the travel time for the virtual probe vehicle to pass the whole link is represented using these similar complete trajectories.

4.2.1 K-Nearest Neighbor Algorithm

The K-Nearest Neighbors algorithm (k-NN) is a non-parametric technique that has the assumption that similar objects have similar characters, which is as same as the assumption in transportation field. Finding several similar complete trajectories for an incomplete trajectory, link travel time for the incomplete trajectory can be replaced by these complete trajectories. Since the incomplete trajectory has similar characters as its similar complete trajectories, the traffic environment is similar.

4.2.2 Dimension Reduction

Probe vehicle trajectory needs to be processed before implementing the k-NN method. Probe vehicle data has many attributes and not all of them are meaningful in travel time estimation. Only longitude, latitude and timestamp were selected as the inputs of the k-NN algorithm. The longitude and latitude attributes of probe vehicle data represent only the shape of the trajectory and they are not related with travel time. In order to build the relationship between trajectory data and link travel time, three-dimensional probe vehicle trajectory data was reduced to two-dimensional distance-time data, where distance is the distance between GPS points and the upstream intersection and timestamp is the timestamp for each of the GPS points (e.g., 2015-03-01 12:01:34).

The dimension reduction for six hypothetical trajectories is shown in Figure 12, each with different characteristics. The green trajectory is a complete trajectory under free flow condition and its travel time is free flow travel time. The yellow trajectory is a complete trajectory in non-peak hour and it is slightly affected by the downstream intersection. The red line is a complete trajectory in peak hour. Compared with the trajectory in non-peak hour, it has a lower driving speed, a longer queue length, and a longer delay. The blue line, black line and brown line represent three types of incomplete trajectories. Note that each dot in the diagram is a GPS point and they are connected by a line. The line is smoothed using spline method in case the interval between consecutive GPS points is too long.

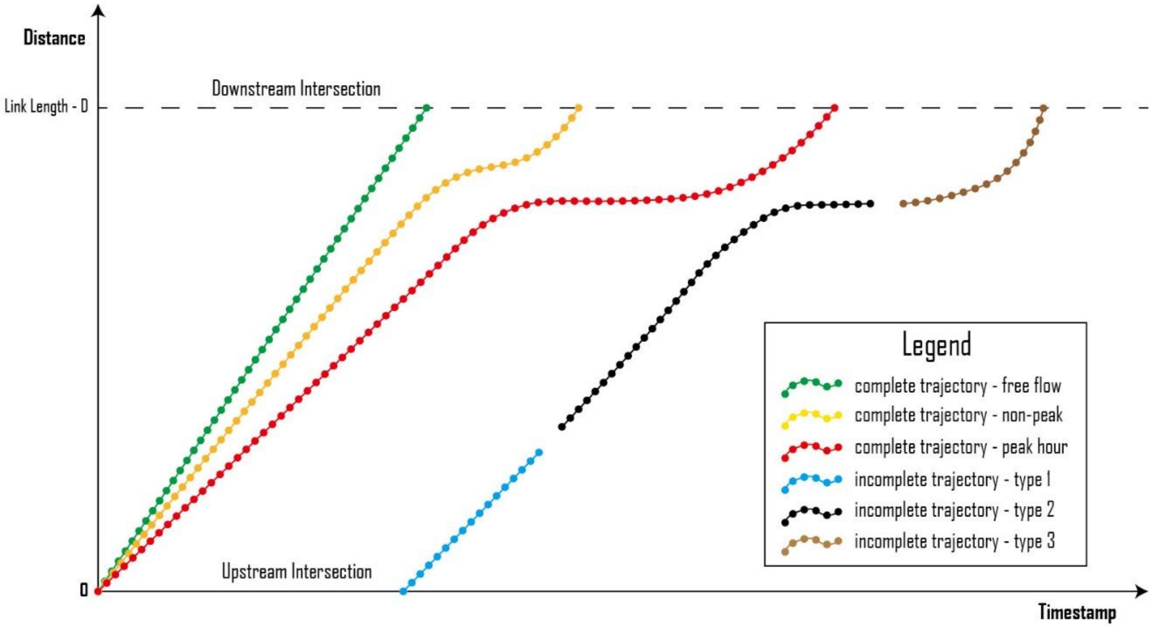

Distance
Link Length - D
Downstream Intersection
0
Upstream Intersection
Timestamp
Legend
complete trajectory - free flow
complete trajectory - non-peak
complete trajectory - peak hour
incomplete trajectory - type 1
incomplete trajectory - type 2
incomplete trajectory - type 3

Figure 12 Dimension Reduction Result of Different Trajectories

Figure 13 shows the time and distance diagram of real world complete trajectories and incomplete trajectories. There were 955 complete trajectories and 40 incomplete trajectories. In the figure, X-axis shows the time spent after vehicle entering the link and

Y-axis shows the distance from upstream intersection. Although complete trajectories may not share the same origin point with incomplete trajectories since they may enter the link by different location, however, the derivative (speed) of the complete trajectories and incomplete trajectories may similar at the same location. The similarity of the derivative of trajectories can reflect the traffic condition of the vehicle has experienced since if derivative of two trajectories are same everywhere then these two trajectories are identical. This similarity is used to find similar complete trajectories of incomplete trajectory, then link travel time of incomplete trajectory can be inferred from its similar complete trajectories.

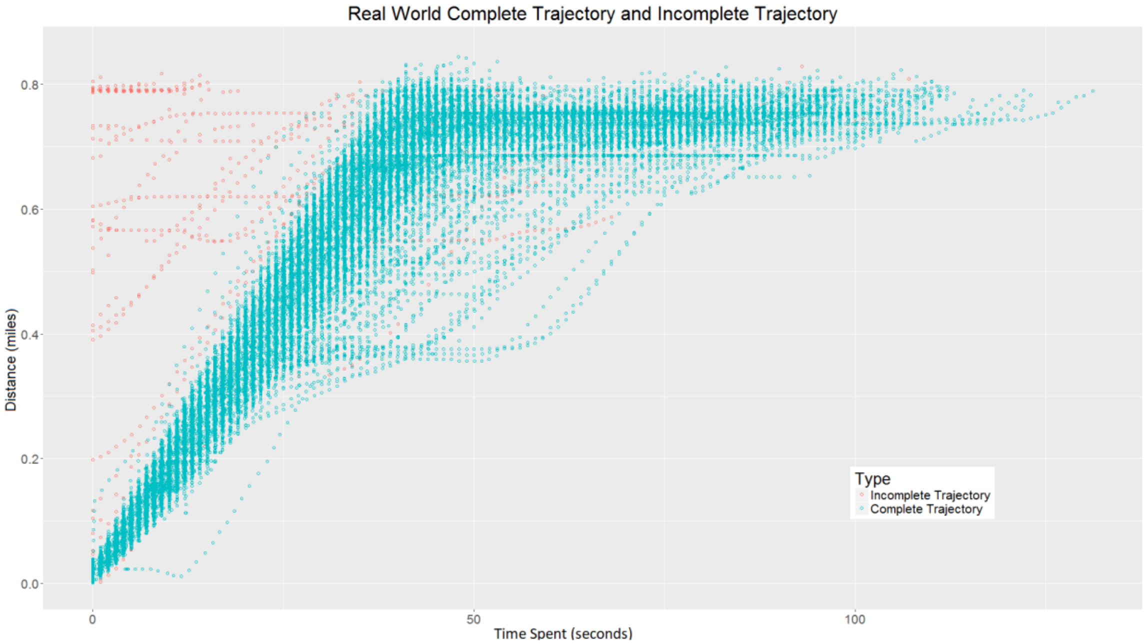


Figure 13 Real World Complete Trajectory and Incomplete Trajectory

4.2.3 K-Nearest Neighbors Regression

k-NN regression algorithm map patterns to continuous labels. The problem in regression is to predict labels $y' \in \mathbb{R}^d$ for new patterns $x' \in \mathbb{R}^q$ based on a set of N observations, i.e., labeled patterns $\{(x_1, y_1), \ldots, (x_n, y_n)\}$. For an unknown pattern x', k-NN regression computes the mean of the function values of its K-nearest neighbors:

$$f_{k-NN}(x') = \frac{1}{K} \sum_{i \in \mathcal{N}_k(x')} y_i \tag{2}$$

where $\mathcal{N}$ is neighborhood set, set $\mathcal{N}_k(x')$ containing the indices of the k-nearest neighbors of x' (Kramer, 2013).

Here is how Equation (2) used in our study. As mentioned before, we aimed to estimate link travel time when the vehicle of incomplete trajectories travelling on the link. Since incomplete trajectory not pass the whole link, link travel time cannot be directly calculated. However, link travel time of complete trajectory can be measured. Assuming the vehicle of incomplete trajectory pass the whole link under the same traffic, it has a simulated link travel time, the simulated link travel time is $f_{k-NN}(x')$. Using Equation (2), the simulated link travel time of the incomplete trajectory can be represented by the travel time of its k most similar complete trajectories, which are y in Equation (2). If the link travel time of the k most similar complete trajectories are $y_{i1}, y_{i2}, y_{i3}, y_{i4}, y_{i5}$ separately, the simulated link travel time of the incomplete trajectory is $\frac{(y_{i1}+ y_{i2}+y_{i3}+y_{i4}+y_{i5})}{5}$.

