Transit Origin Destination Estimation using Automated Data

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Pramesh Kumar

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Alireza Khani

December, 2019

Acknowledgements

There are many people that I would like to acknowledge for their contribution to my time in graduate school. First of all, I would like to express gratitude to my advisor Dr. Alireza Khani. His immense knowledge of transportation, academic excellence, and fabulous mentorship has led me to finish my thesis. I would like to thank him for motivating me to work in the area of Transportation Networks and for countless discussions on various interesting problems in this area. The work presented in this thesis would not have been possible without him. Next, I thank Prof. Gary A. Davis for his valuable contribution in this thesis. His expertise and mentorship strengthened the methodologies presented in Chapter 5. I also thank Prof. William Cooper for serving in my thesis committee.

I am grateful to have found some amazing friends during my graduate school. I would like to thank current and former transit lab members - Jack, Jackie, Benj, Eugene, Kai, Yufeng, Ben, and Alex who made countless work in the lab far more enjoyable. I would also like to thank Tarun, Chris and Rongsheng for their wonderful company.

The work presented in this thesis would also not have been possible without the support of Metro Transit. I would like to acknowledge John Levin, Eric Lind, and other Metro Transit members for sharing the data and helpful feedback on the research. The work presented in this thesis was funded in part by National Science Foundation Award CMMI-1637548.

Dedication

To my family for their love, endless support, encouragement, & sacrifices.

Abstract

Development of an origin-destination (OD) demand matrix is crucial for transit planning. The development process is facilitated by transit automated data, making it possible to mine boarding and alighting patterns on an individual basis. This thesis presents novel methods for estimating transit OD matrix using automatically collected data. Depending on the type of transit automated data, there are two methods presented. A novel trip chaining method which uses Automatic Fare Collection (AFC), Automatic Vehicle Location (AVL), and General Transit Feed Specification (GTFS) data is proposed to infer the most likely trajectory of individual transit passenger. The method relaxes the assumptions on various parameters used in the existing trip chaining algorithms such as transfer walking distance threshold, buffer distance for selecting the boarding location, the time window for selecting the vehicle trip, etc. The thesis also proposes a method for estimating the transit route origin-destination (OD) matrix utilizing Automatic Passenger Count (APC) data. It uses l_0 norm regularizer, which leverages the sparsity present in the actual OD matrix. The technique is popularly known as compressed sensing (CS). The applications of both methods using automated data from Twin Cities, MN are also presented. The results show improved accuracy and more inference rate in calculating the OD matrix using trip chaining. Similarly, compressed sensing was found to work impressively well in evaluating transit route OD matrix within small errors.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Public transport agencies have historically planned their service with limited knowledge of their customers' travel behavior. For example, they used farebox data to determine ridership of a transit route. To evaluate the passenger-centric information, they have relied on on-board surveys to collect data about passengers' boarding and alighting location, and the purpose of travel. As the survey data is limited, it is expanded to the whole population using expansion factors. There are also various limitations associated with these surveys, such as cost, small sample size, bias, and other general reporting errors [1]. Conversely, emerging automated data collection systems (ADCS) - namely Automated Fare Collection (AFC) system, Automated Passenger Count (APC) system, and Automated Vehicle Location (AVL) system - which are designed for administrative purposes such as revenue management, provide a rich source of information about passengers travel pattern on an individual basis. This data is useful not only for improving day-to-day transit operations but also for long-term strategic planning of transit network [2].

This thesis builds upon recent work on the synthesis of large-scale automated transit data with optimization and statistical techniques to understand the passenger travel pattern in a transit network. Using Twin Cities' automated transit data as an example, origin-destination (OD) matrices are estimated at different levels. Specifically, AFC and AVL data are used to estimate a network-wide stop-level OD matrix, and APC data is used to evaluate transit route-level OD matrix.

The remainder of this chapter describes the motivation for this research, presents its specific objectives, and finally outlines the structure of this thesis.

1.1 Motivation

This thesis focuses on one of the important input for analyzing a public transit system, which is the flow of passengers between different stations/stops known as an origin-destination (OD) matrix. OD estimation using automated smart card (or AFC) data has attracted attention of many researchers over the last decade [3–15]. AFC system records the fare related information when a passenger pays for a trip using a smart card. This includes a serial ID assigned to the card, date and time of transaction, route information, and coordinates.

The OD estimation requires a sequence of trips made by the passenger throughout the day recorded using AFC system. However, the information available with this data is limited and the full sequence of trips is usually not available. This is because of the type of the fare collection system (open or closed) employed by a transit agency. In closed transit systems [16], origin and destination is known for the trips as passengers tap their card both when boarding as well as when alighting, whereas in open transit systems [3–6,8–11,13,14], the boarding of passengers is usually known, and the alighting is unknown as passengers tap their card only when boarding a transit vehicle. The algorithm which infers missing boarding/alighting location in AFC data is known as trip chaining. It uses a sequence of tags (smart card transactions) to make a chain of trips made by the passenger by supplementing information from other data sources. The trip chaining algorithms developed so far use assumptions on various parameters, e.g. buffer radius to find the closest stop to the boarding location, walking distance threshold after alighting to board the next route, time threshold to distinguish between boarding and transfer, etc. These parameters can vary among different transit systems and can affect the trip chaining results and therefore the origin-destination matrix. The current research tries to relax the assumptions related to these parameters by proposing a robust trip chaining algorithm. Specific problems and their solutions related to current trip

chaining algorithms are discussed in §4.1.

The success of OD estimation using AFC data depends on the quality of data, the percentage of passengers using a smart card, and assumptions involved in the trip-chaining algorithm. Moreover, due to strict rules in the trip chaining algorithm, the inference rate may not be high. On the other hand, the APC system collect information about the number of passengers boarding and alighting at each transit stop. These boarding and alighting counts can be used to evaluate an OD matrix. However, the problem is hard to solve exactly as it requires solving an underdetermined system of equations, in which case the number of unknowns to solve is far more than the number of equations available. Usually, multiple solutions are possible for this problem, which satisfy the given system of equations. In Chapter 5, we propose a method to evaluate the transit route OD matrix using APC data. The problem is to estimate the flow of passengers between stops for a single transit trip. The route matrix problem has a special structure that provides an extra piece of information to reduce the ill-posedness of the system of equations involved. The estimation requires the selection of the correct estimate out of the multiple solutions. We use an estimation method that encourages the sparse OD matrix using l_0 norm regularizer. This helps in mitigating the ill-posedness of the system and offers interpretability [17] as there is only a subset of the origin-destination pairs which carries flow in an actual OD matrix. The method is popularly known as compressed sensing [18].

1.2 Objectives

This thesis proposes new methods which promise the estimation of transit OD matrix flexibly and efficiently using large scale automated data from a public transport agency, in a way that can be performed on a continuous basis. More specifically, this thesis seeks to fulfil the following objectives:

- *Infer boarding and alighting location of regular route transactions in AFC data*: AFC transactions on regular routes (passengers pay while boarding) lack alighting location of passengers. The objective is to develop a trip chaining algorithm to evaluate the missing alighting and transfer locations.

- *Infer boarding and alighting location of pay-exit route transactions in AFC data*: Some transit systems are more complex than others because of the provision of pay-exit routes. The pay exit buses are generally outbound trips from central areas such as Downtown or University campus to suburban areas. In that case, passengers tap their card when exiting the bus. This provides us with alighting information rather than boarding locations. This objective tries to develop a trip chaining algorithm which can also evaluate boarding and alighting for pay-exit routes.

- *Evaluate transit route OD matrix using APC data*: The objective is to develop an efficient optimization program to estimate transit route OD matrix using APC data which leverages special structure of resulting OD matrix.

1.3 Thesis Organization

The methods developed in this thesis can be grouped into two categories: OD estimation using AFC data and OD estimation using APC data. Since each category has a distinct methodological background, each is presented in its own chapter. Following literature review in Chapter 2, an overview of transit automated data is given in Chapter 3. Then proposed methods in each category are described in Chapter 4 and 5 and their technical implementation is presented in Chapter 6. The demonstration of applications is presented in Chapter 7 which is finally followed by conclusions and recommendations in Chapter 8.

Chapter 2

Literature Review

This chapter describes previous work related to the research presented in this thesis which is grouped into two categories:

2.1 Origin Destination inference using AFC data

As most of the fare collection systems record passengers' boarding information only, alighting information must be inferred using the sequence of taps (or tags) made by the passenger throughout the day. Thus, a significant amount of research has been done to develop algorithms to determine the alighting location [19]. Navick and Furth used location-stamped fare box data of Los Angeles area bus routes to determine alighting location using an assumption that boarding pattern of the current trip and alighting pattern of the opposite trip are symmetric for the entire day which means passengers board the bus again from the same stop where they alighted during the previous trip [20]. Building on that assumption, [3, 5, 8, 14] developed a method of trip chaining for the origin and destination inference with the following assumptions:

1. passengers return to the same location to board the bus where they alighted during the previous trip,
2. no private mode of transportation is used between trips,
3. passengers do not walk a long (more than a certain threshold) distance to board a bus or train,

4. passengers end their last trip at the same location where they started their journey of the day.

Based on the above assumptions, Trépanier et. al. proposed a model which infers alighting stops by minimizing the distance between the alighting stop of the current trip and boarding of the next trip [4] . They applied their method on AFC data from Quebec, Canada and inferred 66% of the trips. Similarly, Wang et al. proposed a method which combines AVL data with AFC data from London to infer the origin and destination of different trips and validated the results using bus passenger origin and destination survey (BODS) data [11]. Then Seaborn et al. stated some rules for trip chaining such as maximum acceptable transfer time of 20 minutes for underground subway-to-bus, 35 minutes for the bus-to-underground subway, and 45 minutes for bus-to-bus trips [21]. Building on the work of [21] and [11] in estimating origin-destination matrix using London smart card (Oyster) data and iBus vehicle location data, Gordon et al. specified the importance of the return trips, bus wait time, repeated service and circuity in trips [14]. The researchers suggested a circuity rule to account for the return trips. By using 750m as the maximum alighting distance, circuity factor of 1.7 and minimum transfer time of 5 minutes and maximum time from 30 to 90 minutes, they inferred 96% of the boarding locations and 74.5% of the alighting locations.

Nassir et al. used AFC data with General Transit Feed Specification (GTFS) data [22] instead of commonly used AVL data to infer origins and destinations [10]. They used the closest stop found within an upper bound distance of the smart card tag location as the boarding. Using the route information given in the AFC tag (transaction), a search is done for a trip closest in time within an interval of AFC transaction time. Using that trip, the stop found closest to the next boarding is inferred as the alighting stop given that the distance between inferred alighting and next boarding is less than 0.5 miles. Gordon et al. extended the research on origin-destination estimation of smart card users to non-smart card transit users [23]. They proposed a scaling method for expanding the OD matrix using the fare box data from London and compared the results with the Iterative Proportional Fitting (IPF) method.

Researchers have also tried to validate the trip chaining assumptions either by doing

a survey [21], [11] or using data from closed transit systems [16]. For example, Farzin validated the assumptions of the closest stops and daily symmetry using a travel diary survey in New York, which showed 90% accuracy [7]. Similarly, Alsger et al. used South-East Queensland public transport smart card data, which has both boarding and alighting information, to implement and validate the current trip chaining algorithms [16]. The researchers also suggested some improvements in the current algorithm, e.g., the alighting of the last tag on a day is the stop nearest to the first boarding of the day on the given transit route. They also suggested the average distance between the actual and estimated alighting stops as 0.33 miles instead of 0.5 miles. Of course, this distance parameter can vary for different transit systems, which we try to relax in this study.

Recent research on trip chaining has pointed out some limitations in trip chaining algorithms and suggested improvements for that [24]. For example, Munizaga identified that wrong alighting can be inferred if a passenger takes a bus which runs in both directions to go a few blocks away because the passenger would just cross the street to board the next bus rather than taking a long route in the opposite direction [13]. To alleviate this problem, the researchers suggested a cost function which is the sum of the current transaction time and the walking time multiplied by some penalty factor obtained from a discrete choice model. The adopted methodology inferred 80% of the trips using data from Santiago, Chile. The algorithm proposed in Chapter 4 avoids such situations by discarding the trip which is less likely to be taken by the passenger. He and Trepaniér followed their previous work [4] and proposed a method to infer the boarding and alighting of unlinked trips. The method multiplies the temporal and spatial probabilities calculated using historical location and time of tags to infer the potential alighting.

The quality of trip chaining results depends on fare collection system correctly recording the tag information which is assumed to be correct by most of the studies. This assumption may result in a wrong inference of boarding, alighting or especially transfer detections. Robinson et al. pointed out various causes for why different systems may not record correct information [25]. The possible causes are AVL system failure, card reader failure, software failure, etc. They proposed a method to identify

such erroneous smart card data and suggested where transit agencies should target resources to enhance the performance of their AVL and AFC systems. They applied the proposed method to Singapore smart card data and found that alighting for about 7.7% of the tags was found one stop before the actual alighting location and for 0.7% of the tags, the alighting location was found one stop after the actual alighting.

While applying the current trip chaining algorithms to the Twin Cities' AFC data, similar errors in results were found. To improve the accuracy of the results, the current research proposes a robust trip chaining method to alleviate the effect of various assumptions on the parameters such as GPS inaccuracy (buffer zone for boarding stop inference), finding most likely trip from GTFS data, etc. The method is similar to the one used for map matching problem for multi-modal transportation network modeling [26] and can be applied to other transit systems with any smart card data structure. The research also deals with complex transit systems consisting of "pay-exit" buses (passengers tap their card while alighting) such as Twin Cities, in which case passengers' alighting is known but not their boarding.

2.2 Origin Destination estimation using APC data

APC systems collect information about the number of passenger boarding and alighting at each transit stop. OD estimation using the boarding and the alighting counts is a classic problem, which is hard to solve. The problem requires solving an underdetermined system of equations, in which case the number of unknowns to solve is far more than the number of equations available. To deal with this underdetermined system of equations, various methods have been proposed in the literature, which are summarized below:

1. *Iterative Proportional Fitting (IPF) method* This is a popular and easy-to-apply method to evaluate the OD matrix using count data [27, 28]. The method starts with a base matrix, which is improved iteratively by multiplying the columns and rows of the matrix by a constant factor. The base matrix can be taken as a null matrix or any other seed matrix. Mishalani et al. found that using onboard survey data as a base matrix gives more accurate results than using null base matrix [29].

The method has several issues such as the problem of non-structural zeros [27], due to which a zero entry remains zero in every iteration. The method also fails to converge if the number of zero entries become large in the matrix.

2. *Bayesian inference methods*: These methods use Bayesian approach to evaluate an OD matrix by formulating the problem as a partially observed Markov chain and utilizing prior information along with current observations of count data [30–33].

3. *Optimization methods*: As there are multiple solutions possible for this system of equations, these methods try to find the one, which optimizes an objective function. The objective can be maximizing entropy [34] or the likelihood [35–37] function. With isotropic Gaussian noise, the maximum likelihood estimation turns into a classic least squares problem.

Another class of optimization methods consider the above objectives along with a regularizer. The regularizer helps to mitigate the ill-posedness of the system of equations [38]. The regularization can be included as a least square term between the unknown and a prior OD matrix obtained from a survey or from domain knowledge. This technique is quite popular in the literature. For example, Cascetta and Nguyen minimized generalized least square objective with a prior matrix [28], Van Zuylen and Willumsen maximized the relative entropy or minimized the Kullback-Leibler (KL) divergence of unobserved and observed flow distributions [34]. This approach tries to force the solution, as close to the prior matrix as possible which may result in poor estimates if the prior or seed matrix used is not reliable.

In this research, we evaluate the transit route OD matrix using APC data. The problem is the estimation of the flow of passengers between stops for a single trip. The route matrix problem has a special structure that provides an extra piece of information to reduce the ill-posedness of the system of equations. The estimation requires the selection of the correct estimate out of the multiple solutions. We use an estimation method that encourages the sparse OD matrix using l_0 norm regularizer. This helps in mitigating the ill-posedness of the system and offers interpretability [17] as there is only a subset of the origin-destination pairs which carries flow in an actual OD matrix. The method is popularly known as compressed sensing [18] and can also be viewed as

the least absolute shrinkage and selection operator (LASSO) regression proposed by Tibshirani [39].

Chapter 3

Transit Automated Data

The methods presented in this thesis were developed and tested using data from the Twin Cities' public transportation network. Although the data used for this research comes from a particular public transport agency, the methods are applicable worldwide. In this chapter, we describe the network structure of the Twin Cities' transit network and introduce the data sources used to conduct this research.

3.1 Twin Cities' public transportation network

Metro Transit is the transportation resource for the Twin Cities, offering an integrated network of buses, light rail and commuter trains as well as resources for those who carpool, vanpool, walk or bike [40]. In 2019, Metro Transit manages more than 190 transit routes on more than 13,000 stops all over the Minneapolis-St. Paul region and its suburban areas. The automated data used in this study is collected by Metro Transit.

3.2 Transit data

The different data sources used to conduct this research are General Transit Feed Specification (GTFS), Automatic Fare Collection (AFC), Automatic Passenger Count (APC), and Automatic Vehicle Location (AVL) data. There are various benefits of using automated transit data for transit planning. It offers several advantages [11] over traditional surveys by:

1. providing a link to passenger's trips over a longer period of time
2. providing information about the share of different transit commuters (e.g. students, workers, etc.)
3. storing the information in SQL database systems and using it efficiently
4. providing various research opportunities for analyzing passengers' travel pattern

It can be classified as follows:

3.2.1 Automatic Fare Collection (AFC) Data

The AFC data used for this research comes from the University of Minnesota student transit pass (U-Pass) transactions. The AFC system records the fare related information when a passenger pays for a trip. This includes a particular serial ID assigned to the pass, date and time of the tag, route information, geographical coordinates of the tag, fare type, and transfer information.

3.2.2 Automatic Passenger Count (APC) Data

The automatic passenger count system records date, time, transit route, stop and trip information, number of boarding and alighting at every stop, and geographical coordinates of stop locations. The primary purpose of this data is to evaluate ridership at different aggregation level. It is useful to evaluate service frequency based on the demand. This data is used to obtain boarding and alighting counts which are used as inputs in the method described in Chapter 5.

3.2.3 Automatic Vehicle Location (AVL) Data

The automatic vehicle location system records date, time, transit route, stop and trip information, departure and arrival time at time point stops, and geographical coordinates of stops. The system is primarily used to provide real-time bus arrival information to passengers. It is also used to evaluate the quality of service of passengers by evaluating the delay experienced by them. This data is used to calibrate the probability distribution of delay of buses required for robust trip chaining algorithm described in Chapter 4 in §4.2.

3.2.4 General Transit Feed Specification (GTFS) Data

General Transit Feed Specification (GTFS) is a standard format of transit schedule data provided by many transit agencies all over the world [22]. It contains schedule information of the buses and light rail, including their stop location, route information, scheduled arrival and departure time. Different tables of GTFS data which are used for this research are briefly described below:

1. *Agency*: It contains information of one or more transit agencies offering service
2. *Stops*: It contains information about the individual location of stops where passengers can pick up or drop off.
3. *Routes*: It contains information about individual routes such as route ID, name, type, and sort order.
4. *Trips*: It contains the information about trips made by individual route throughout the day
5. *Stop times*: It contains information about the time that a vehicle arrives and depart from individual bus stops for each trip.
6. *Calendar*: It contains information about service ID for the different type of service. For example, the service of weekdays, weekends, and holidays can be different.

For trip chaining algorithm described in Chapter 4 in §4.2, we need to select the appropriate service ID for the study period and then query the data.

Chapter 4

Origin and Destination Inference using Trip Chaining

In order to observe passenger movement in a transit network using AFC data, boarding, and alighting location and times must be inferred based on the incomplete information available. This is achieved by using a suitable trip chaining algorithm. This chapter describes the development of robust trip chaining algorithm. Before that, various issues related to existing trip chaining algorithm which we try to address in this research are explained.

4.1 Problems in existing trip chaining algorithm

This section explains problems associated with current trip chaining algorithm and the desired improvements. We use reference of trip chaining algorithm developed by Nassir et al. to point out various problems [10]. The algorithm uses consecutive tags of a card holder which are termed as "current" and "next" tag throughout this thesis. For the last tag of the day, next tag can be assumed as the first tag of the day. First, the trip chaining algorithm developed by [10] is summarized below:

1. Read AFC data and select the "current" and "next" tag.
2. Extract GTFS schedule of the current tag's route and direction to find the closest stop to the current tag location.

3. Go to step 4 if the distance between the current tag and closest stop found is less than 0.1 miles otherwise exclude the tag and go back to step 1.

4. Find a trip within TrT–α and $TrT + \beta$ closest to the current tag time. Here, TrT is the current tag time and α and β are schedule adherence parameters determined using AVL data.

5. Find the closest stop to the next tag location on the trip found in step 4 for the stops sequence greater than the stop found in step 2.

6. Go to step 7 if the distance between the inferred alighting location of the current tag and the next tag location is less than 0.5 miles, otherwise exclude the tag.

7. Go to step 8 if the boarding time of the next tag is greater than the alighting time of the previous tag, otherwise exclude the tag and go to step 2.

8. Determine if the current tag is the first tag of the day. If it is, mark it as "boarding", otherwise determine if it is a transfer. A detailed discussion about transfer detection is given §4.2.4.

The method, although working in most of the cases, may result in wrong inference or no inference in some cases. These cases are described below.

4.1.1 The sub-route problem

To manage some of the transit routes efficiently, the Twin Cities transit system has sub-routes for most of the high frequency routes. For example, route 2 has sub-routes 2A, 2C, 2E and route 3 has sub-routes 3A, 3B, 3C, 3E, 3K. Generally, one of the sub-routes is more common than the others and runs throughout the day, whereas others are either short turns or branches to serve more areas. To better understand the sub-route problem, let us consider following instance (Figure 4.1):

A passenger took the bus route 2 from Coffman Memorial Union stop and alighted at Hennepin Ave and 8th Street to transfer to route 10. The current trip chaining algorithm selects any trip from GTFS data which is closest in time to the current tag time. If it selects the trip within route 2A that only goes up to TCF Bank Stadium

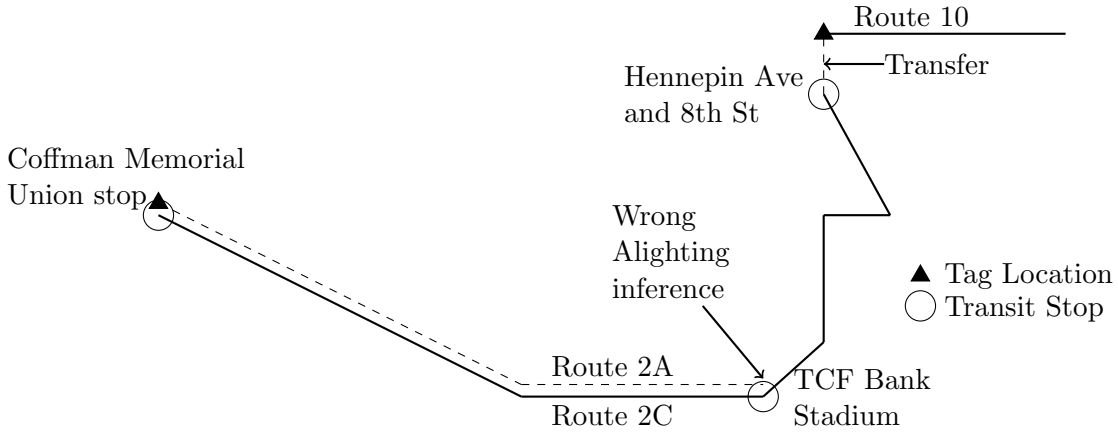


Figure 4.1: Incorrect alighting inference due to selection of incorrect sub route

stop and infer it as alighting stop, then the distance between this stop and the next tag location is more than the walking distance threshold and the algorithm does not infer any alighting stop (discards this record). In this case, a more robust inference method is required to correctly infer the trip within route 2C, which connects with route 10 at Hennepin Ave and 8th St.

4.1.2 The boarding stop inference problem

The GPS location of tags provided by AFC system may consist of location measurement errors [25]. If the algorithm simply finds the closest stop to the tag location, then a potentially wrong boarding stop inference may result in wrong trip inference, wrong alighting stop inference or no inference at all.

4.1.3 The "pay-exit" route problem

Because of high commuter demand to Downtown Minneapolis, Downtown St. Paul, and the University of Minnesota campus, some of the outbound bus routes in the evening peak let passengers enter the bus while boarding and pay while alighting (unlike the regular routes where riders tap while entering the bus). Such cases were not considered during previous studies. In these cases, we do not know the boarding but know the alighting location. Depending on the combination of tags made by a passenger throughout the day, missing boarding or alighting may or may not be inferred. This

arises four different cases depending on the consecutive tags of the passenger (Figure 4.2).

1. *Current tag (B1) is regular and next tag (B2) is regular*
   This is the normal case which has been considered previously in the research. Here, we know the boarding of the current as well as the next tag. Using the route and direction information of the current tag, we can infer the alighting location of the current tag.

2. *Current tag (A1) is pay exit and next tag (B2) is regular*
   In this case, we know the alighting of the current tag and boarding of the next tag. This is the easiest case among four cases as we need not to infer any location. The only thing to determine in this case is to detect whether or not the next tag is a transfer. Note that the possibility of inferring the boarding of the current tag depends on its previous tag. Similarly, the possibility of inferring the alighting of the next tag depends on its next tag.

3. *Current tag is regular (B1) and next tag (A2) is pay exit*
   This is the most difficult case among all as we know the boarding of the current tag and the alighting of the next tag which means alighting of the current tag and the boarding of the next tag is missing. Two sub-cases arise in this case depending on the bus route used.

   - If two different bus routes (which are not geographically parallel) are used for both tags, then we can find stops connecting two routes which gives the least distance between the inferred alighting of the current tag and the inferred boarding of the next tag.
   - If same or parallel routes are used for both tags, then we cannot infer the alighting of the current tag and boarding of the next tag. This sub case is quite usual for commuters who take a bus from sub-urban areas which is regular in the inbound direction in the morning but when they return to their home, the same bus is pay exit in the outbound direction in the evening. We propose a method of proportion in §4.2.3 to approximate these cases.