The distance between incomplete trajectory and complete trajectories are used to find nearest neighbors. The procedure to calculate the distance between an incomplete trajectory and a complete trajectory is shown in Figure 14. The distance between an upstream point and a downstream point is L; in other words, the length of the study link is L. There are two trajectories, the longer one is a complete trajectory and the shorter one is an incomplete trajectory. The study link is divided into n segments (n is 8 in Figure 14) on average and each segment has a length of $\frac{L}{n}$. The complete trajectory passed all the segments and the incomplete trajectory passed 5 segments in Figure 14, 3 of which were fully passed. T_i is the segment travel time for segment i that complete trajectory has fully passed, and t_i is the segment travel time for segment i that incomplete trajectory has fully passed. Segment travel time is only calculated when a trajectory is fully passed. For example, the red line in Figure 14 is the trajectory that did not fully pass any segments and those red portions are not used to calculate segment travel time.

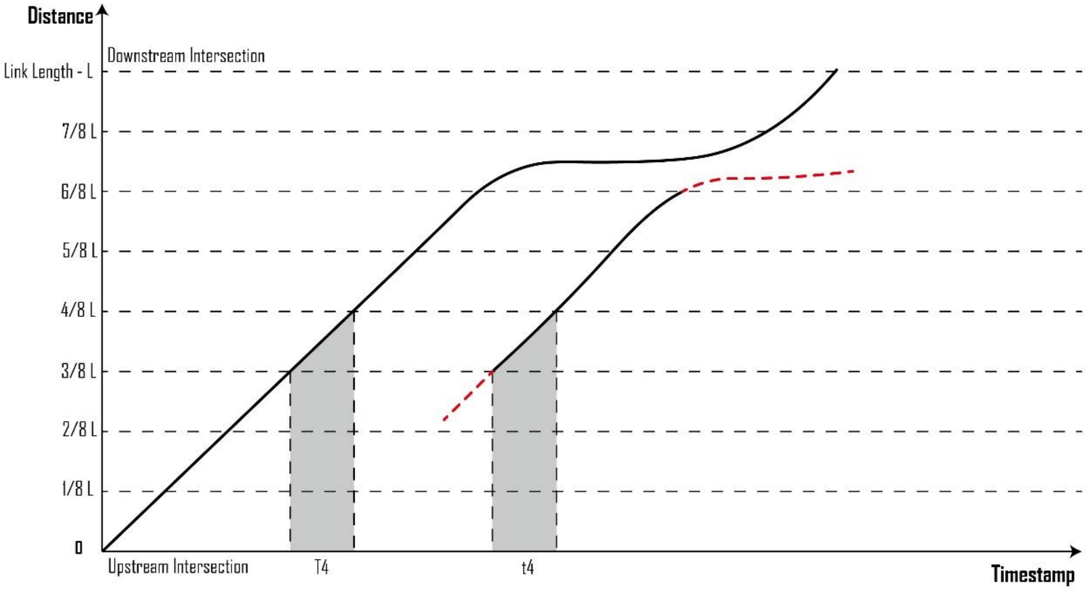


Figure 14 Trajectory Similarity Calculation

The distance between two trajectories S is

$$S = \sum_{i \in M}(T_i - t_i)^2 \qquad (3)$$

Where S is the distance between two trajectories, T_i is the time interval for a complete trajectory to pass the i_{th} segment*;* t_i is the time interval for an incomplete trajectory to pass the i_{th} segment*;* i is the sequence of segments; and M is the set of segments that an incomplete trajectory has fully passed.

Here is an example of using Equation (3) to calculate the distance between an incomplete trajectory and a complete trajectory. As shown in Figure 14, assume a link has a length of 1000 feet and it is divided into 8 segments separately. The travel time of a complete trajectory to pass each segment are 2.3s, 3s, 2.4s, 2.3s, 2.6s, 2.4s, 5s, 4.3s separately. There is an incomplete trajectory passed 5 segments and 3 out of 5 segments are fully passed, the travel time of the incomplete trajectory to fully pass the segments are 2.6s, 3s, 4s, separately. The distance of incomplete trajectory and complete trajectory is $(2.6-2.3)^2+\ (3-2.6)^2+\ (4-2.4)^2=2.81$.

4.2.4 Parameters and Input Data

Many factors can affect the performance of the algorithm. These can be summarized in two categories: algorithm parameters and input data. Algorithm parameters can be manually selected, and different parameters may adapt to different analysis purposes. Input data cannot be selected, but different types of input data can lead to different results.

The algorithm parameters mainly include the number of similar samples k and the number of divided link segments n. The number of similar samples are mainly correlated with the variance and bias of the estimator. If the number of similar samples is small, then the estimator will have a large bias but a small variance. If the number of similar samples is large, then the estimator will have a small bias and large variance. There is always a trade-off between variance and bias and the appropriate k depends on the project requirements.

Another parameter in the k-NN algorithm is the number of divided link segments n. As mentioned before, incomplete trajectories are divided into several segments and the part that not passed any full segment is removed (see the red part in Figure 14). When a vehicle is entering a link or leaving a link, it always has an adaptive process. In this adaptive process, vehicles need to adjust their speed to adapt to the new link and this process, i.e., accelerating or braking, and it likely does not reflect the traffic status for the link. The trajectory of this process needs to be removed to avoid systematic error. More segments can provide better data granularity but less of the trajectory in the adaptive process is removed, which may induce more systematic error.

There are three key characteristics of input data:

(1) Length of an incomplete trajectory. The length of an incomplete trajectory is directly corrected to the dimensions of the data. The longer the trajectory, the higher the chance that a found trajectory is similar.

(2) Incomplete trajectory types. As mentioned before, different types of trajectories may contain different information. For example, incomplete trajectory type 3 may reflect the effect of intersection delay since it has the queueing information.

(3) Time of day. Feng et. al (2013) compared the distribution of link travel time during peak hour and non-peak hour and they found that the distribution of travel time is different for different times of day.

4.3 Sensitivity Analysis

Sensitivity analysis can help engineers and agencies understand algorithm performance in different scenarios so that best parameters can be chosen and the prediction error can be approximated. To evaluate the k-NN module under different parameters and input data, leave-one-out cross validation and variable control method was used. Several experiments were designed to evaluate algorithm performance in different scenarios. Since there are many factors that may affect the result of the algorithm – such as the number of similar samples and the length of incomplete trajectory – the cross-validation process controlled only one factor at a time and kept all the other factors the same.

Cross validation is a model evaluation method that is commonly used to assess the stability of a parameter estimate, the accuracy of a classification algorithm, the adequacy of a fitted model, and in many other applications. The principle of cross validation is to evaluate the model or parameters multiple times using different data to avoid the estimator being affected by over-fitting. Cross validation divides the training set into a set of n equal-sized groups. For each group, cross validation uses the other $(n - 1)$ groups for training and that group for testing; there are, thus, n rounds in total. Leave-one-out cross validation (LOOCV) is a special case of cross validation: the model is trained on all the data except for the one point and a prediction is made for that point. The number of LOOCV rounds is the same as the number of complete trajectories in the dataset. Compared with other cross validation methods, LOOCV can reduce the bias of the estimator.

Theoretically, there is no ground truth for an incomplete trajectory since the link travel time for the incomplete trajectory cannot be calculated. However, we developed a method to evaluate the algorithm, by using an incomplete trajectory generated from a complete trajectory as the input. The complete trajectory was cut-off to become an incomplete trajectory to simulate real world condition. Since link travel time can be calculated from a complete trajectory and the incomplete trajectory has been converted from a complete trajectory, the ground truth is the travel time of the complete trajectory before conversion. Using this method, the performance of the algorithm can be evaluated. The dataset used in LOOCV is a historical complete trajectory dataset. In each round of LOOCV, one complete trajectory is converted into an incomplete trajectory and this incomplete trajectory is used as the input of the algorithm.

The input of the algorithm is the incomplete trajectory that converted from a complete trajectory and the output of the algorithm is the average travel time of complete trajectories that are similar to the converted incomplete trajectory. Since the incomplete trajectory is cut off from the complete trajectory, the ground truth is the link travel time of the original complete trajectory. Two measures of accuracy were used to verify the algorithm's performance: mean absolute error (MAE) and mean absolute percentage error (MAPE). MAE shows the average error of each round in LOOCV. Since link travel time is related to link length, MAE shows an average time difference between algorithm output and ground truth but it cannot reflect the performance difference between links. MAPE shows the error as a percentage and the performance can be compared between links. The definitions of the two measures are shown in Equations (4) and (5).