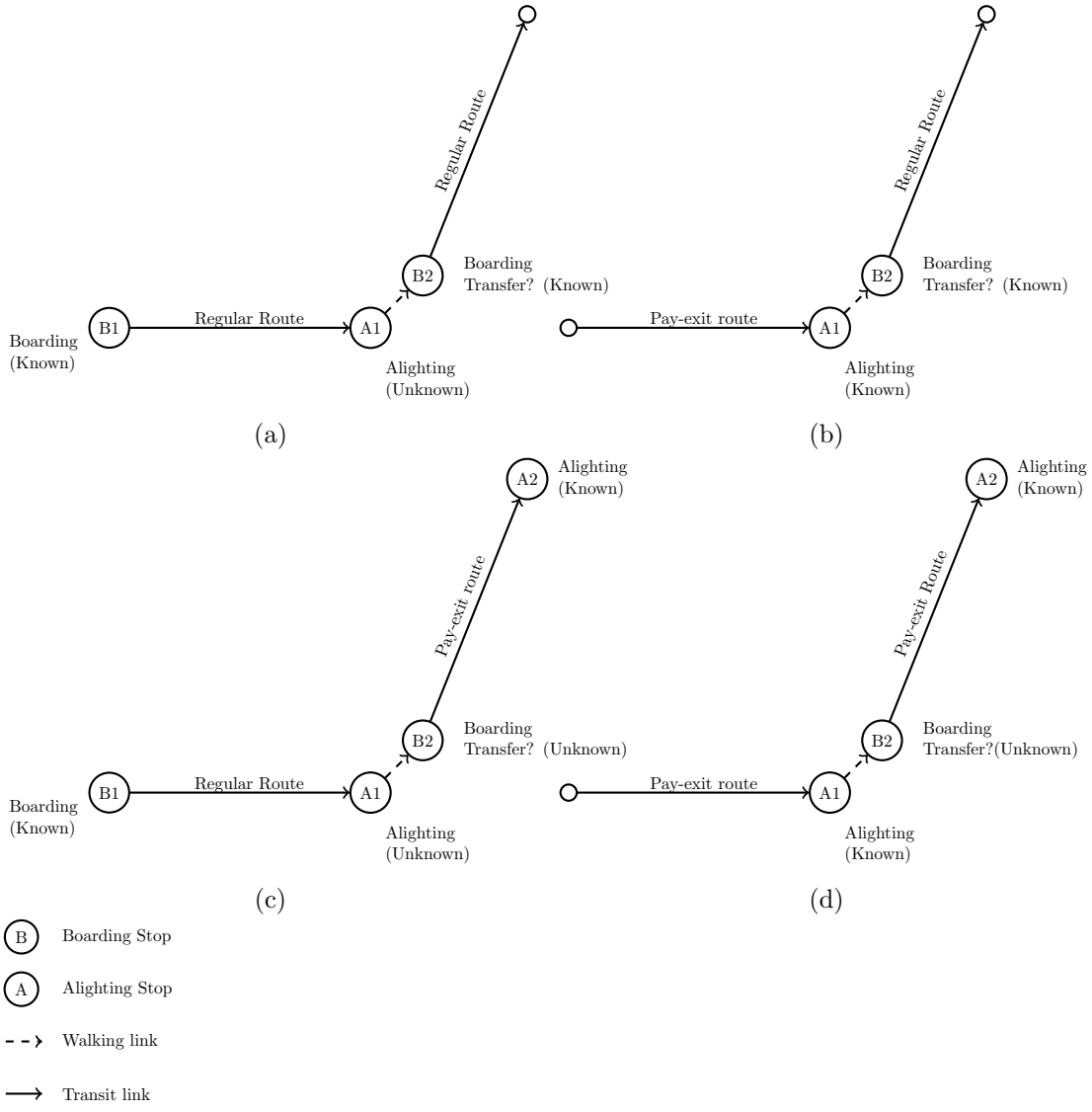


Figure 4.2: Four cases depending on the pay exit or regular route

4. *Current tag is pay exit (A1) and next tag (A2) is pay exit*
   In this case, we know the alighting of both current and next tag. We can make a search list of the stops that come before the alighting stop of the next tag and infer the boarding of the next tag by finding the stop closest to the alighting location of the current tag. Again, the boarding of the first tag may or may not be inferred depending on its previous tag.

4.2 The robust trip chaining algorithm

The trip chaining method described in this thesis is inspired by map matching algorithms used for multi-modal transportation network modeling [41], [26]. The map matching algorithm is used to map the public transit stops from GTFS data to a road network by creating a restricted shortest path problem. In this way, it avoids the problems like complicated road geometry, and lack of dynamic vehicle information like vehicle trajectory, speed, turning and heading. Similar methods are common for matching GPS locations to existing road networks to track the trajectory of a vehicle using probability models such as Hidden Markov Model [42]. The proposed trip chaining method also finds a set of candidate trips for a given AFC tag to reach the next tag, calculates the probability of each trip, then the most likely trip is found to infer the boarding and alighting stops. In this way, different problems faced by the current trip chaining algorithm are addressed. We start with the basic case when both of the consecutive tags are regular which can be applied to any transit system and then we can expand this method to specific cases for the Twin Cities data.

4.2.1 Trip set generation

Consider two consecutive tags n and $n+1$ of a particular card number on a given date. Using GTFS data, we can make a list of candidate stops $S_n = \{s_{nk}, k = 1, 2, ...\}$ found within a buffer distance of α miles of the tag location θ_n given route r_n and direction δ_n. The value of α can be suitably taken depending on the accuracy of the GPS. For example, previous studies have used $\alpha = 0.1$ miles to find the boarding stop. This will consider the possibility of all the stops which are close to the tag location θ_n being the boarding stop and help in obviating the problem of wrong boarding stop being selected.

The error in the GPS location is usually modeled using great circle distance [42] which is the shortest distance between two points on the surface of a sphere [43]. We can find the great circle distance d_{nk} between θ_n and s_{nk} as

$$d_{nk} = \mathcal{GC}(\theta_n, s_{nk}) \qquad \forall k \tag{4.1}$$

The next step is to find possible trips from these stop locations which go in the direction of the next tag location. For each stop s_{nk}, find the possible trips $\mathcal{T}_{nk} = \{tr_{kl}, l = 1, 2, ...\}$ which are within τ minutes of tag time t_n assuming that bus can be late or early on a given stop s_{nk} by τ minutes. This delay parameter τ is flexible and can be adjusted for the given algorithm. With greater value of τ, more trip options will be created. This will obviate the problem of incorrect sub-route (§4.1.1) trip being selected. Then we calculate the delay for different trips as:

$$\Delta_{kl} = |t_{tr_{kl}} - t_n| \qquad \forall k, l \tag{4.2}$$

Using the trip information, for each trip l, find a set of alighting stops $A_{nkl} = \{a_{klm}, m = 1, 2, ...\}$ which are within ϵ miles of next tag location θ_{n+1}. Again, ϵ is flexible and can be assumed as any suitable value. This will avoid the problem of finding wrong alighting stop mentioned in [13]. Let $\mathcal{IV}_{klm}$ be the in-vehicle time for the trip tr_{kl} with alighting stop a_{klm} and w_{klm} be the walking distance from alighting location a_{klm} to the next tag location θ_{n+1}. All the potential stops and trips can be connected via a graph shown in Figure 4.3 as an example.

4.2.2 Probability calculation for possible trips

Let $P(s_{nk})$ be the probability of boarding stop s_{nk} from tag location θ_n. This probability is a function of great circle distance d_{nk} which is created because of the GPS inaccuracy and can be modeled as a zero mean Gaussian distribution [44], given as:

$$P(s_{nk}) = f(\sigma_k, d_{nk}) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp^{-0.5(\frac{d_{nk}}{\sigma_k})^2} \qquad \forall k \tag{4.3}$$

If we assume s_{nk} was the actual boarding location, then d_{nk} is an estimate of the

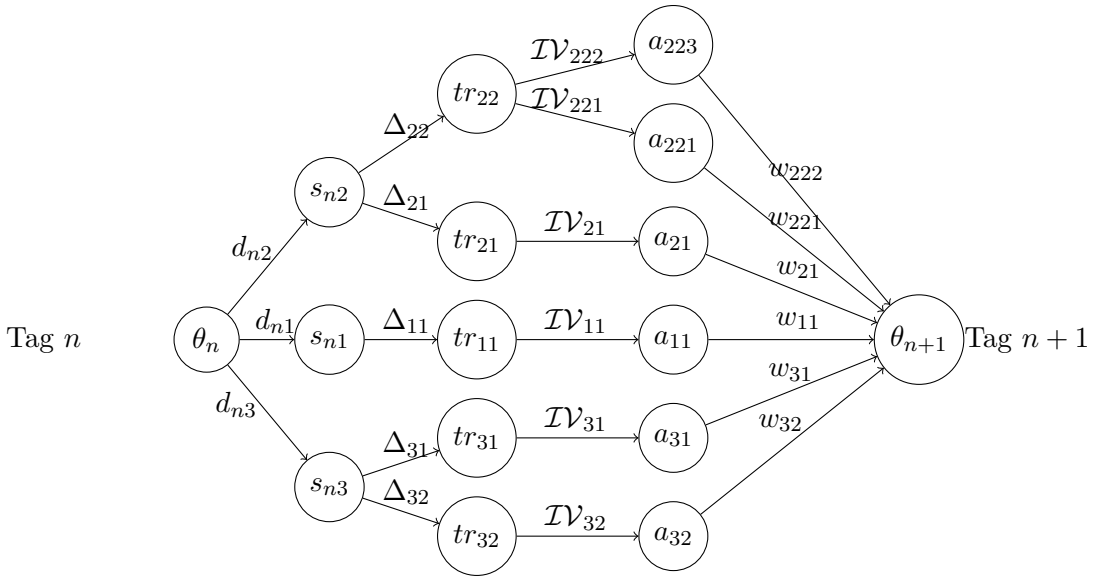


Figure 4.3: Network of possible trips

magnitude of GPS error. The standard deviation of these values, i.e. σ_k, is our estimate of the GPS error. We estimate σ_k using the median absolute deviation, which is a robust estimator of standard deviation. The value of σ_k can be given as:

$$\sigma_k = 1.4826 * \text{median}(d_{nk}) \qquad \forall k \tag{4.4}$$

The probability of taking a trip tr_{kl} from stop s_{nk}, i.e., $P(tr_{kl}|s_{nk})$, is a function of bus delay Δ_{kl}:

$$P(tr_{kl}|s_{nk}) = f(\Delta_{kl}) \qquad \forall k, l \tag{4.5}$$

The probability distribution function $f(\Delta_{kl})$ of bus delay can be calculated using AVL data, which contains vehicle arrival times on limited stops for a given bus route trip l. We can model the probability of reaching the next tag location θ_{n+1} by taking trip tr_{kl} and alighting at stop a_{klm} using a multinomial logit route choice model given as:

$$P(a_{klm}|tr_{kl}, s_{nk}) = \frac{\exp^{-(\beta_1 \mathcal{IV}_{klm} + \beta_2 \frac{w_{klm}}{s})}}{\sum_{p,g} \exp^{-(\beta_1 \mathcal{IV}_{kpg} + \beta_2 \frac{w_{kpg}}{s})}} \quad \forall l, k \tag{4.6}$$

where, s is the walking speed which is assumed as 3.0 miles per hour. β_1 and β_2 are the parameters which shows the disutility of walking in comparison to in-vehicle travel time according to user behavior.
Finally, assuming the random variables describing the probability distributions are independent, we can evaluate the probability of traversing from location θ_n to θ_{n+1} using any of the trips by multiplying (3), (5) and (6) which is the product of the following components.

- GPS inaccuracy of the current tag
- Bus delay of the current tag
- Route choice model consisting of in-vehicle and walking time between the current tag and the next tag.

$$\begin{aligned} P(a_{klm}, tr_{kl}, s_{nk}|\theta_n, \theta_{n+1}) &= P(a_{klm}|tr_{kl}, s_{nk}, \theta_n, \theta_{n+1})P(tr_{kl}|s_{nk}, \theta_n, \theta_{n+1})P(s_{nk}|\theta_n, \theta_{n+1}) \\ &= f(\sigma_k, d_{nk})f(\Delta_{kl})P(a_{klm}|tr_{kl}, s_{nk}) \quad \forall l, k, m \end{aligned} \tag{4.7}$$

Hence, the most likely boarding and alighting stops for this tag n can be inferred using the trip for which $P(a_{klm}, tr_{kl}, s_{nk}|\theta_n, \theta_{n+1})$ is maximum.

4.2.3 Extension to pay-exit cases

If there is a combination of pay-exit and regular tags (§4.1.3), then the probability calculations change according to available information. These cases are discussed below:

Current tag is pay exit and next tag is regular

In this case, the probability of each trip consists of three components:

- GPS inaccuracy of the current tag
- Bus delay of the current tag
- Route choice model consisting of only walking time between the current tag and the next tag.

The final expression is given below:

$$P(a_{klm}, tr_{kl}, s_{nk}|\theta_n, \theta_{n+1}) = f(\sigma_k, d_{nk}) f(\Delta_{kl}) \frac{\exp^{-(\beta_2 \frac{w_{klm}}{s})}}{\sum_{p,g} \exp^{-(\beta_2 \frac{w_{kpg}}{s})}} \quad \forall l, k \tag{4.8}$$

Current tag is regular and next tag is pay exit

For this case, if two different routes are used for making these two trips, then the probability of each alternative to go from the current boarding to the next alighting consists of three components:

- GPS inaccuracy of the current tag and the next tag
- Bus delay of the current tag and the next tag
- A common route choice model consisting of in-vehicle travel time of the two trips and the walking time between the trips.

The final expression is given below:

$$P(a_{klm^{12}}, tr_{kl^1}, tr_{kl^2}, s_{nk^1}, s_{nk^2}|\theta_n, \theta_{n+1}) = f(\sigma_k^1, d_{nk}^1) f(\sigma_k^2, d_{nk}^2) f^1(\Delta_{kl}^1) f^2(\Delta_{kl}^2) \frac{\exp^{-(\beta_1 \mathcal{IV}_{klm^1} + \beta_1 \mathcal{IV}_{klm^2} + \beta_2 \frac{w_{klm^{12}}}{s})}}{\sum_{g,p^1,p^2} \exp^{-(\beta_1 \mathcal{IV}_{kpg^1} + \beta_1 \mathcal{IV}_{kpg^2} + \beta_2 \frac{w_{kpg^{12}}}{s})}} \quad \forall l, k \tag{4.9}$$

If both tags use the same or parallel routes, we can make use of APC data to assign the alighting of the current tag and boarding of the next tag. Usually some particular stops at the end of the routes are more common stops for alighting. Using route information, we calculate the proportion of alighting at these stops for each route, then assign the required boarding and alighting stops proportionally for each case in the AFC data. In this way, we may not get exact inference in the individual level, but on an aggregate level, the results will be consistent. Anyhow, the percentage of these cases in the AFC database is very low.