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|g_i - e_i| \tag{4}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|g_i - e_i|}{g_i} \tag{5}$$

where N is the number of LOOCV times and, g_i is the ground truth of link travel time in the i_{th} LOOCV, and e_i is the estimated link travel time in i_{th} LOOCV.

5. RESULTS

As discussed, the link travel time estimation result is the combination of direct link travel time measurement and link travel time estimation using incomplete trajectory. In this chapter, the result of direct travel time measurement is introduced first. Then, the result of travel time estimation using incomplete trajectories to simulate complete trajectories is discussed. Afterwards, sensitivity analysis of k-NN algorithm under different parameters and input data is presented.

5.1 Direct Travel Time Measurement

Direct travel time measurement is the process that to measure the travel time using complete trajectory. Table 9 shows the distribution of direct measurement link travel time result in 2015. There were 10,072 link travel time samples distributed on 13 links. Eastbound Oracle Rd to Stone Ave, eastbound Stone Ave to First Ave, westbound First Ave to Park Ave, and westbound Stone Ave to First Ave are selected to calculate movement-based link travel time. Comparing the sample size of through movement, it can be found on the one hand that the sample size of through movement was much more than the sample size for turning movements. On the other hand, since the data was collected from the public, the ratios reflected the proportion of through movements and turning movements. On boundary conditions (red rows in Table 9), e.g., Freeway to Fairview Ave westbound through movement and Columbus to Swan Ave eastbound through movement, the sample size was low. The reason is that, when a vehicle leaving the study network, the movement of the vehicle is hard to identify since there is no data of the downstream intersection.

The boundary condition of a link is when link located at the edge of the road network or the link itself is a dead-end. Since there is no downstream link for boundary conditions, the movement of the vehicle is unknown. Figure 15 shows the boundary condition in the figure, when vehicle leaves the network, the movement of vehicle cannot be determined.

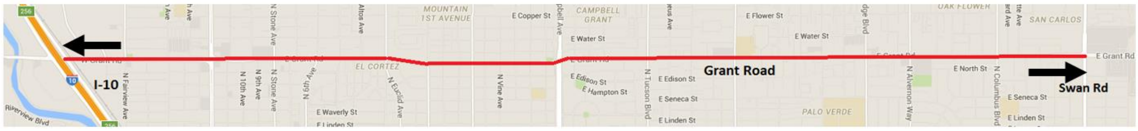


Figure 15 Boundary Conditions (Black arrow in the Figure)

Table 9 Distribution of Direct Calculated Link Travel Time Samples, 2015

Link Name	Direction	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Sum
Freeway to Fairview Ave	WB Through	0	0	0	1	0	0	1	0	0	0	13	1	16
Fairview Ave to Oracle	WB Through	15	3	24	74	66	36	42	38	32	32	35	26	423
Oracle Rd to Stone Av	WB Through	15	1	25	67	59	33	40	40	32	32	32	29	405
Stone Ave to First Ave	WB Through	13	0	18	58	39	28	30	30	23	17	29	15	300
First Ave to Park	WB Through	4	0	16	52	35	20	29	34	24	19	30	22	285
Park to Mountain	WB Through	5	0	25	64	57	36	31	46	33	28	32	43	400
Mountain to Campbell	WB Through	2	1	19	51	40	28	26	44	30	22	33	37	333
Campbell to Tucson	WB Through	3	1	22	45	29	25	30	48	34	27	46	37	347
Tucson to Country Club	WB Through	2	2	25	63	33	21	26	48	43	23	40	27	353
Country Club to Dodge	WB Through	2	1	24	64	31	23	27	48	47	31	42	34	374
Dodge to Alvernon	WB Through	1	1	27	72	42	36	25	53	61	44	54	45	461
Alvernon to Columbus	WB Through	1	1	24	72	31	29	25	35	55	37	45	43	398
Columbus to Swan	WB Through	1	1	36	89	50	28	28	52	63	43	49	53	493
Freeway to Fairview Ave	EB Through	19	5	68	140	101	72	75	66	69	68	88	80	851
Fairview Ave to Oracle	EB Through	20	5	63	130	100	71	74	59	57	64	82	77	802
Oracle Rd to Stone Av	EB Through	15	1	20	57	67	44	57	36	35	38	33	41	444
Stone Ave to First Ave	EB Through	12	1	15	39	40	23	28	22	21	28	38	29	296
First Ave to Park	EB Through	4	0	17	36	40	27	34	33	24	37	49	38	339
Park to Mountain	EB Through	3	0	12	50	58	36	28	34	23	39	58	46	387

Mountain to Campbell	EB Through	3	1	7	30	42	23	24	28	21	34	52	36	301
Campbell to Tucson	EB Through	3	2	18	31	35	23	23	29	26	27	49	33	299
Tucson to Country Club	EB Through	2	1	18	27	23	15	22	26	26	28	48	39	275
Country Club to Dodge	EB Through	2	1	19	22	21	14	24	31	27	23	47	40	271
Dodge to Alvernon	EB Through	0	2	19	30	30	22	31	32	34	32	56	40	328
Alvernon to Columbus	EB Through	0	1	15	31	33	16	31	27	29	29	52	37	301
Columbus to Swan	EB Through	0	0	0	1	1	1	5	7	3	3	17	4	42
Oracle Rd to Stone Av	EB Left	0	0	4	36	18	18	10	8	13	17	23	29	176
Oracle Rd to Stone Av	EB Right	1	0	3	14	1	0	0	1	2	1	5	2	30
Stone Ave to First Ave	EB Left	2	0	2	14	17	22	24	15	12	9	2	11	130
Stone Ave to First Ave	EB Right	0	0	1	5	9	4	7	0	1	0	4	0	31
First Ave to Park	WB Left	2	0	3	9	20	10	3	5	6	7	5	14	84
First Ave to Park	WB Right	0	0	5	7	7	8	2	4	1	5	3	8	50
Stone Ave to First Ave	WB Left	2	0	1	8	2	3	4	7	4	3	4	2	40
Stone Ave to First Ave	WB Right	0	0	1	2	1	0	1	0	0	0	1	1	7
Sum		**154**	**32**	**596**	**1491**	**1178**	**795**	**867**	**986**	**911**	**847**	**1196**	**1019**	**10072**
Link Name	**Direction**	**Jan**	**Feb**	**Mar**	**Apr**	**May**	**Jun**	**Jul**	**Aug**	**Sep**	**Oct**	**Nov**	**Dec**	**Sum**

5.2 Travel Time Estimation using Incomplete Trajectory to Simulate Complete Trajectory

As stated before, the k-NN regression method requires a large historical dataset. According to the sample distribution in Table 9, the sample size of turning movements was not sufficient to effectively implement the k-NN algorithm, so the k-NN module was applied only to through movements in the case study. Incomplete trajectories in November 2015 were used as the input and complete trajectories from January to October 2015 were used as the training set. The travel time and sample size comparison is shown in Figure 16 and Table 10. After the implementation of the k-NN algorithm, the sample size was increased 42% averagely and the maximum increase was 153%. There were two links that have no improvements, westbound through movement from First Ave to Stone Ave and eastbound through movement from Dodge Blvd to Alvernon Way. By analyzing these two links, we find that the performance of the algorithm is related to link geometry characteristics and land use around the link. Land use on the north side from First Ave to Stone Ave was mostly residential, which may explain the shortage of incomplete trajectories. The link of Dodge Blvd to Alvernon Way was very short and there were very few access points and this could be the reason why incomplete trajectories were not captured.

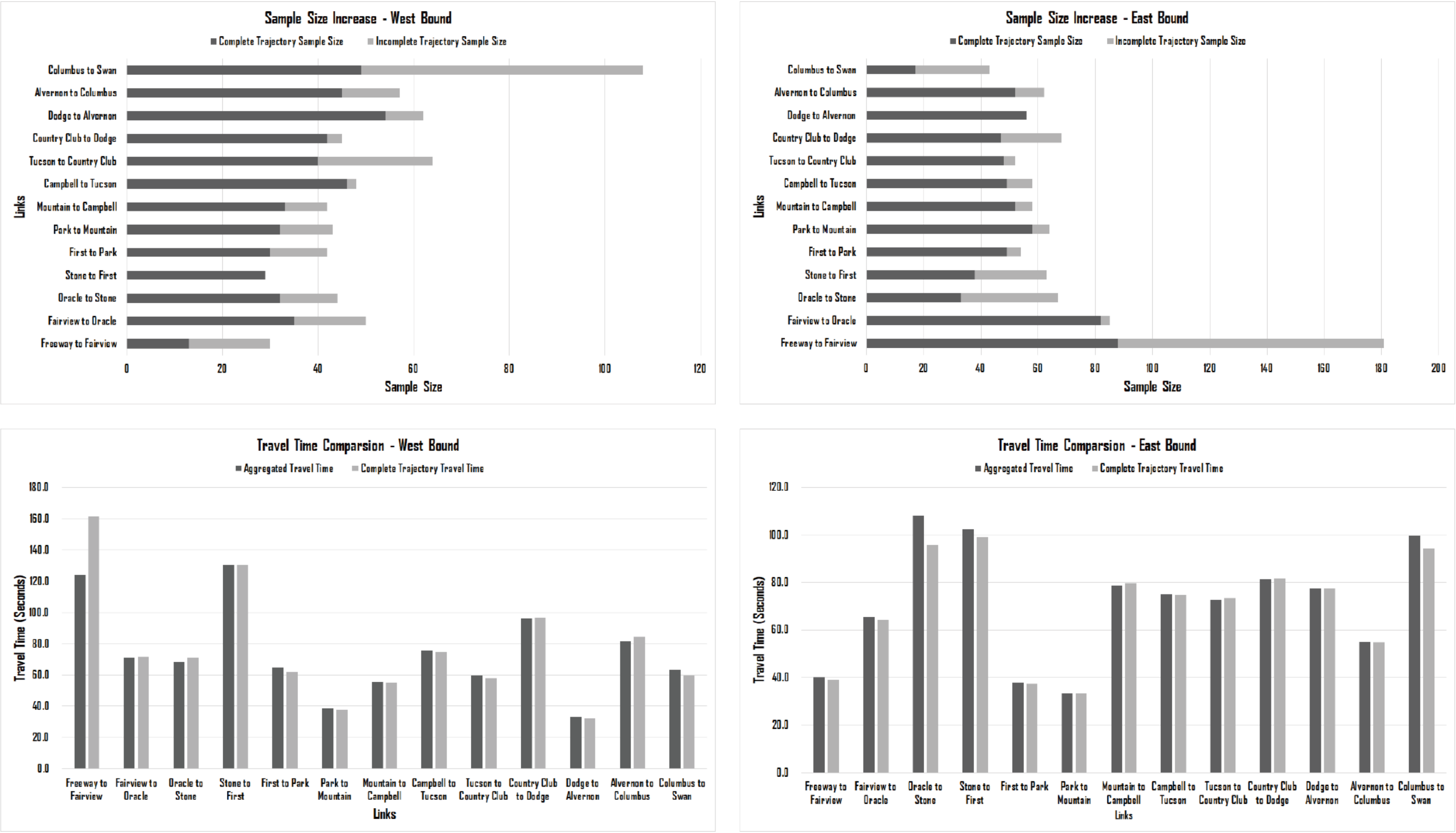


Figure 16 Travel Time Estimation Result, November 2015

Table 10 Link Travel Time Estimation Result, November 2015

Link Name	Direction	Complete Trajectory Sample Size	Incomplete Trajectory Sample Size	Sample Size Increase	Total Sample Size
Freeway to Fairview Ave	WB Through	13	17	131%	30
Fairview Ave to Oracle	WB Through	35	15	43%	50
Oracle Rd to Stone Ave	WB Through	32	12	38%	44
Stone Ave to First Ave	WB Through	29	0	0%	29
First Ave to Park	WB Through	30	12	40%	42
Park to Mountain	WB Through	32	11	34%	43
Mountain to Campbell	WB Through	33	9	27%	42
Campbell to Tucson	WB Through	46	2	4%	48
Tucson to Country Club	WB Through	40	24	60%	64
Country Club to Dodge	WB Through	42	3	7%	45
Dodge to Alvernon	WB Through	54	8	15%	62
Alvernon to Columbus	WB Through	45	12	27%	57
Columbus to Swan	WB Through	49	59	120%	108
Freeway to Fairview Ave	EB Through	88	93	106%	181
Fairview Ave to Oracle	EB Through	82	3	4%	85
Oracle Rd to Stone Ave	EB Through	33	34	103%	67
Stone Ave to First Ave	EB Through	38	25	66%	63
First Ave to Park	EB Through	49	5	10%	54
Park to Mountain	EB Through	58	6	10%	64
Mountain to Campbell	EB Through	52	6	12%	58
Campbell to Tucson	EB Through	49	9	18%	58
Tucson to Country Club	EB Through	48	4	8%	52
Country Club to Dodge	EB Through	47	21	45%	68
Dodge to Alvernon	EB Through	56	0	0%	56
Alvernon to Columbus	EB Through	52	10	19%	62
Columbus to Swan	EB Through	17	26	153%	43

5.3 Sensitivity Analysis

Sensitivity analysis was conducted to evaluate the performance of the k-NN module. Leave-one-out cross validation and the variable controlling method was used to measure the performance of the module in different scenarios. Five factors were analyzed and they fall into two categories: algorithm parameters and input data. The category of algorithm parameters includes the number of similar samples and the number of road segments.

Input data includes the length of incomplete trajectory, different type of incomplete trajectory, and time of day of incomplete trajectory. In each round of the LOOCV process, one complete trajectory that was not used before was cut off as an incomplete trajectory, which is the input of the algorithm. Other complete trajectories were used as the training set. The incomplete trajectory was compared with the training set to find the most similar trajectories. The ground truth is the link travel time that calculated by the complete trajectory before conversion and the algorithm output is the average link travel time of the k most similar complete trajectories.

Eastbound through movement of Grant Road from Fairview to Oracle in 2015 was selected as the study link for the sensitivity analysis. There were 955 link travel time samples – in other words, there were 955 complete trajectories. There were 470 samples in peak hour and 485 samples in non-peak hour. The peak hours in this study were defined as 7:30 AM to 9:30 AM and 4:00 PM to 6:00 PM on weekdays. The non-peak hours were defined as the rest of time except for the defined peak hours.

5.3.1 Parameters

There were two main parameters that were varied in the study: the number of similar samples (k) and the number of road segments (or the dimension of data, n). The number of similar samples is the number of the most similar complete trajectories and the average travel time of these complete trajectory samples is the output of the algorithm. As discussed before, the input of the algorithm was the time for a vehicle to pass each segment of the link. Thus, the number of road segments is the dimension of input data. High dimension data can show smaller changes of speed but may make it hard to find similar trajectories. A case study was designed to investigate the influence of these two parameters.

As mentioned before, it is not only algorithm parameters that may affect the performance of the algorithm, but also different input data that may influence the result. To eliminate the influence of different input data, in each round of LOOCV, a complete trajectory was converted into a 50% length, type 2 incomplete trajectory (i.e., a trajectory that enters the link after the upstream location and leaves before downstream location) as the input – and the ground truth was calculated link travel time of the complete trajectory before conversion. All the complete trajectories had converted into incomplete trajectories once and the LOOCV evaluated average performance of each round. The LOOCV result may not be the same as real-world performance since the incomplete trajectory input was converted from complete trajectory. However, it can still be a great references indicator for the real-world condition and can also show the impact of different parameters.

The experimental conditions of different parameters are shown below:

- **Scenario 1:** The link was divided into 4 road segments and the number of similar samples varied between 2 and 50.
- **Scenario 2:** The link was divided into 20 road segments and the number of similar samples varied between 2 and 50.
- **Scenario 3:** The link was divided into 50 road segments and the number of similar samples varied between 2 and 50.

Figure 17 and Figure 18 show the performance of k-NN algorithm using different parameter combinations. In the case study, the algorithm reached its best performance when the link was cut into 4 road segments and used 5 similar samples. It can be found that there is a negative relationship between the performance of the algorithm and the number of road segments. With the increase in the number of road segments, the performance of the algorithm got worse. However, decreasing the number of road segments significantly reduces the utilization rate of incomplete trajectory since the part of an incomplete trajectory that has not fully passed a link will be removed. In real-world applications, a small number of road segments is not suggested.

With the increase in the number of similar samples, the performance of the algorithm improved at first. When the number of similar samples reached a threshold, the algorithm performed worse with the increase of the number of similar samples. The result was similar that is found in many other k-NN applications. We suggest that an appropriate number of similar samples needs to be selected before real-world application of the algorithm.