Current tag is pay exit and next tag is pay exit

In this case, the probability of each trip consists of three components

- GPS inaccuracy of the next tag
- Bus delay of the next tag
- route choice model consisting of in-vehicle travel time and walking time of the next trip.

The final expression is given below:

$$P(a_{klm}, tr_{kl}, s_{nk}|\theta_n, \theta_{n+1}) = f(\sigma_k, d_{n+1,k}) f(\Delta_{kl}) f(\frac{\exp^{-(\beta_1 \mathcal{IV}_{klm} + \beta_2 \frac{w_{klm}}{s})}}{\sum_{p,g} \exp^{-(\beta_1 \mathcal{IV}_{kpg} + \beta_2 \frac{w_{kpg}}{s})}}) \quad \forall l, k \tag{4.10}$$

4.2.4 Transfer detection

Transfer information given in the AFC data may not be reliable. Consistent with the fair policy, the AFC system considers a tag as a transfer if it has been made within 150 minutes of the previous tag time. The method described in [10] is used to detect transfers. The method infers next tag as transfer if it has been made within 30 minutes and boarding if it has been made after 90 minutes of alighting. Between 30 and 90 minutes, after alighting at a station, the walking time (W) and setback delay time (D) (due to possible minor activities like buying coffee or newspaper) is considered and a time t_{acc} is calculated which is the time when boarding stop becomes accessible. Then, the number of opportunities (N_{opp}) to catch the next bus is calculated between the time t_{acc} and the actual boarding time of the next tag by counting the number of trips in GTFS data within the time range. If $N_{opp} \leq 1$, we infer the next tag as transfer, otherwise, there is a possibility of an activity and we mark the next tag as boarding. Pseudocode for this trip chaining algorithm is given in Algorithm 1.

Algorithm 1 Robust Trip Chaining Algorithm

Data structures
n: an AFC tag
pe: 1, if tag is pay exit, 0, otherwise
seq: sequence number of the tag serial number for the given date
ser: sequence number of a transit stop for a given trip ID in GTFS data
P: list of possible stops around tag location
L: list of possible trips for a given stop
All other notations are consistent with Appendix A
function FindPossibleStops($tag[n]$)
    $P \leftarrow []$
    $st\_list \leftarrow$ find a list of stops for $tag[n].r$ and $tag[n].\delta$ from GTFS
    for each stop s in st_list **do**
        if $dist(s, tag[n].\theta) < \alpha$ **then**
            append s to P
    return P
function FindPossibleTrips(p)
    $L \leftarrow []$
    $tr\_list \leftarrow$ find all the trips for given stop $p.r$, $p.\delta$ from GTFS
    for each trip l in tr_list **do**
        if $abs(l.dep - tag[n].t) \leq \tau$ **then**
            append l to L
    return L
function InferBoardingAlighting(l, $tag[n]$, $tag[n+1]$)
    if the inference is for alighting **then**
        $al\_stops \leftarrow$ find stops with stop sequence greater than $l.ser$
        return alighting stops within distance ϵ of the $tag[n+1]$
    else
        $bo\_stops \leftarrow$ find stops with stop sequence less than $l.ser$
        return boarding stops within distance ϵ of the $tag[n]$
Algorithm
for each n **do**
    $Prob \leftarrow []$
    if $tag[n].seq =$ last tag of the day **then**
        take $tag[n+1] =$ first tag of the day for that serial number
    $P \leftarrow$ FindPossibleStops($tag[n]$)
    for each stop p in P **do**
        $L \leftarrow$ FindPossibleTrips(p)
        for each trip l in L **do**
            Depending on $tag[n].pe$ and $tag[n+1].pe$
            $L \leftarrow$ InferBoardingAlighting(l, $tag[n]$, $tag[n+1]$)
            Calculate $Prob[l]$
    Find the trip with maximum probability
    Infer the boarding and alighting of $tag[n]$ and $tag[n+1]$ based on that trip

Chapter 5

Origin and Destination Estimation using Compressed Sensing

In this chapter, we describe the method to estimate the route level OD matrix using boarding and alighting counts available from APC data.

5.1 Preliminaries

Let N be the set of stops along a transit route at which passenger board or alight. We consider the boarding and alighting in a single direction. Let b_i and a_i be the observed number of passengers who board and alight at stop $i = (1, 2, ..., |N|)$ respectively. The values of b_i and a_i are obtained from APC data. Let $X = \{x_{ij}\} \in \mathbb{R}^{|N|\times|N|}$ be the origin-destination flow matrix, where x_{ij} denotes the number of passengers boarding at stop i and alighting at stop j. The overall setup is shown in Figure 5.1. Let $x \in \mathbb{R}^{|N|^2}$ be the vectorized form of matrix X i.e., $x = Vec(X)$.

5.2 Formulating transit route OD estimation problem

The estimation procedure is subject to the following constraints:

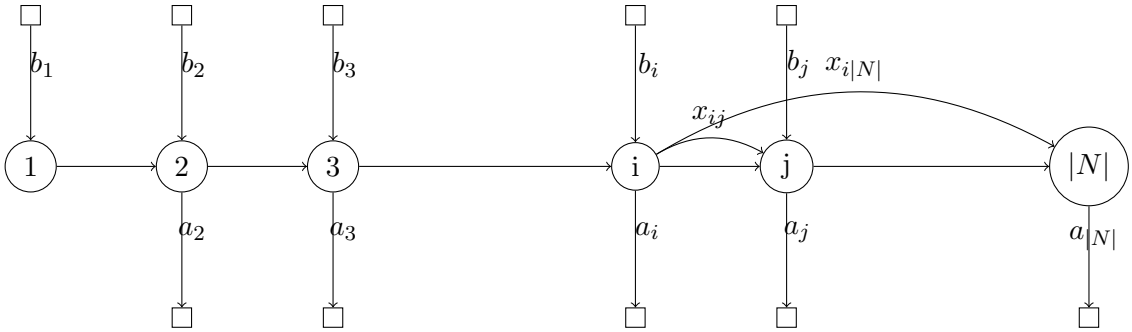


Figure 5.1: Transit route origin-destination (OD) flow

1. If we sum the values of x_{ij} along all the columns, then we get the total number of passengers boarding at stop i i.e. b_i,

$$\sum_{j=1}^{|N|} x_{ij} = \ b_i \ \forall \ i \in N \tag{5.1}$$

2. Similarly, if we sum the values of x_{ij} along all the rows, then we get the total number of passengers alighting at stop j, i.e., a_j

$$\sum_{i=1}^{|N|} x_{ij} = \ a_j \ \forall \ j \in N \tag{5.2}$$

3. The total number of boarding at all the stops should be equal to the total number of alighting.

$$\sum_{j=1}^{|N|} b_j = \sum_{i=1}^{|N|} a_i \tag{5.3}$$

4. The number of boarding and alighting at the same stop is zero, which means the diagonal elements of the matrix X should be equal to zero.

$$x_{ii} = \ 0 \ \forall \ i \in N \tag{5.4}$$

5. As the transit vehicle runs in a single direction, a passenger boarding at one stop

		Alighting						
		1	2	.	.	.	n	Total Boarding
Boarding	1	0						b_1
	2	0	0					b_2
	.	0	0	0				.
	.	0	0	0	0			.
	.	0	0	0	0	0		.
	n	0	0	0	0	0	0	b_n
	Total Alighting	a_1	a_2	.	.	.	a_n	T

Table 5.1: OD matrix for a route in a single direction

cannot alight at the previous stops that vehicle has already visited. This means,

$$x_{ij} = 0 \ \ \forall \ i \ > j, \quad \forall \ i, \ j \in N \tag{5.5}$$

6. The total load on a link between two stops is equal to the passengers boarding between those stops.

$$\sum_{i=1}^{k} (b_i - a_i) = \sum_{i=1}^{k} \sum_{j=k+1}^{n} x_{ij} \tag{5.6}$$

By imposing these constraints, the structure of the matrix will look as in Table 5.1.

We can express the linear constraints (5.1) - (5.6), in form of a matrix as

$$\mathcal{A}\,(x) = b \tag{5.7}$$

where, $\mathcal{A} \in \mathbb{R}^{p \times |N|^2}$ is the linear map (which is a matrix in this case) for p number of constraints and $b \in \mathbb{R}^p$ represents the constant vector for these constraints. In the following subsections, we describe the proposed solution to the given problem.

5.3 Transit route OD estimation using compressed sensing technique

As discussed, 5.7 is usually an ill-posed problem for which one can expect multiple solutions. A generic regularizer can help in mitigating the ill-posedness of the problem.

One such regularizer is the generalized least square with prior matrix available from survey data. The quality of the solution depends upon the availability of a good prior matrix as the optimal solution is forced to be as close as possible to the prior matrix. We can use other regularizers based on the domain knowledge on the space of the plausible OD flows in the network [17]. To use one such regularizer, we make the following assumption:

Assumption 1 *The planted OD matrix in the set of linear equations is sparse which means that the flow between many of the OD pairs should be equal to zero. The observed flow is only due to a small subset of* $\frac{N(N-1)}{2}$ *pairs.*

The intuition behind the above assumption is that there is a large number of OD pairs for a transit route, but the travel happens only along few pairs. For example, during the morning peak hours, there are only a few popular origin stops such as residential locations and few destinations stops such as central business areas, park and rides, etc. Moreover, it is unlikely that passengers boarding at initial stops of the route will alight at all the following stops. This makes the flow between most of the OD pairs equal to zero. This is opposite to the solution evaluated using entropy maximization, which tries to achieve the solution, as uniform as possible to minimize the errors. The sparsity as a regularizer has been used before for highway network OD estimation and has found promising results [17, 45–47]. For example, Menon et al. leverages sparsity in highway OD matrix to estimate a set of suitable traffic analysis zones (TAZs) and use those zones to evaluate an OD matrix [17]. The method proposed in [17] has a bi-level structure with sparse OD estimation on upper level and traffic assignment using user equilibrium at lower level. The use of non-negativity constraints for improving the solution is also emphasized. We use a similar optimization for the transit route OD estimation problem, which has a special structure as we get an extra set of constraints because of the transit movement in one direction. We also describe the conditions under which sparse recovery is possible.

5.3.1 Using sparsity as the regularizer for OD estimation

To achieve the sparsity in the solution, we minimize the number of non-zero entries in the solution, which can be done by minimizing l_0 norm of the vector x. We can state

the problem as the minimization of l_0 norm of x subject to linear constraints. The optimization formulation is given below:

$$\begin{aligned} \underset{x \geq 0}{\text{minimize}} \quad & \|x\|_0 \\ \text{subject to} \quad & \mathcal{A}(x) = b \end{aligned} \tag{5.8}$$

The non-negativity should not be dropped from (5.8) as it helps to mitigate the ill-posedness of the problem [17]. Using Lagrangian relaxation, the linear constraints can be included in the objective function as a least square term and formulated as following:

$$\underset{x \geq 0}{\text{minimize}} \quad \|A(x) - b\|_2 + \mu \|x\|_0 \tag{5.9}$$

The problem (5.9) tries to find the sparse vector x planted in the given ill-posed system of linear equations. The regularization parameter μ controls the sparsity of the vector and requires tuning to get the best results. A higher value of the μ will impose more sparsity in the solution. When $\mu = 0$, (5.9) reduces to an ordinary least squares problem. The optimization program (5.9) is useful for the APC data when the total number of boarding and alighting do not match as the least square term will try to find a solution which best explains the observed flows. This happens quite often in the APC systems due to the errors in recording data. The given problem (5.9) is an NP-hard as the minimization of l_0 norm cannot be done in polynomial time. Recent work in compressed sensing has proposed a tightest convex relaxation of the l_0 norm which is l_1 norm [48]. The problem (5.9) can be restated as follows.

$$\underset{x \geq 0}{\text{minimize}} \quad \|A(x) - b\|_2 + \mu \|x\|_1 \tag{5.10}$$

Where, $\|x\|_1 = \sum_i |x_i|$. (5.10) is a convex optimization program as the absolute value of x_i can be written as a set of linear inequality constraints. The use of l_1 norm is better than the l_2 norm (also called ridge regression) to achieve sparsity. This is because the l_1 norm ball has corner points that can intersect the given plane at the sparsest solutions, unlike l_2 norm ball. The problem can also be viewed as least absolute shrinkage and selection operator (or Lasso regression) proposed by [39] as given a set of observations, we try to estimate the coefficients which satisfies the given equations.

However, there is a key difference between compressed sensing and LASSO. The former provides conditions under which the linear map $\mathcal{A}$ nicely behaves and the uniqueness of the solution can be proved (these conditions are discussed in the next subsection). In other words, we can design $\mathcal{A}$ in such a way that it can guarantee to recover the actual solution. On the other hand, LASSO is a regression method in which we have no control over the data and we try to find the best coefficients which are sparse and satisfy the equations obtained from data. We can also interpret these estimates as a Bayesian posterior mode estimate when the regression parameters have independent Laplacian (i.e., double exponential) priors [49]. Now the natural question which arises is that when does solving (5.10) gives a good solution to (5.9). In other words, what natural conditions can be applied on a linear map $\mathcal{A}$ so that we can say that the solution is unique. Candés and Tao, 2005 proposed the idea of restricted isometry property (RIP) of the matrices, which states that if $\mathcal{A}$ satisfies the isometry property, then there exists a unique solution to the problem (5.10) which is equal to the solution of (5.9).