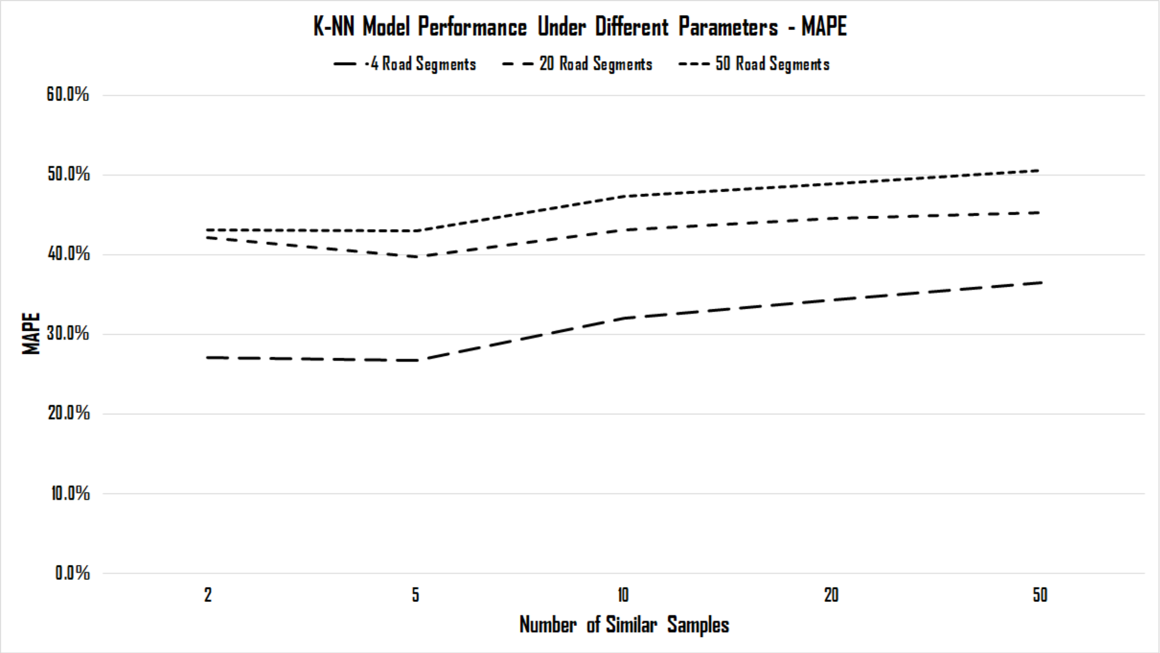


Figure 17 k-NN Performance under Different Parameters - MAPE

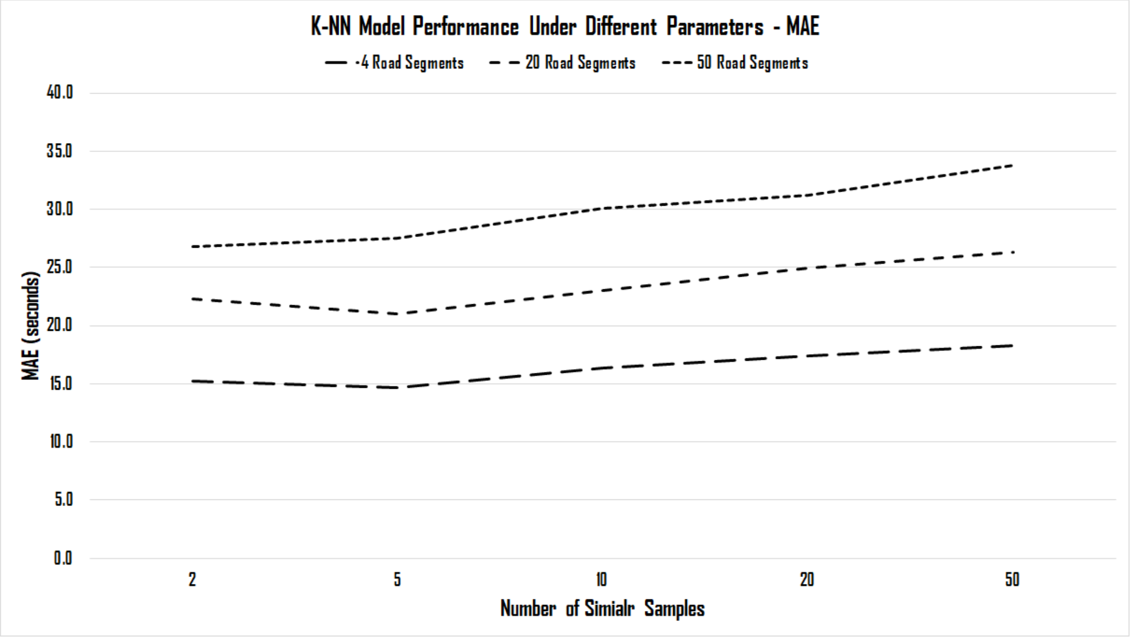


Figure 18 K-NN Performance Under Different Parameters - MAE

5.3.2 Input Data

There were three main characteristics of input data: the length of incomplete trajectory, types of incomplete trajectories, and time of day of incomplete trajectory. To eliminate the influence of algorithm parameters, the number of similar samples was selected as 5 and the number of divided link segments was selected as 20. Complete trajectories were cut-off so as to become incomplete trajectories and used as input data. The experimental conditions with respect to different input data are shown below:

- **Scenario 1 (incomplete trajectory length)**: Complete trajectories were converted into different lengths of incomplete trajectories from 5% to 90% of the link length. Complete trajectories were converted as type 2, which is the type of incomplete trajectories that enter after the upstream intersection and leave before the downstream intersection. Only complete trajectories that were in peak hours were used.

- **Scenario 2 (incomplete trajectory types)**: Complete trajectories were converted into different types of incomplete trajectories. Complete trajectories were converted into 50% of the link length. Only complete trajectories that were in peak hours were used.

- **Scenario 3 (time of day)**: Complete trajectories were converted into 80% of the link length as incomplete trajectory type 2. Complete trajectories that occurred at different times of day were compared. Note that time of day of the complete trajectories only affects the number of cross validation rounds, but the training set was still composed of complete trajectories from all times of day.

The result of scenario 1 is shown in Table 11, Figure 19 and Figure 20. It can be found that incomplete trajectory length has an impact on the performance of k-NN module. The performance of the algorithm continuously drops with the shortening of incomplete trajectory length. Both MAPE and MAE reach to their minimum values when incomplete trajectory length is 90% of the link length. The minimum and maximum value of MAPE is 6.8% and 33.3% respectively and the minimum and maximum value of MAE is 5.7s and 22.1s respectively. MAPE and MAE were negatively associated with incomplete trajectory length. Since long incomplete trajectories are similar to complete trajectories, it is easier for long incomplete trajectories to find complete trajectories that experienced the same traffic.

Table 11 LOOCV Result of Different Incomplete Trajectory Length

Incomplete Trajectory Length / Link Length	MAPE	MAE(seconds)
5%	33.3%	22.1
10%	32.0%	20.9
15%	32.2%	21.3
20%	30.8%	20.0
25%	30.1%	19.5
30%	28.7%	18.9
35%	28.0%	18.6
40%	27.9%	18.7
45%	28.9%	18.9
50%	28.0%	18.3
55%	26.3%	17.5
60%	23.9%	16.6
65%	23.1%	16.1
70%	20.5%	14.5
75%	20.5%	14.4
80%	14.7%	11.1
85%	13.7%	10.6
90%	6.8%	5.7

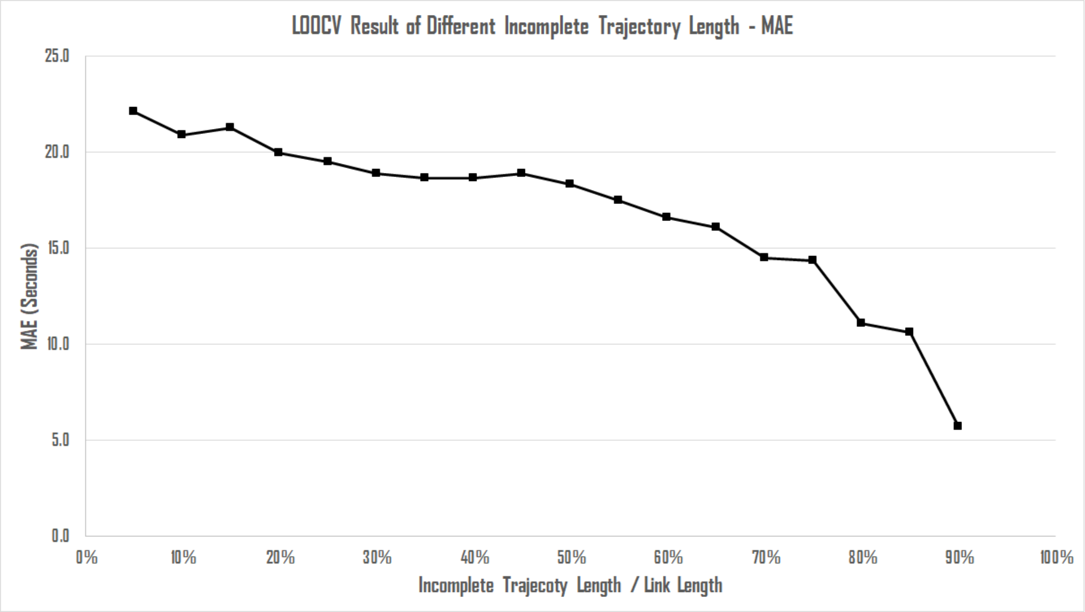


Figure 19 LOOCV Result of Different Trajectory Length – MAE

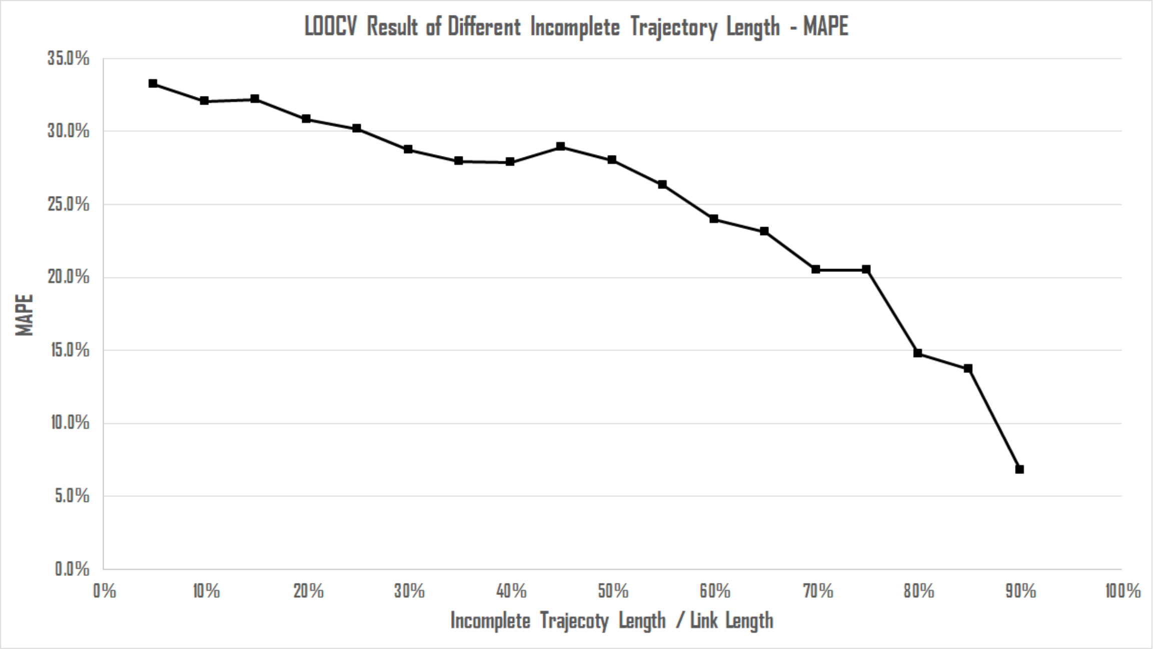


Figure 20 LOOCV Result of Different Trajectory Length – MAPE

Table 12 shows the result of the algorithm under different trajectory types. Incomplete trajectory type 1 performed worst with an MAE of 18.3s and an MAPE of 39.1%. Incomplete trajectory type 2 and type 3 performed almost the same with an MAE around 7s and an MAPE around 20.7%. Incomplete trajectory type 2 and type 3 performed better than type 1 on average. It can be inferred from the result that incomplete trajectory that contains queue information performs better.

Table 12 LOOCV Result of Different Incomplete Trajectory Types

Incomplete Trajectory Types	Type 1	Type 2	Type 3
MAE/Seconds	18.3	6.5	7.5
MAPE	39.10%	20.50%	20.90%

Table 12 shows the comparison of the result by different time of day. Since the attribute of time of day cannot be simulated, input data was classified into several categories by time of day. Morning peak is defined as 7:30 AM to 9:30 AM on weekdays and evening peak is defined as 4:00 PM to 6:00 PM on weekdays. Peak hour is defined as the combination of morning peak and evening peak. Non-peak is all the time except peak hours. The definition of peak hour and non-peak came from the real-world traffic conditions. Note that, even though the input data is classified as being at peak, the training set still covers all time periods because the assumption is that a trajectory can reflect all traffic conditions.

The result shows that the algorithm performed better during peak hours. The average of MAPE during peak hour is around 14% and the average of MAE during peak hour is about 10s. The algorithm performed worse during non-peak hour with a MAPE of 26.4% and a MAE of 12.4s. The reason why the algorithm performed better during peak hour may be because vehicles have similar trajectories during peak hour since traffic is more congested. Vehicles trajectories during non-peak hour may depend more on drivers’ behavior. Similar trajectories during non-peak hour may be due to similar driving behavior rather than traffic condition.

Table 13 LOOCV Result of Different Time of Day

Time of Day	**Morning Peak**	**Evening Peak**	**Peak**	**Non-Peak**	**All**
Sample Size	372	113	485	470	955
MAPE	13.6%	15.6%	14.0%	26.4%	20.1%
MAE/seconds	10.3	9.49	10.1	12.4	11.2

6. CONCLUSIONS AND FUTURE RESEARCH

6.1 Conclusions

Link travel time plays a significant role in traffic planning, traffic management and Advanced Traveler Information Systems (ATIS). Previously, travel time was mainly collected from fixed sensors or small-scale surveys (probe vehicle, floating cars, etc.). Large scale data collection is expensive due to the cost of devices and labor. A public probe vehicle dataset is the probe vehicle dataset that is collected from public people or public transport. The appearance of a public probe vehicle dataset can support travel time collection at a large temporal and spatial scale but at a relatively low cost.

Traditionally, link travel time is the aggregation of travel time by different movements. A recent study proved that link travel time of different movements is significantly different from their aggregation. Movement-based link travel time has not been popular previously mainly because most fixed sensors cannot identify the movement of traffic. Further, there is not a complete framework for large scale movement-based travel time estimation using mobile sensors. This study proposed a detailed framework to estimate movement-based link travel time using high sampling rate of a public probe vehicle dataset. The result of the case study shows that the framework can successfully calculate movement-based link travel time. The framework performs well on most links, but not so well with respect to movement on the boundary of the road network since the movement of a vehicle cannot be identified.

The quality of a probe vehicle dataset is mainly decided by its sampling rate and its penetration rate. Sampling rate is the collecting frequency of GPS points; penetration rate is the ratio of probe vehicles out of all vehicles on the research corridor. When sampling rate is low, link travel time is hard to estimate because adjacent GPS points may be located on different links. When penetration rate is low, link travel time samples may not be able to represent the entire population. Some research exists concerning travel time estimation using probe vehicle data with a low sampling rate, but very few concerning travel time estimations when penetration rate is low. Our study proposed a method to calculate travel time samples using incomplete trajectory, which had not been utilized before. Incomplete trajectory is the trajectory that does not pass fully through the study link, so link travel time cannot be directly calculated from incomplete trajectory.

Our study proposed a k-NN based travel time calculation method using incomplete trajectories to generate additional travel time samples. Incomplete trajectories were compared with historical complete trajectories and link travel times of incomplete trajectories were represented by these similar complete trajectories. The result of our study shows that the method can significantly increase link travel time samples. However, there are still some limitations. One limitation is that the algorithm requires a large historical dataset. Thus, through movements were more suitable to implement k-NN algorithm than turning movements since the historical dataset of through movements is larger than for turning movements. Another limitation is that the accuracy of the algorithm cannot be estimated in real world since the ground truth of link travel time for the real world incomplete trajectory is unknown.

Our study also evaluated the performance of the k-NN algorithm in different scenarios. In sensitivity analysis, incomplete trajectory that was converted from a complete trajectory was used as input of the algorithm. The sensitivity analysis of the k-

NN algorithm shows that the algorithm performed differently under different parameters and input data. In real world application, optimal parameters need to be selected for an accurate result. It is suggested that these optimal parameters should be selected using a historical dataset before real-world application. In the case study reported here, the research into the key parameters and input data concluded the following:

- Both the number of similar samples and the number of road segments influence the accuracy of the algorithm. Although a small number of road segments can improve the performance of the algorithm in the case study, in a real-world application, a small number of road segments would also reduce the number of incomplete trajectories that can be used for the algorithm.

- The length of incomplete trajectory has a positive correlation with performance of the algorithm. A long incomplete trajectory is similar to a complete trajectory, so that length makes it easier to find similar complete trajectories.

- The performance of the algorithm is correlated with the time of day. Incomplete trajectory utilized during peak hours performed better than during non-peak hours.

- Incomplete trajectories that contain queue information performed better than incomplete trajectories that did not contain queue information. Link travel time is the summary of free flow travel time and delay, in which delay is the key element that decides link travel time. Queue information is highly correlated with delay so the algorithm performed better when the input data reflected queue information.

6.2 Future Work

Future studies could evaluate the performance of the k-NN algorithm on turning movements. Due to the limitation of there being only a small historical dataset concerning turning movement, the experiment with respect to the turning movement is not applicable until now. Study on turning movements can further evaluate the application of the algorithm and provide more samples for link travel time study.

Future studies could also focus on performance evaluation of the framework under a connected or autonomous vehicle environment. Since the confidence level of the estimated travel time samples can only be measured when the whole set is known, the real-world performance of the framework can only be verified under connected vehicle environment.