Definition 1 (Restricted Isometry Property (RIP)) *The linear map $\mathcal{A}$ has RIP with constants k and δ_k, if $\forall \|x\|_0 \leq k, \mathcal{A}$ behaves almost as an isometry in following sense i.e., l_2 norm of $\mathcal{A}(x)$ is close to the l_2 norm of vector x:*

$$(1 - \delta_k)\|x\|_2^2 \leq \|\mathcal{A}(x)\|_2^2 \leq (1 + \delta_k)\|x\|_2^2 \tag{5.11}$$

Loosely speaking, a matrix $\mathcal{A}$ satisfies RIP when δ_k is small (i.e. less than 1). When 5.11 is satisfied, then $\mathcal{A}$ approximately preserves the Euclidean length of $k-$sparse vectors and it cannot lie in the null space of $\mathcal{A}$. This property is important otherwise, there is no hope of recovering x from 5.7 [50]. The implication of RIP in compressed sensing is that the pairwise distances between $k-$sparse vectors must be preserved in the measurement space due to which the $\|x\|_1$ and $\|x\|_2$ can be scaled into each other by some constant. This idea is related to the old notion of Almost Euclidean subspaces. This results in the recovery of x close to exact k-sparse x by solving 5.8 with an overwhelming probability. RIP matrices are extremely common in practice and most of the random matrices satisfy this property. Based on the above definition of RIP, a theorem is proposed by Candés and Tao, 2005 [48].

Theorem 1 (Candés and Tao, 2005 [48]) *If $\mathcal{A}(x) = b$ and b is constructed using a*

sparse solution with $\|x\|_0 \leq k$*, and the RIP condition is satisfied with constants* δ_{2k} *and* δ_{3k}*, satisfying* $\delta_{2k} + \delta_{3k} < 1$*, then (5.10) can obtain a unique solution to the problem (5.9) with as few as* $\mathcal{O}\left(klog\left(\frac{|N|^2}{k}\right)\right)$ *number of equations.*

As the passenger flow cannot be negative, we can replace the l_1 norm with sum of the components of vector x, which allow us to use the gradient-based approaches to solve the optimization program (5.10) efficiently. If we have some idea about the number of non-zero entries (say less than k), we can constraint the solution as follows:

$$\begin{aligned} \underset{x \geq 0}{\text{minimize}} \quad & \|A(x) - b\|_2 \\ \text{subject to} \quad & \|x\|_0 \leq k \end{aligned} \tag{5.12}$$

We use the optimization program (5.8) with l_1 norm for solving the transit route OD estimation problem. The problem is convex and can be solved easily using a standard convex optimization solver such as CVX [51]. We could also employ an iterative algorithm proposed in [39] to evaluate a sparse solution but the algorithm does not guarantee convergence to a unique solution. Applications of this method are presented in §7.2.

Chapter 6

Implementation

Various issues related to automated data needs to be resolved before implementing the methods. This includes the cleaning of data, removing inconsistencies, and inferring fields required for the algorithm. This chapter describes the data preparation steps along with implementation strategies of the algorithm described in §4.2.

6.1 Processing of AFC data

AFC data does not contain a sequence of trips made by a passenger in a day. Based on the time of a transaction, a sequence field was added to the data which keeps track of the sequence of the tags made by a passenger on a particular day. A pay-exit field was also added to the data by checking the buses and their direction in which they are pay-exit. The field takes the value 1 if given transaction route is pay-exit. Several other issues with data were resolved before running the trip chaining algorithm. For example, AFC data for light rail does not have geographical coordinates but contains the station information where the passenger boarded the light rail, in which case we do not have to search for possible boarding stops. Another issue is that light rail AFC data does not have direction information. This is because light rail stations serve the trains in both directions. We inferred the direction of light rail trips using the next tag location.

After the initial data processing, there are still some tags which do not have any geographic information. These mainly consist of the buses not operated by Metro Transit (e.g operated by Minnesota Valley Transit Authority (MVTA), or First Transit).

We removed such entries for the analysis because the GTFS data was unavailable for these services. The data also contains some tags which have a geographic location outside the transit service region, so we removed such entries from the dataset. We also removed the cases where a single tag is made by a passenger on a day as trip chaining requires at least two trips made by a passenger in order to estimate the origin and destination. Table 6.1 shows the number of tags in the data set for four typical weekdays (March 07, 2016 to March 10, 2016).

Table 6.1: Tag Description

Description	**Number of tags**	**Percentage**
Total tags	85,456	
Missing geographical coordinates	4,785	5.6
Outlier geographical location	3,515	4.1
Single tags	10,782	12.6
Total remaining tags	66,374	77.7

6.2 Processing of APC data

The APC data used for the method described in §3.2.2 was uploaded to Microsoft SQL server and queried using the R package RODBC [52]. We select A-Line, which is a bus rapid transit (BRT) route in Twin Cities for this analysis. It serves 20 stations along Snelling Av and 46th St. We select a trip from the data during peak hour. The results are presented in §7.2.2

6.3 Trip chaining model calibration

6.3.1 Gaussian model for GPS inaccuracy

To calibrate (4.3)-(4.4), we created a list of the AFC tag locations for which only one stop is found within a buffer distance of 0.1 miles and calculated the values of the d_{nk}. These stops can be regarded as ground truth data required for calibration. Using these values, σ_k was calculated equal to 55.25 ft.

6.3.2 Bus delay probability distribution

As mentioned before, AVL data contains bus arrival time at limited stops. Therefore, the available arrival times are used to calculate the probability of bus route being early or late. For this purpose, a discrete distribution for the bus delay distribution (4.5) with a class range of one-minute intervals is calibrated.

6.3.3 Route choice model

For (4.6), we assumed the value of $\beta_1 = 1$, $\beta_2 = 2$, and the walking speed, $s = 3$ miles per hour for our route choice model. These values are consistent with the literature [53–55].

Chapter 7

Applications

In this chapter, we analyze the results obtained after applying the methods described in Chapter 4 and 5 on Twin Cities' transit data and then present a few applications of these results.

7.1 Application of AFC data

7.1.1 Analysis of the results

After data preparation, Algorithm 1 was implemented in R [56] for U-Pass (University of Minnesota Pass) AFC data from March 07, 2016 to March 10, 2016. Figure 7.1 shows the number of trips made by the U-Pass holders during the analysis period. We can observe the morning peak between 6:30 A.M. to 9:30 P.M. and the afternoon peak between 3:00 P.M. to 6:30 P.M.

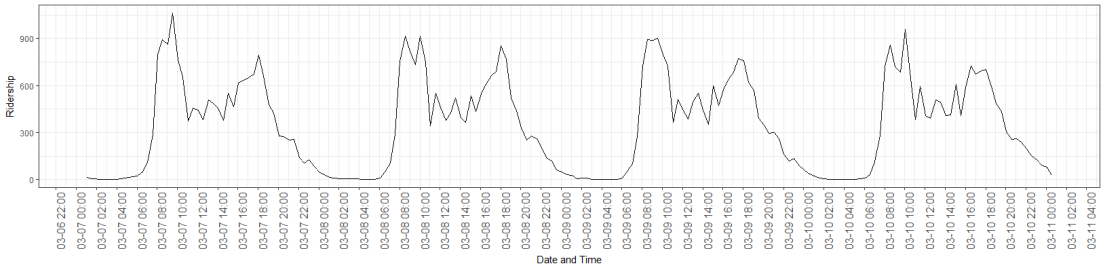


Figure 7.1: Time distribution of the trips in U-Pass data

Table 7.1: Comparison of the results between the baseline and the proposed method

Inference	**Baseline method**	**Proposed method**
Regular tags	46,507	51,919
Pay exit tags	0	4,504
Total	46,507	56,423

Total number of tags considered: 66,374 (60,812 Regular tags and 5,562 Pay exit tags)

After removing all the outliers described in §6.1, 66,374 out of 85,456 tags were left. Out of the remaining 66,374 tags, both origin and destination of 56,423 (85%) tags were successfully inferred in comparison to 46,507 (70%) tags being inferred using the baseline algorithm described in [10]. Table 7.1 summarizes the results in which about 81% of pay exit cases were inferred using the proposed algorithm in comparison to no inference using the baseline algorithm. Another comparison was done between the two algorithms for inferred boarding and alighting. Out of 46,507 inferred regular cases, 384 (0.8%) boardings and 300 (0.6%) alightings were different. About 9% of the tags were inferred as transfers in comparison to 17% in the original AFC data which considers every tag as a transfer if it is made within 2 hours and 30 minutes of the previous tag time. One point of interest is whether the last tag of the day can be inferred using the first tag of the day. We found that out of 26,275 last tags, the algorithm is able to infer the boarding and alighting of 21,110 tags (80%). This shows that this assumption works well in practice. Among the tags which are not inferred, about 59% are not inferred because no stop was found within walking distance from the current alighting location to the next boarding location. The likely reason for this non-inference is the use of another mode of transportation between two transit trips. We also observed that due to wrong selection of trip IDs from GTFS data, around 558 tags were not inferred using the baseline algorithm because the boarding time of the next tag was less than the alighting time of the current tag. The proposed algorithm eliminated this problem. This is because of the consideration of a list of possible trajectories for a given tag in the proposed algorithm in comparison to only one trip in the baseline algorithm.

The selection of the most likely trajectory based on the highest probability may result in accumulation of the inference error if there are multiple likely trajectories instead of a dominant one. In order to check for this possibility, we calculated the percentage

difference between the probabilities of the first and the second (if exists) most likely trajectories for every tag. The percentage difference is calculated with respect to the highest probability. A histogram of the percentage difference of these probabilities is shown in Figure 7.2. We found that more than 95% of the values were greater than 19% difference. To test if there exist a significant number of trips with multiple likely trajectories, we extracted 5% of the trips from lower tail of the distribution (shown by the dashed line) to compare the means of the probabilities of the first and the second most likely trajectories. We used the paired two sample T-test to compare the means.

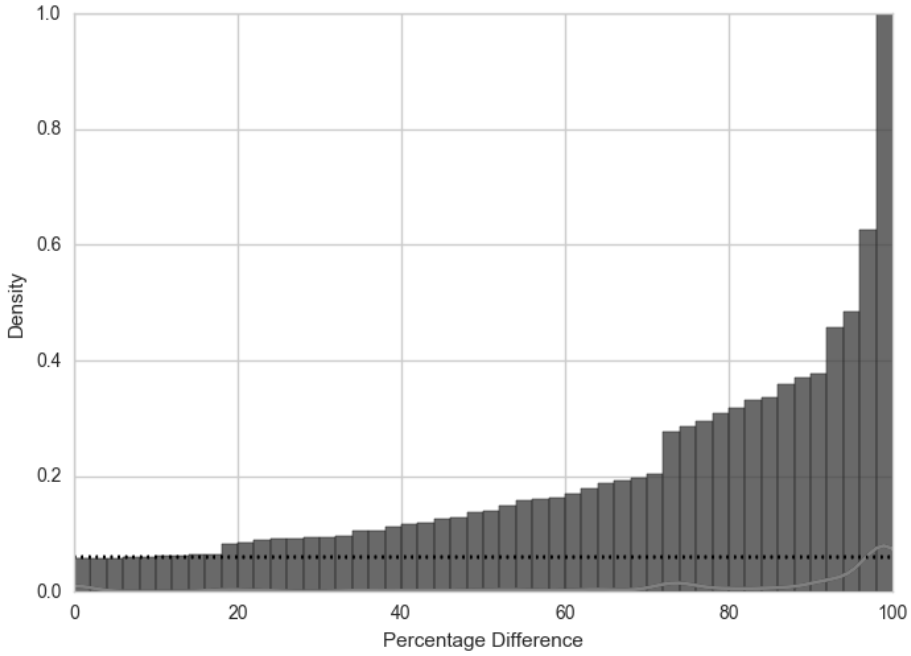


Figure 7.2: Distribution of the percentage difference between the probabilities of the first and the second (if exists) most likely trajectories

$$\begin{aligned} H_0 &: \mu_{\text{first}} = \mu_{\text{second}} \\ H_1 &: \mu_{\text{first}} \neq \mu_{\text{second}} \end{aligned} \tag{7.1}$$

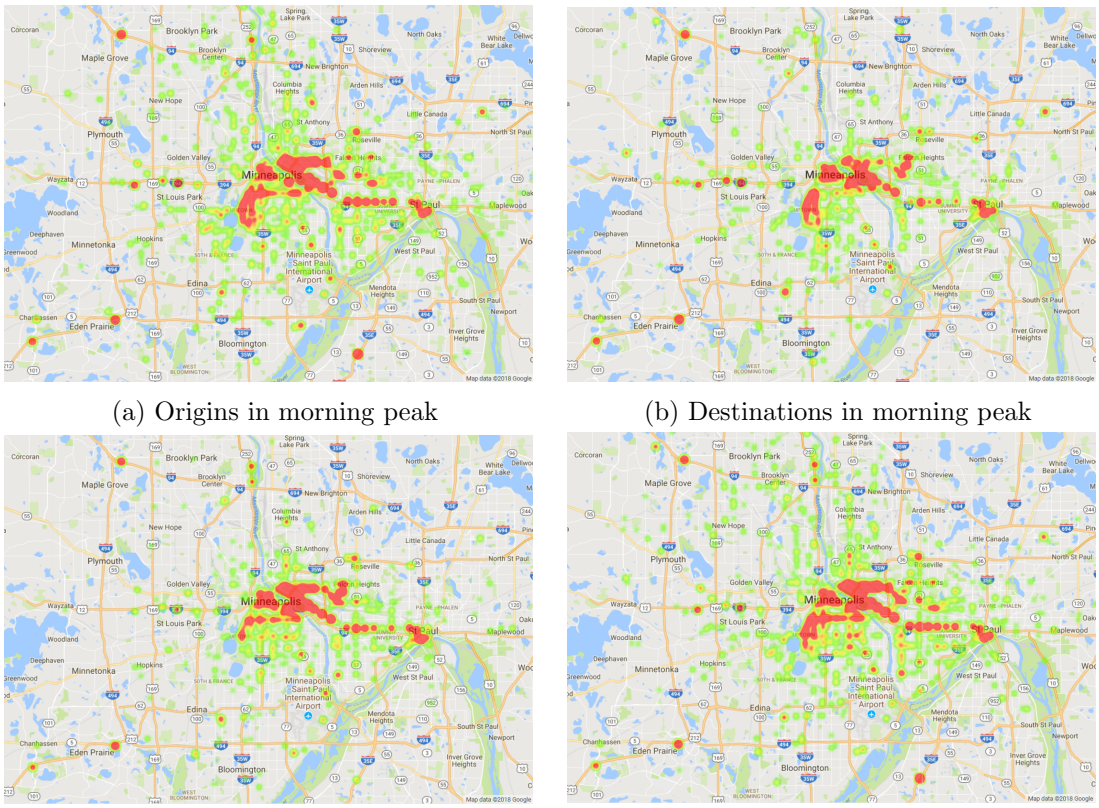
We found a T-statistic value of 24.383 which is greater than the critical value at 99% confidence level. This rejects the null hypothesis that the means of the probabilities of

the first and second most likely trajectories are equal. We recommend performing this test to check the quality of the results. If there exists a significant number of trips with multiple likely trajectories, then we either should consider all the likely trajectories for that tag or choose a trajectory randomly from the set of likely trajectories.

7.1.2 Applications using the inferred results

To summarize the outputs, heat maps of trip origins and destinations are prepared (Figure 7.3). The maps show that during morning peak hours, most of the trips originate from the areas east of the campus, Downtown and southwest Minneapolis, Downtown St. Paul, area around the university campus and Metro Green Line, while trip destinations are mainly at the university campus. Looking at the results for the evening peak hours, the origins and destinations look reversed, where most trips begin from the university campus and end at popular morning origin locations.

We compared the route ridership to assess the most common transit routes used by university students. Table 7.2 shows the high ridership routes and stops. In this table, as expected Metro Green Line has the highest ridership as it connects Downtown Minneapolis and Downtown St. Paul via university campus through two stations, East Bank Station and West Bank Station, which are also the popular locations for boarding and alighting in the stop table. Route 2 and route 3 are the most common bus routes used by the university students who live close to the campus. Route 3 connects Downtown Minneapolis and Downtown St. Paul via university by serving areas around the campus. Route 6, route 114 and route 113 serve the southwest suburbs while route 465 and 87 serve the southern suburbs. It is interesting to see that many students from the suburbs use the bus to commute to the campus. In the stop table (Table 7.2), stops located in the university campus such as East Bank Station, Pleasant Street & Jones Hall, West Bank Station, Washington Avenue & Coffman Union and Washington Avenue & Oak Street SE show high ridership. Other high ridership stops shown in the table are Metro Green Line stations. Finally, 15th Avenue SE and Como Avenue is also a popular stop for boarding and alighting served by route 3.



(a) Origins in morning peak

(b) Destinations in morning peak

(c) Origins in evening peak

(d) Destinations in evening peak

Figure 7.3: Intensity of trip origins and destinations. (For interpretation of colors in this figure, the reader is referred to the web version of this thesis.)

The highest number of tags was made on the Metro Green Line stations for which we did stop level origin-destination analysis. In Figure 7.4(a), we can observe that in the morning peak and eastbound direction, most trips start from Downtown Minneapolis at the western end of the line to the East Bank and West Bank Stations on the university campus or from Downtown St. Paul Union Depot (Figure 7.4(b)) at the eastern end of the line to the East Bank Station. Most of the students commute from the stations east of campus, for example, Stadium Village, Prospect Park and Westgate which are closer to the university. Conversely, during the evening peak, most trips go from East Bank and West Bank Stations to the popular origin locations in the morning (Figure 7.4(c) and Figure 7.4(d)).

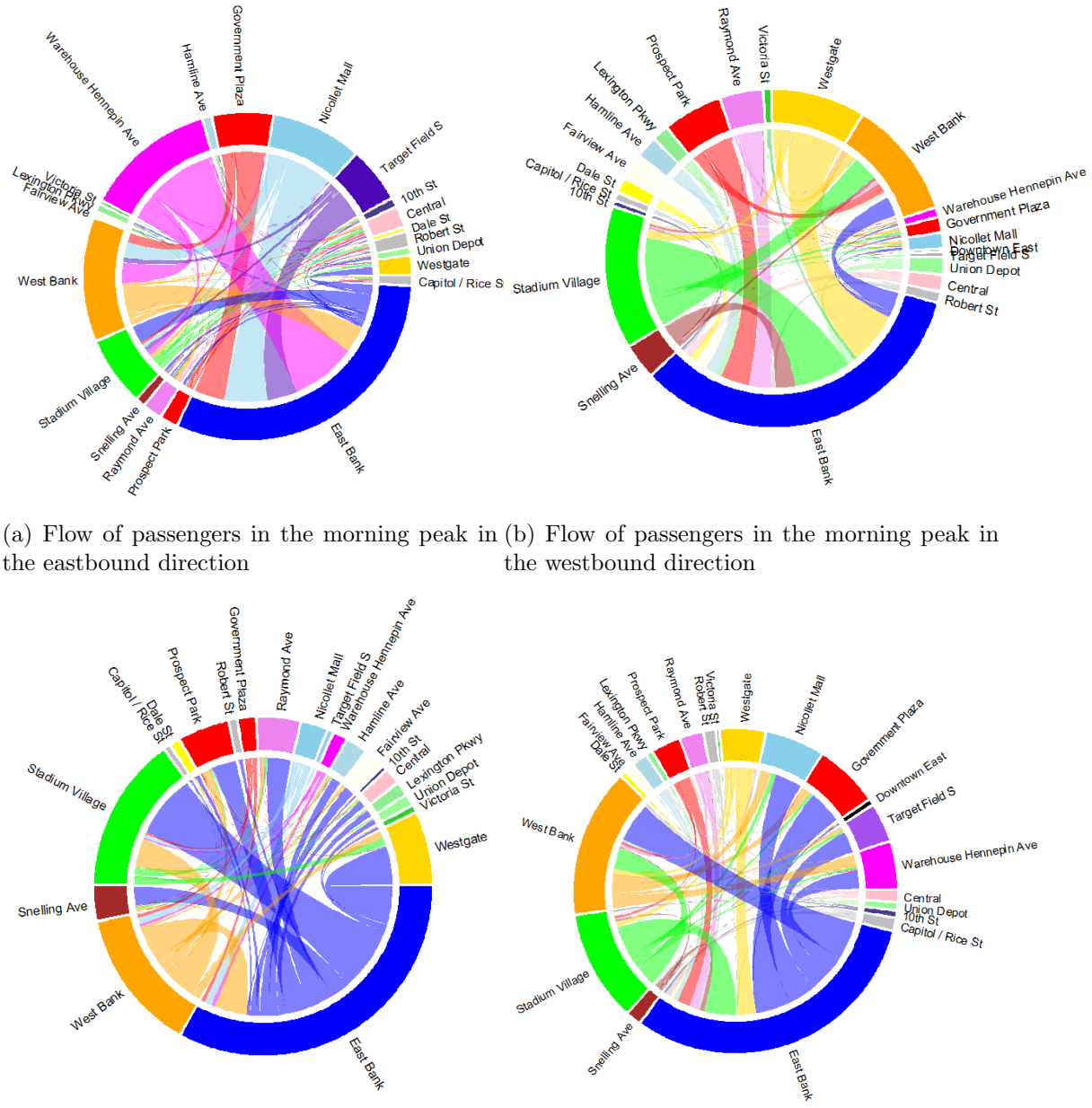
Table 7.2: Routes and stop locations with high ridership

Route	Ridership
Metro Green Line	22,144
3	12,213
2	6,340
6	3,014
465	2,274
114	1,569
113	1,207
901	1,126
87	1,073
698	794

Stop/Station	Boarding	Alighting
East Bank Station & Platform	7,052	7,314
Pleasant St & Jones Hall	3,423	3,265
Stadium Village Station & Platform	2,928	2,783
West Bank Station & Platform	2,924	2,723
Washington Ave & Coffman Union	1,637	2,006
Westgate Station & Platform	1,441	1,280
15th Ave SE & Como Ave SE	1,105	1,013
Washington Av & Oak St SE	971	971
Prospect Park Station & Platform	923	828
Warehouse Hennepin Ave Station & Platform	714	626

7.1.3 Discussion

In this subsection, we discuss the possible ways to infer the non-inferred tags. The proposed method infers the boarding and alighting of the tags made by the passenger during the day based on the assumptions given in §2.1. If these assumptions are not satisfied, then it cannot infer the boarding and alighting location of a given tag. Such trips (tags) are called unlinked trips [15]. The inference of such trips is possible using a method proposed by [15], which assumes that passengers tend to follow the same routine, and the historical alighting location and time information can be used to infer the alighting location of an unlinked trip. The method extracts the historical destinations for a passenger and tries to estimate the probability of alighting on these locations. The probability is found using spatial and temporal proximity of the historical alighting and the potential alighting. The method can be used in our case for the regular tags. We need to repeat the procedure of finding the spatial and the temporal probabilities for all the possible trajectories found for a given tag. However, the method may not be useful for pay-exit cases. For example, for a commuter who takes a regular route in the morning and pay-exit route in the evening, there will be no historical alighting and boarding location for the current and the next tag location respectively. Another disadvantage of combining the method proposed by [15] and the proposed method is heavy computational time as the spatial and temporal probabilities need to be calculated for each possible trajectory.



(a) Flow of passengers in the morning peak in the eastbound direction

(b) Flow of passengers in the morning peak in the westbound direction

(c) Flow of passengers in the evening peak in the eastbound direction

(d) Flow of passengers in the evening peak in the westbound direction

Figure 7.4: Passenger origin-destination flow on Metro Green Line light rail. (For interpretation of colors in this figure, the reader is referred to the web version of this thesis.)

Transit agencies require full O-D matrix for all the trips made by users given the errors and the missing information. This can be achieved using the boarding and alighting count data available from APC data. The O-D matrix obtained from AFC data using trip chaining algorithm can be used as a seed or prior matrix in optimization methods proposed by [34] or [36]. These optimization methods promise to perform better with a good quality seed matrix, which we can obtain from the trip chaining results. Another possibility is to proportionally assign the non-inferred boarding and alighting based on the APC data. Although these methods may not infer the correct boarding and alighting on an individual level, they will improve the results on an aggregate level.

7.2 Application of APC data

In this section, we present two numerical examples of OD estimation using the proposed methodology. First, simulation is used to assess the consistency and accuracy of the estimation method. Second, the OD estimation of a bus route in Twin Cities, MN is presented.

7.2.1 OD estimation using simulation

We prepare a synthetic OD matrix to set up a simulation environment. There can be different ways to simulating OD matrices for this experiment. In real APC data, there is likely to be a regular pattern of flow with some noise in it. However, to test the method in worst case scenarios, we simulate random matrices. To prepare such synthetic matrices, we make some assumptions on the probability distribution of arrival of the passengers on different stops. To facilitate the presentation of results, only 10 stops along a transit route are considered. The passenger arrival at the stop is assumed to follow a Poisson distribution.

$$b_i \sim Poisson(k) \ \forall i \in N \tag{7.2}$$

where, k is the mean arrival rate at the stop and b_i is the number of boarding at stop i. We recommend fitting a Poisson distribution to the real data to calculate the value of k. For example, the mean value of the arrival rate of the passengers on the A-line was

found to be equal to 0.86 during peak hours, which is quite low. To assess the significant errors produced by the estimation, the mean value equal to 15 passengers is assumed. Then the sparsity level is set for the O-D matrix. The sparsity level will make the value of the probability of flow from one stop to another stop zero if this probability value is less than the threshold sparsity level. This is done to create sparsity in the matrix and to test whether the method works more efficiently when the sparsity is high. Then the flow from one stop is assigned to others by assuming a multinomial distribution i.e.,

$$x_{ij} \sim MNL(b_i, p_{i1}, p_{i2}, ..., p_{i|N|}) \tag{7.3}$$

where, p_{ij} is the probability of movement from stop i to stop j. The diagonal and lower triangle elements of the matrix are set to zero because of the constraints (5.4)-(5.5). To calculate the boarding and alighting flows for O-D estimation, we sum the rows and columns of the simulated matrix. After that, an optimization model is set up using the Python API of CVX [51]. To avoid choosing the value of μ in optimization program (5.10), the program (5.8) is solved with l_1 norm. However, we recommend using the optimization program (5.10) when the sum of boarding and alighting count do not match in the APC data, which happens because of the errors in data collection. Figure 7.5 shows an example of recovered matrix using the proposed method. Using 200 Monte-Carlo samples of OD matrices, the root mean square error (RMSE) between the actual OD x and estimated OD x_{est} vector is calculated.