APPENDIX A: DISTRIBUTION OF DIRECT CALCULATED LINK TRAVEL TIME

Table 14 Through Movement Travel Time Distribution, 2015, Morning Peak (7:30 AM - 9:30AM)

Link Name	Direction	2015/1	2015/2	2015/3	2015/4	2015/5	2015/6	2015/7	2015/8	2015/9	2015/10	2015/11	2015/12
Freeway to Fairview Ave	WB Through	0	0	0	0	0	0	0	0	0	0	2	0
Fairview Ave to Oracle	WB Through	0	0	5	12	11	11	11	6	2	7	7	2
Oracle Rd to Stone Ave	WB Through	0	0	6	13	9	13	9	6	3	8	5	1
Stone Ave to First Ave	WB Through	1	0	6	13	7	12	9	4	2	3	5	1
First Ave to Park	WB Through	0	0	8	17	8	13	10	7	5	3	5	2
Park to Mountain	WB Through	0	0	9	25	20	14	12	11	7	4	4	5
Mountain to Campbell	WB Through	0	0	6	25	16	17	13	9	5	3	4	6
Campbell to Tucson	WB Through	0	0	7	19	8	11	16	10	7	4	6	6
Tucson to Country Club	WB Through	0	1	9	28	10	9	10	9	7	0	5	5
Country Club to Dodge	WB Through	0	0	8	30	9	10	8	10	9	3	6	6
Dodge to Alvernon	WB Through	0	0	8	31	9	11	4	13	9	8	10	5
Alvernon to Columbus	WB Through	0	0	5	31	6	10	4	7	6	7	9	5
Columbus to Swan	WB Through	0	0	8	35	11	10	3	7	8	9	9	6
Freeway to Fairview Ave	EB Through	2	2	8	41	29	25	33	38	44	36	38	40
Fairview Ave to Oracle	EB Through	2	2	7	37	30	25	35	32	31	33	31	40
Oracle Rd to Stone Ave	EB Through	2	0	0	18	23	23	33	17	18	23	5	20
Stone Ave to First Ave	EB Through	1	0	0	10	9	5	12	3	6	11	9	16
First Ave to Park	EB Through	1	0	0	8	9	5	13	8	7	15	11	18
Park to Mountain	EB Through	0	0	0	6	3	1	8	8	4	15	9	15
Mountain to Campbell	EB Through	0	0	0	3	3	2	7	5	4	14	9	11
Campbell to Tucson	EB Through	0	1	0	3	5	2	8	5	3	14	9	11
Tucson to Country Club	EB Through	0	0	1	3	5	1	8	6	3	15	10	11
Country Club to Dodge	EB Through	0	0	1	3	5	0	9	8	3	13	11	10
Dodge to Alvernon	EB Through	0	0	1	4	5	3	10	10	4	14	10	9
Alvernon to Columbus	EB Through	0	0	1	4	5	2	11	12	4	15	11	10
Columbus to Swan	EB Through	0	0	0	0	0	0	2	3	0	1	3	0

Table 15 Through Movement Travel Time Distribution, 2015, Evening Peak (4:00 PM - 6:00 PM)

Link Name	Direction	2015/1	2015/2	2015/3	2015/4	2015/5	2015/6	2015/7	2015/8	2015/9	2015/10	2015/11	2015/12
Freeway to Fairview Ave	WB Through	0	0	0	0	0	0	0	0	0	0	5	0
Fairview Ave to Oracle	WB Through	1	0	4	9	10	2	3	5	3	10	9	6
Oracle Rd to Stone Ave	WB Through	0	0	4	8	10	2	4	5	3	9	9	6
Stone Ave to First Ave	WB Through	0	0	4	2	3	1	1	4	0	2	6	4
First Ave to Park	WB Through	0	0	2	2	3	1	4	3	0	0	4	4
Park to Mountain	WB Through	0	0	6	3	6	7	4	2	1	2	3	9
Mountain to Campbell	WB Through	0	0	2	1	5	2	4	2	1	0	6	6
Campbell to Tucson	WB Through	0	0	3	4	4	0	2	2	1	0	7	4
Tucson to Country Club	WB Through	0	0	3	4	2	0	2	2	2	3	7	4
Country Club to Dodge	WB Through	0	0	3	4	2	0	3	4	4	5	11	6
Dodge to Alvernon	WB Through	0	0	2	11	6	5	2	7	17	6	10	6
Alvernon to Columbus	WB Through	0	0	2	10	3	1	4	6	19	5	11	6
Columbus to Swan	WB Through	0	0	3	12	4	0	4	12	20	5	12	8
Freeway to Fairview Ave	EB Through	6	0	9	19	15	7	9	14	5	7	11	4
Fairview Ave to Oracle	EB Through	6	0	9	19	15	7	9	14	5	6	11	3
Oracle Rd to Stone Ave	EB Through	6	0	4	6	9	3	4	3	2	2	8	1
Stone Ave to First Ave	EB Through	5	0	4	5	7	4	3	3	3	2	8	0
First Ave to Park	EB Through	1	0	4	8	10	5	4	7	4	4	8	1
Park to Mountain	EB Through	0	0	2	8	11	5	3	8	4	4	9	6
Mountain to Campbell	EB Through	0	0	1	6	8	3	4	9	3	2	8	5
Campbell to Tucson	EB Through	0	0	4	10	6	4	4	9	4	2	6	4
Tucson to Country Club	EB Through	1	0	5	11	5	2	5	6	4	3	7	5
Country Club to Dodge	EB Through	1	0	6	9	3	2	6	5	3	1	6	5
Dodge to Alvernon	EB Through	0	0	6	10	9	7	5	5	3	2	10	5
Alvernon to Columbus	EB Through	0	0	3	6	9	4	5	4	4	2	12	5
Columbus to Swan	EB Through	0	0	0	0	0	0	1	0	2	0	5	0

APPENDIX B: PSEUDO CODE

B.1 Direct Travel Time Measurement

```
For each trajectory that pass the study link {
    // Eliminate map matching error
    Calculate all the corridors that trajectory has entered;
    For all corridors that trajectory has entered {
        Find the timestamp when vehicle enter the corridor and leave the corridor;
        If the time difference between vehicle enter the link and leave the link < a threshold (used 3 seconds) {
            Match GPS points in this time period to the last corridor and the next corridor temporal evenly;
        }
    }

    // Eliminate error generate from long GPS reporting time
    Find the first GPS point when vehicle enter the study link (point F) and last GPS point before leaving the study link (point L);
    If the distance of point F and upstream point of the link > a threshold (used 10 feet) {
        continue;
    }
    If the distance of point L and downstream point of the link > a threshold (used 10 feet) {
        continue;
    }

    // Calculate link travel time
    Link_travel_time = time difference between point F and point L;

    // Determine the movement of calculated link travel time
    Find the corridor before the vehicle enter the study corridor (upstream corridor);
    Find the corridor after the vehicle leave the study corridor (downstream corridor);
    For corridor movement table {
        Find if the sequence of (upstream corridor, study corridor, downstream corridor) appeared in the movement table;
        If Yes {
            Get a link travel time sample of that movement
        }
        else {
            The trajectory is an incomplete trajectory
        }
    }
}
```

B.2 Travel Time Estimation using Incomplete Trajectory to Simulate Complete Trajectory

```
// Parameters
k = the number of similar samples;
n = the number of road segments;
spline_num = the number of smoothed spline points

// Generate training set
For each trajectory that has fully pass the study link {
    Fit a smoothing spline for the trajectory using spline parameter;
```

```
        Divided smoothed spline into n segments according to the distance of GPS points and upstream point of the link;
        Calculate travel time for each segment;
        Generate a n dimension array of segment travel time;
}

// Deal with incomplete trajectory
Fit a smoothing spline for the trajectory using spline parameter;
Find the segments that incomplete trajectory that has fully passed, which is set M here and dimension is m;
Calculate travel time for each segment that incomplete trajectory has fully passed;
Generate a m dimension array of segment travel time that incomplete trajectory has fully passed;

// Calculate Distance between incomplete trajectory and training set
For each one in the training set {
        Extract the corresponding travel time of the same segment that incomplete trajectory has fully passed and generate a m dimension array;
        Calculate the Euclid distance between m dimension array of incomplete trajectory and complete trajectory;
}

// Calculate simulated travel time of incomplete trajectory
Find k complete trajectories that has least Euclid distance to incomplete trajectory;
Simulated travel time of incomplete trajectory = the average value of travel time of k complete trajectories
```

REFERENCES

Abdulhai, B., Porwal, H., & Recker, W. (2002). Short-term traffic flow prediction using neuro-genetic algorithms. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, *7*(November), 3–41. https://doi.org/10.1080/10248070190048664

Argote-Cabañero, J., Christofa, E., & Skabardonis, A. (2015). Connected vehicle penetration rate for estimation of arterial measures of effectiveness. *Transportation Research Part C: Emerging Technologies*, *60*, 298–312. https://doi.org/10.1016/j.trc.2015.08.013

Boyce, D.E., Kirson, A.M., Schofer, J.L. (1994). Advance: The Illinois dynamic navigation and route guidance demonstration program. In: Catling, I. (Ed.), Advanced Technology for Road Transport: IVHS and ATT. Artech House, London, pp. 247–270, Chapter 11.