Actual OD

Stop #	1	2	3	4	5	6	7	8	9	10
1	0	3	0	3	0	0	2	2	2	0
2	0	0	12	0	0	0	0	0	3	0
3	0	0	0	0	1	9	0	4	0	0
4	0	0	0	0	1	0	4	0	1	3
5	0	0	0	0	0	6	1	1	0	1
6	0	0	0	0	0	0	4	6	4	1
7	0	0	0	0	0	0	0	0	4	3
8	0	0	0	0	0	0	0	0	2	0
9	0	0	0	0	0	0	0	0	0	2
10	0	0	0	0	0	0	0	0	0	0

Estimated OD

Stop #	1	2	3	4	5	6	7	8	9	10
1	0	3	0	3	2	0	0	4	0	0
2	0	0	12	0	0	1	2	0	0	0
3	0	0	0	0	0	14	0	0	0	0
4	0	0	0	0	0	0	0	0	9	0
5	0	0	0	0	0	0	0	1	0	8
6	0	0	0	0	0	0	9	1	5	0
7	0	0	0	0	0	0	0	7	0	0
8	0	0	0	0	0	0	0	0	2	0
9	0	0	0	0	0	0	0	0	0	2
10	0	0	0	0	0	0	0	0	0	0

Figure 7.5: An illustration of actual and recovered matrix

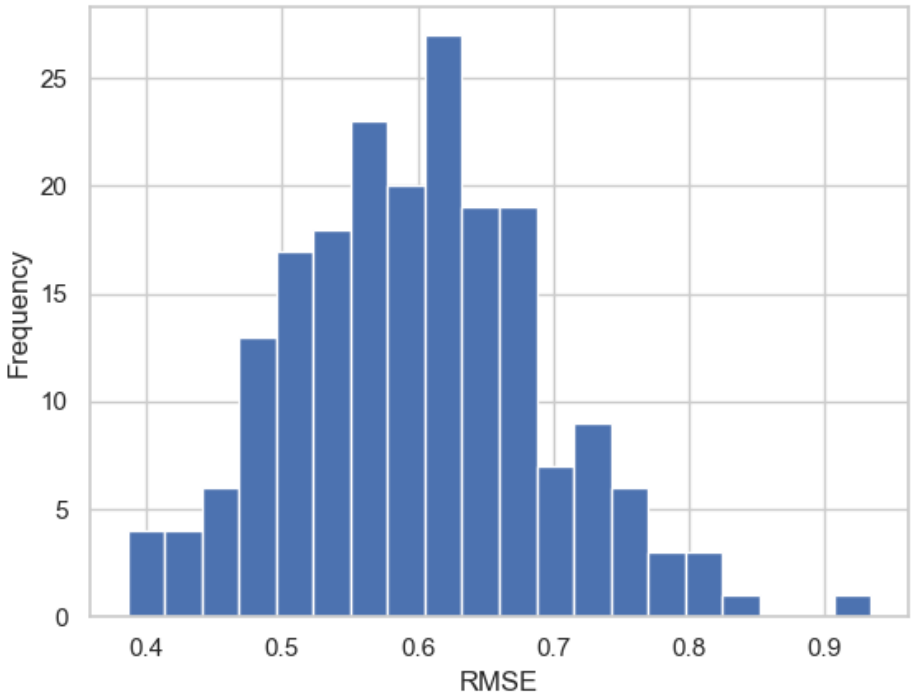


Figure 7.6: l_2 error between the actual and estimated OD matrix

Figure 7.6 shows the histogram of RMSE in the estimation for each sample. We can observe that the mean value of the error is 0.59 and with a standard deviation of 0.09. The 95% confidence interval of the l_2 error was found to be equal to (0.585, 0.611). This shows that the results obtained from this estimation method are consistent and small. To see how the method performed in predicting the individual origin-destination pair flow value, we created a box plot for the estimation error (Figure 7.7). The proposed method predicted the actual value of the non-zero entries 41.5% of the time. In case of errors, the method seems to overpredict the values except some of the O-D pairs such as 0-4, 1-2 and 5-6.

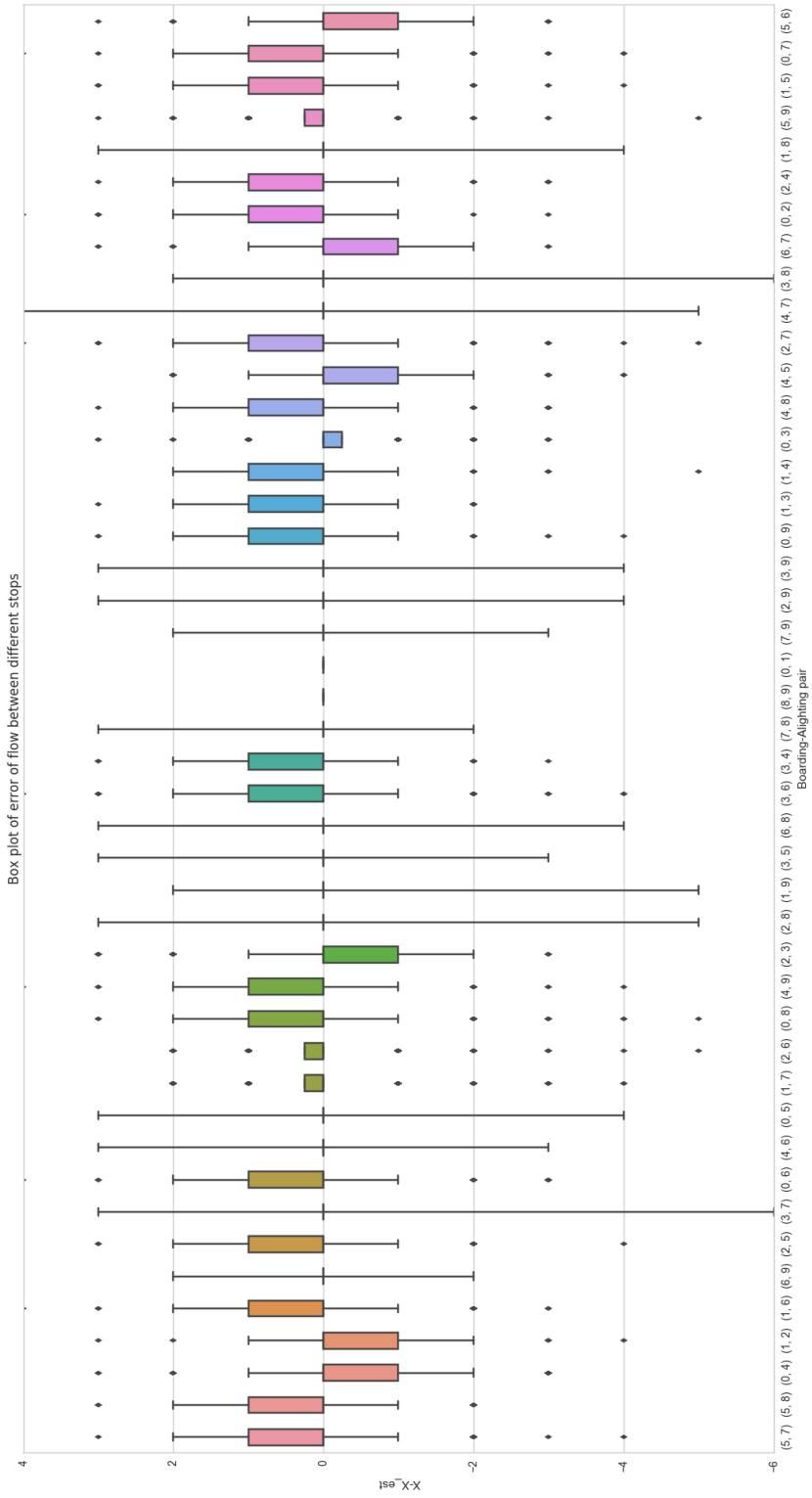


Figure 7.7: Box plot for the errors in estimation of O-D flows

Figure 7.8(a) shows the average load profile of the passengers on the transit route. The width of the 95% confidence interval is small which shows that the method is reliable in estimating demand and therefore in deciding the adequate frequency to handle the load of the passengers. We can also observe that the errors in estimating the load of the passengers is also quite small (Figure 7.8(b)).

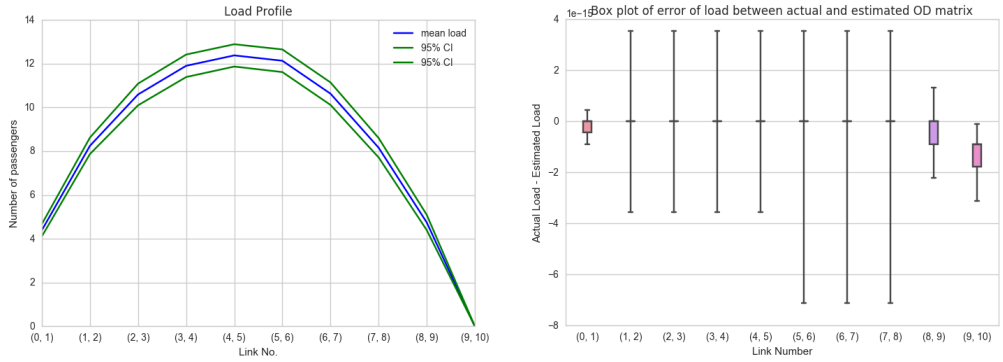


Figure 7.8: (a) Average load profile of the transit route. (b) Box plot of error between actual and estimated load

To understand the effect of sparsity, we solved the problem for several levels of sparsity and calculated the root mean square error (RMSE) between the estimated and actual OD matrix. Figure 7.9 shows the RMSE value with respect to the sparsity in the matrix. We can observe that the RMSE value is reduced with increased sparsity. For example, when the OD matrix has only 10% non-zero values, the corresponding RMSE value was found to be less than 0.35, which is quite impressive. This shows that the accuracy of the method is improved when there is more sparsity. Comparing the results to common least squares solution (Figure 7.9), the proposed method is able to recover solutions with lower RMSE value. It can also be observed that when the sparsity is low, the proposed method is more efficient than least squares as the gap between two lines is high but when there are a greater number of non-zero entries, the RMSE gap between these two methods reduces.

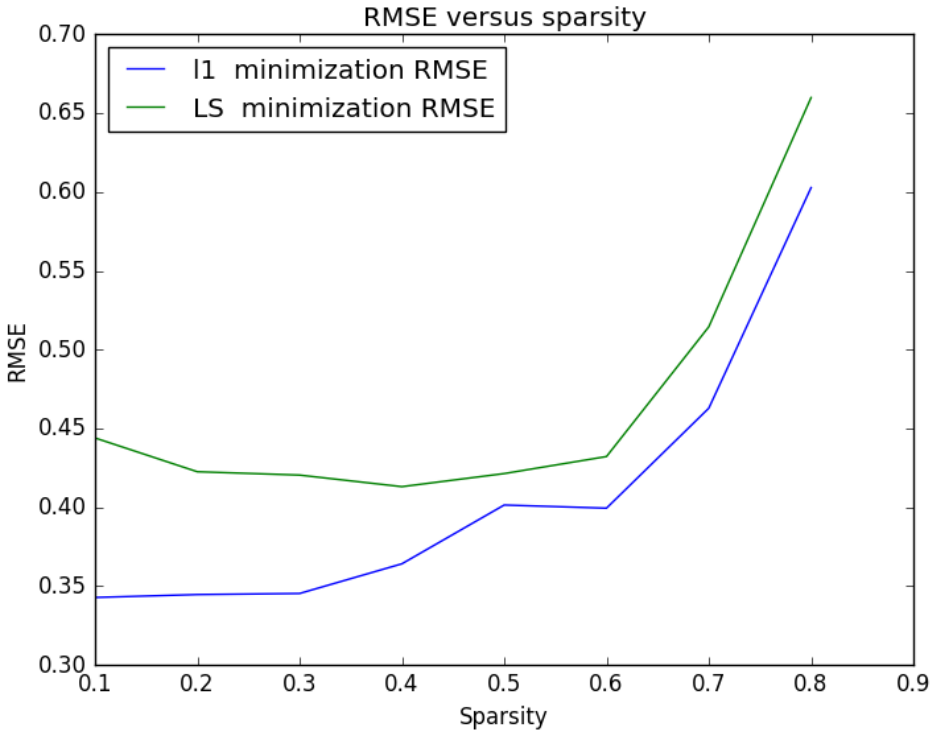


Figure 7.9: Root mean square error (RMSE) versus sparsity in OD estimation (Sparsity is in terms of proportion of non-zero values)

To see how different demand patterns affect the OD estimation, a similar simulation for several mean arrival rates (k) of passengers at stops is performed. Figure 7.10 shows normalized RMSE values with respect to sparsity in the random matrix for different mean arrival rate. The normalization is done by simply dividing RMSE by mean arrival rate. The results are presented in separate panels. We can see that the normalized RMSE decreases with an increase in demand. At $k = 20$, the normalized RMSE value was found to be almost equal to 1, which is still quite low. At lower demand, the matrix is already sparse, so we see less effect of sparsity parameter.

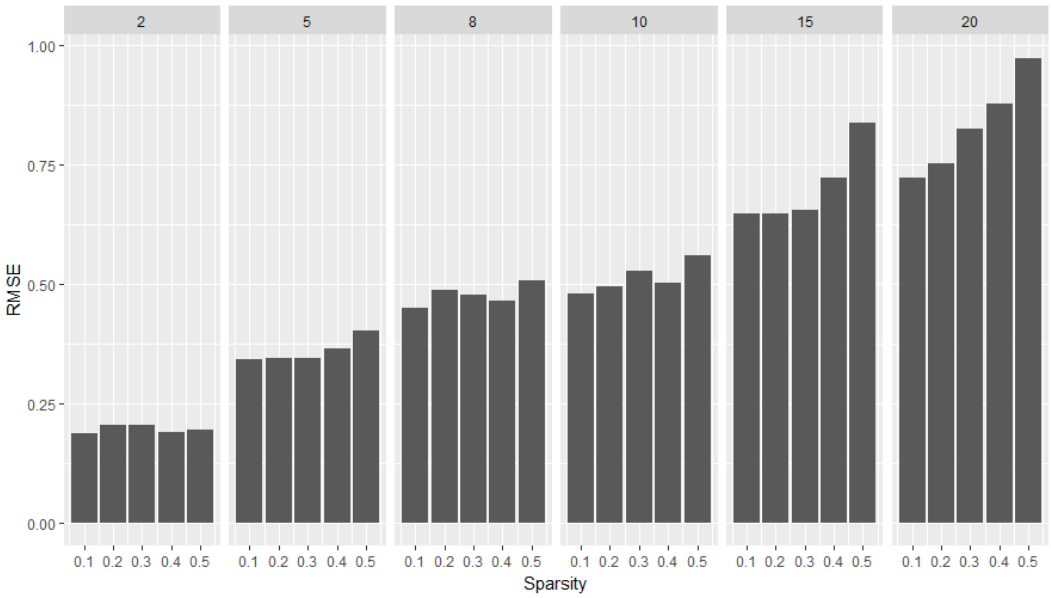


Figure 7.10: Comparing RMSE with sparsity for different mean arrival rates (each panel represents different mean arrival rate of passengers)

7.2.2 OD estimation of A Line BRT route in Twin Cities

We select A Line, which is a bus rapid transit (BRT) route in Twin Cities for this analysis. It serves 20 stations along Snelling Av and 46th St. We select a trip from the data during peak hour. The number of boarding and alighting at different stops in the northbound direction is shown in Figure 7.11. We can observe the popular boarding locations such as 46th street station, 46th & Minnehaha station, and Snelling & Highland station and alighting stops such as Rosedale transit center, Snelling & Highland station and Snelling & Clair st. station. The optimization program (5.10) is used to solve the given problem with a value of $\mu = 0.2$. A few recommendations for choosing the value of μ is given in [39].