Bucknell, C., & Herrera, J. C. (2014). A trade-off analysis between penetration rate and sampling frequency of mobile sensors in traffic state estimation. *Transportation Research Part C: Emerging Technologies*, *46*, 132–150. https://doi.org/10.1016/j.trc.2014.05.007

Byon, Y.J., Shalaby, A., Abdulhai, B., 2006. GISTT: GPS-GIS integrated system for travel time surveys. Transportation Research Board 2006 annual meeting CD-ROM paper number 1555.

Cao, P., Miwa, T., & Morikawa, T. (2014). Modeling Distribution of Travel Time in Signalized Road Section Using Truncated Distribution. *Procedia - Social and Behavioral Sciences*, *138(0)*, 137–147. https://doi.org/10.1016/j.sbspro.2014.07.189

Cheu, R. L., Xie, C., & Lee, D. H. (2002). Probe vehicle population and sample size for arterial speed estimation. *Computer-Aided Civil and Infrastructure Engineering*, *17*(1), 53–60. https://doi.org/10.1111/1467-8667.00252

Coifman, B., & Cassidy, M. (2002). Vehicle reidentification and travel time measurement on congested freeways. *Transportation Research Part A: Policy and Practice*, *36*(10), 899–917. https://doi.org/10.1016/S0965-8564(01)00046-5

Feng, Y., Hourdos, J., & Davis, G. A. (2014). Probe vehicle based real-time traffic monitoring on urban roadways. *Transportation Research Part C: Emerging Technologies*, *40*, 160–178. https://doi.org/10.1016/j.trc.2014.01.010

Handley, S., P. Langley, and F. A. Rauscher. Learning to Predict the Duration of an Automobile Trip. *Proc., 4th International Conference on Knowledge Discovery and Data Mining*, New York, 1998, pp. 219–223.

Haghani, A., Hamedi, M., Sadabadi, K., Young, S., & Tarnoff, P. (2010). Data Collection of Freeway Travel Time Ground Truth with Bluetooth Sensors. *Transportation Research Record: Journal of the Transportation Research Board*, *2160*(2160), 60-68. https://doi.org/10.3141/2160-07

Hellinga, B. R., & Fu, L. (2002). Reducing bias in probe-based arterial link travel time estimates. *Transportation Research Part C: Emerging Technologies, 10(4)*, 257–273. https://doi.org/10.1016/S0968-090X(02)00003-7

Jenelius, E., & Koutsopoulos, H. N. (2013). Travel time estimation for urban road networks using low frequency probe vehicle data. *Transportation Research Part B: Methodological*, *53*, 64–81. https://doi.org/10.1016/j.trb.2013.03.008

Kramer, O. (2013). *Dimensionality Reduction with Unsupervised Nearest Neighbors*. Intelligent Systems Reference Library 51. (New York; Berlin: Springer).

Lee, S., & Fambro, D. B. (1999). Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transportation Research Record: Journal of the Transportation Research Board*, *1678*(99), 179-188. https://doi.org/10.3141/1678-22

Li, R., & Rose, G. (2011). Incorporating uncertainty into short-term travel time predictions. *Transportation Research Part C: Emerging Technologies*, *19*(6), 1006-1018. https://doi.org/10.1016/j.trc.2011.05.014

Li, Y., McDonald, M., 2002. Link travel time estimation using single GPS equipped probe vehicle. In*: The IEEE 5th International Conference on Intelligent Transportation Systems*, Singapore.

Lin, W., (2001). A Gaussian maximum likelihood formulation for short-term forecasting of traffic flow. *Intelligent Transportation Systems, 2001 IEEE Proceedings*, 150-155. https://doi.org/10.1109/ITSC.2001.948646

Liu, H. X., & Ma, W. A. (2009). virtual vehicle probe model for time-dependent travel time estimation on signalized arterials. *Transportation Research Part C: Emerging Technologies*, 17, 11-26.

Pima Association of Governments, 2014. PAG Traffic Counts. Available at:http://www.pagregion.com/Default.aspx?tabid=909[Accessed May 11, 2016].

Mehran, B., Kuwahara, M., & Naznin, F. (2012). Implementing kinematic wave theory to reconstruct vehicle trajectories from fixed and probe sensor data. *Transportation Research Part C: Emerging Technologies*, *20*(1), 144–163. https://doi.org/10.1016/j.trc.2011.05.006

Moorthy, C. K., & Ratcliffe, B. G. (1988). Short term traffic forecasting using time series methods. *Transportation Planning and Technology*, *12*(1), 45–56. https://doi.org/10.1080/03081068808717359

Nantes, A., Ngoduy, D., Bhaskar, A., Miska, M., & Chung, E. (2016). Real-time traffic state estimation in urban corridors from heterogeneous data. *Transportation Research Part C: Emerging Technologies*, *66*, 99–118. https://doi.org/10.1016/j.trc.2015.07.005

Park, D., & Rilett, L. R. (1998). Forecasting multiple-period freeway link travel times using modular neural networks. *Journal of the Transportation Research Board*, (98), 163–170. Retrieved from http://trb.metapress.com/index/n637779538j97k60.pdf

Park, S., Saeedi, A., Kim, D. S., & Porter, J. D. (2016). Measuring Intersection Performance from Bluetooth-Based Data Utilized for Travel Time Data Collection. *Journal of Transportation Engineering*, *142*(4016014), 1–9. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000836.

Patire, A. D., Wright, M., Prodhomme, B., & Bayen, A. M. (2015). How much GPS data do we need? *Transportation Research Part C: Emerging Technologies*, *58*, 325-342. https://doi.org/10.1016/j.trc.2015.02.011

Quantum GIS Development Team (2017). Quantum GIS Geographic Information System. *Open Source Geospatial Foundation Project*. http://qgis.osgeo.org

Robinson, S., & Polak, J. (2005). Modeling urban link travel time with inductive loop detector data by using the k-NN method. *Transportation Research Record: Transportation Research Board*, *1935*(1935), 47–56. https://doi.org/10.3141/1935-06

Seo, T., & Kusakabe, T. (2015). Probe vehicle-based traffic flow estimation method without fundamental diagram. *Transportation Research Procedia*, *9*, 149–163. https://doi.org/10.1016/j.trpro.2015.07.009

Sherali, H. D., Desai, J., & Rakha, H. (2006). A discrete optimization approach for locating Automatic Vehicle Identification readers for the provision of roadway travel times. *Transportation Research Part B: Methodological*, *40*(10), 857–871. https://doi.org/10.1016/j.trb.2005.11.003

Srinivasan, K., & Jovanis, P. (1996). Determination of number of probe vehicles required for reliable travel time measurement in urban network. *Transportation Research Record*, *1537*, 15–22. https://doi.org/10.3141/1537-03

Wan, N., Vahidi, A., & Luckow, A. (2016). Reconstructing maximum likelihood trajectory of probe vehicles between sparse updates. *Transportation Research Part C: Emerging Technologies*, *65*, 16–30. https://doi.org/10.1016/j.trc.2016.01.010

Wang, Y., Malinovskiy, Y., Wu, Y., Lee, U. (2011). Error Modeling and Analysis for Travel Time Data Obtained from Bluetooth MAC Address, (1), 82.

Wu, C., Wei, C., Su, D., Chang, M., & Ho, J. (2004). Travel time prediction with support vector regression. *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*, *2*, 1438–1442. https://doi.org/10.1109/ITSC.2003.1252721

Yeon, J., Elefteriadou, L., & Lawphongpanich, S. (2008). Travel time estimation on a freeway using Discrete Time Markov Chains. *Transportation Research Part B: Methodological*, *42*(4), 325–338. https://doi.org/10.1016/j.trb.2007.08.005

Zhan, X., Hasan, S., Ukkusuri, S. V., & Kamga, C. (2013). Urban link travel time estimation using large-scale taxi data with partial information. *Transportation Research Part C: Emerging Technologies*, *33*, 37–49. https://doi.org/10.1016/j.trc.2013.04.001

Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C*, *58*, 308–324. https://doi.org/10.1016/j.trc.2015.02.019

Zheng, F., & Van Zuylen, H. (2013). Urban link travel time estimation based on sparse probe vehicle data. *Transportation Research Part C: Emerging Technologies*, *31*(111), 145–157. https://doi.org/10.1016/j.trc.2012.04.007

Zheng, Y., & Zhou, X. (Eds.). (2011). Computing with spatial trajectories. *Springer Science & Business Media.*

Zhou, X., Yang, Z., Zhang, W., Tian, X. and Bing, Q., 2016. Urban Link Travel Time Estimation Based on Low Frequency Probe Vehicle Data. *Discrete Dynamics in Nature and Society, 2016.*

Zhu, T., Kong, X., & Lv, W. (2009). Large-scale travel time prediction for urban arterial roads based on Kalman filter. *Proceedings - 2009 International Conference on Computational Intelligence and Software Engineering, CiSE 2009*, 1–5. https://doi.org/10.1109/CISE.2009.5365441