The total ridership of the trip is 16. Because of low ridership, flow along most of the O-D pairs should be equal to zero. We apply the proposed method to the given data and calculate the origin-destination flows. Figure 7.12 shows the origin-destination flows between different O-D pairs. We can see that the flow occurred only between 11 O-D pairs out of 400 pairs (2.75%). The highest flow was observed between Snelling & Highland Av and Rosedale Transit Center, which is the last station along this route.

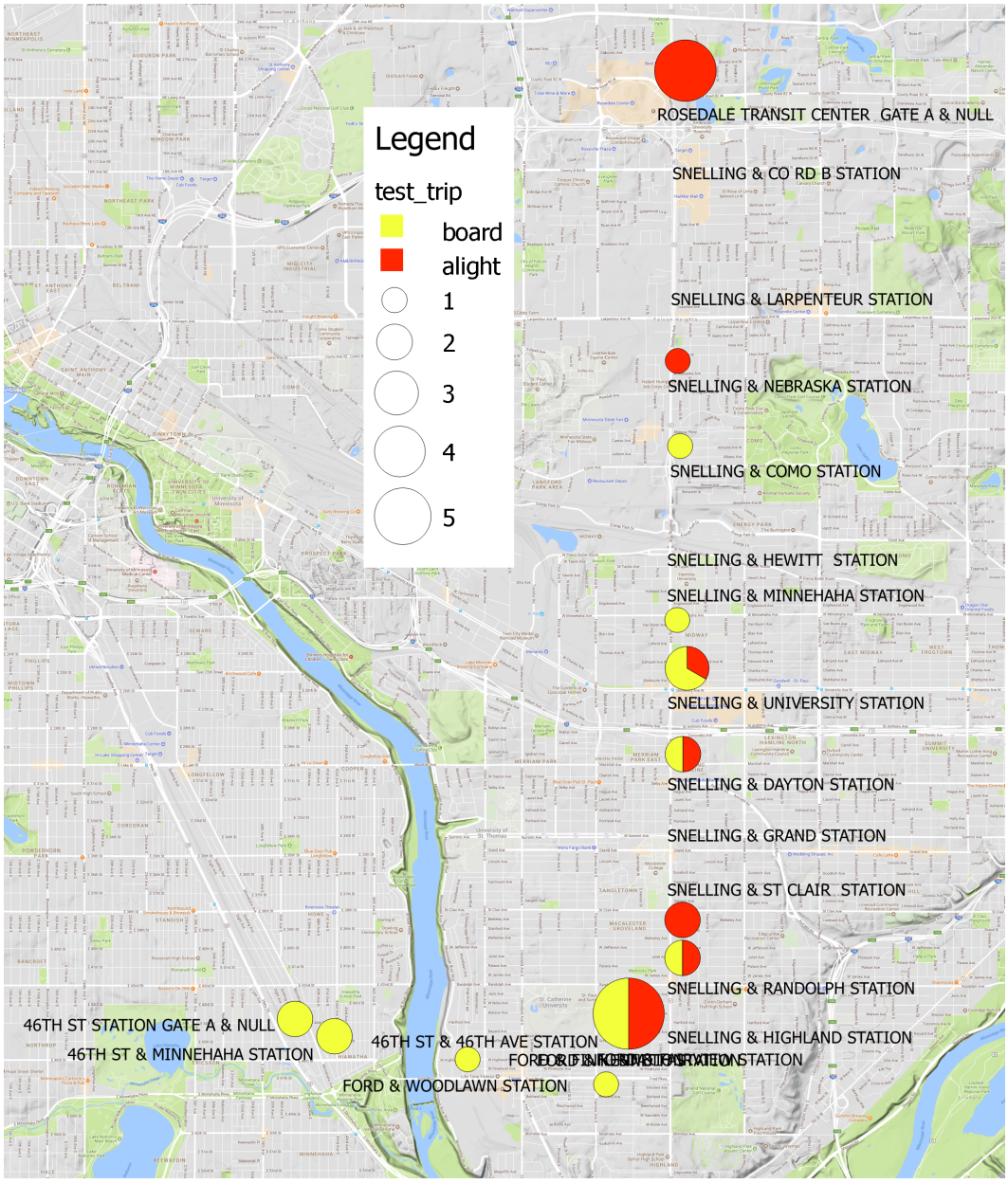


Figure 7.11: Boarding and alighting counts of A Line

Other popular OD pairs are 46th St and Snelling & St. Clair, Snelling & Minnehaha and Snelling & Highland Av. Because of the low ridership, the sparse matrix recovery seems to perform well.

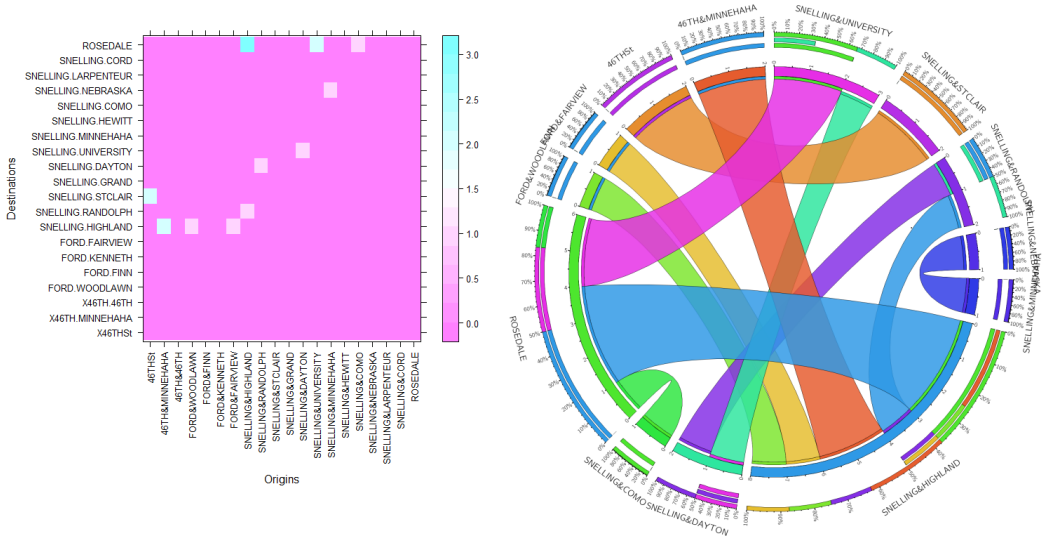


Figure 7.12: Origin-Destination flow for A Line, Twin Cities, MN

Chapter 8

Conclusions and Recommendations

The methods presented in this thesis have shown to infer aspects of passenger trajectories which are not directly observable through raw data. This chapter summarizes the results and findings of these methods and assesses the extent to which the objectives described in §1.2 were satisfied. Various recommendations for future research are then proposed including the improvement in current methodology and potential applications of results obtained.

8.1 Results and Findings

This thesis presents the application of transit automated data to estimate OD matrix at different aggregation levels. A robust method for trip chaining of AFC data was presented to infer origin and destination of transactions, which tries to relax various assumptions on the parameters used in the existing trip chaining algorithms. The parameters can vary according to the quality of data and user behavior in different transit systems, so a fixed value cannot be assumed for different transit systems. This is evident from trip chaining results for the Twin Cities' AFC data. The proposed method provides the flexibility to assume a higher value for these parameters to avoid the wrong inference of origin and destination. The method uses probability distributions for potential boarding stop location, bus delay and passenger's route choice behavior. By combining

these probabilities, it infers the most likely trajectory of the passenger. Though being an open transit system with pay-exit buses and sub-routes, these attributes create various problems for trip chaining. Using the proposed method, various problems such as erroneous GPS locations, selection of the wrong trip for inference, and pay-exit cases are addressed. The proposed algorithm was also suitably modified to deal with different pay exit cases. The O-D matrix results can be used in multiple ways to understand the travel behavior of passengers in a transit system. We presented the ridership analysis on an aggregate level for the Twin Cities and also the route level analysis for a light rail transit line. The trip chaining results can also be used for creating clusters of customers to evaluate similar travel patterns based on their regularity in using the transit system. These results can inform planners for better decisions to improve transit services.

The thesis also proposed a method for estimating an origin-destination OD matrix for a transit route along one direction. The problem was formulated as an undetermined system of linear equations. The adopted strategy was to estimate a sparse O-D matrix, using l_0 norm. Using its convex surrogate l_1 regularizer, the problem can be solved efficiently. The sparsity in the matrix is generated because there are only a few popular O-D pairs along a transit route where the flow occurs. The constraints and sparsity try to force the solution to an actual value. We tested the efficiency of the estimator using simulation. The errors were found to be bound within a small range. With an increased level of sparsity in the matrix, the method was able to recover more accurate results. We also found small errors even for higher demand. For example, the normalized RMSE between estimated and actual matrix value was found to be at most 0.1. It was observed that the proposed method works efficiently by showing a numerical example of A-line BRT route in Twin Cities, MN.

8.2 Recommendations for future research

Current research on trip chaining can be expanded in multiple directions. The case where the current tag is regular and the next tag is pay-exit and both tags use the same route is analyzed using a method of proportions. Additional information from other

data sources can help in the development of a suitable algorithm for this case. Furthermore, the results obtained from trip chaining can also be used for other research such as trip purpose inference, analyzing spatial and temporal travel pattern, route choice behavior analysis of passengers and transit assignment models.

The use of compressed sensing to solve the underdetermined system of equations is new to the transportation field. The research described in chapter 5 can also be expanded in multiple directions. The method can be used to estimate a full transit network OD matrix. The problem can be formulated as a bi-level program with sparse recovery optimization at the upper level and transit assignment at the lower level to capture route choice behavior in the model. We believe that the network level OD will also be sparse because it is unlikely that passengers boarding at one stop can alight at all other stops in the network. The concept can also be extended to matrix sensing which will be helpful in estimating a time-dependent transit OD matrix. As the boardings and alightings follow a regular pattern during various hours of the day, data from several days can be used to learn this pattern. This means the high dimensional data for several days can be used to minimize the rank of the matrix to extract a regular pattern. This can be done by minimizing the nuclear norm of the matrix, which is a convex surrogate for the rank of the matrix. The problem is computationally challenging and needs further attention. Further studies are required to show under which constraints, the OD linear map satisfy the RIP property. Other statistical methods are also required to assess the accuracy of the estimation.

References

[1] N.H.M. Attanucci, J. & Wilson. Bus Transit Monitoring Manual: Volume 1: Data Collection Program Design. *US Department of Transportation*, 1(August), 1981.

[2] Marie Pier Pelletier, Martin Trépanier, and Catherine Morency. Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies*, 19(4):557–568, 2011.

[3] James J. Barry, Robert Newhouser, Adam Rahbee, and Shermeen Sayeda. Origin and Destination Estimation in New York City with Automated Fare System Data. *Transportation Research Record: Journal of the Transportation Research Board*, 1817(-1):183–187, 2007.

[4] Martin Trépanier, Nicolas Tranchant, and Robert Chapleau. Individual Trip Destination Estimation in a Transit Smart Card Automated Fare Collection System. *Journal of Intelligent Transportation Systems*, 11(1):1–14, 2007.

[5] J Zhao, a Rahbee, and N.H.M Wilson. Estimating a rail passenger trip origin-destination using automatic data collection systems. *Computer-Aided Civil and Infrastructure Engineering*, 22(5):376–387, 2007.

[6] Ka Alfred Chu and Robert Chapleau. Enriching Archived Smart Card Transaction Data for Transit Demand Modeling. *Transportation Research Record: Journal of the Transportation Research Board*, 2063:63–72, 2008.

[7] Janine Farzin. Constructing an Automated Bus Origin-Destination Matrix Using Farecard and Global Positioning System Data in São Paulo, Brazil. *Transportation*

Research Record: Journal of the Transportation Research Board, 2072(2072):30–37, 2008.

[8] James Barry, Robert Freimer, and Howard Slavin. Use of Entry-Only Automatic Fare Collection Data to Estimate Linked Transit Trips in New York City. *Transportation Research Record: Journal of the Transportation Research Board*, 2112:53–61, 2009.

[9] Ka Chu and Robert Chapleau. Augmenting Transit Trip Characterization and Travel Behavior Comprehension. *Transportation Research Record: Journal of the Transportation Research Board*, 2183:29–40, 2010.

[10] Neema Nassir, Alireza Khani, Sang Lee, Hyunsoo Noh, and Mark Hickman. Transit Stop-Level Origin-Destination Estimation Through Use of Transit Schedule and Automated Data Collection System. *Transportation Research Record: Journal of the Transportation Research Board*, 2263:140–150, 2011.

[11] Wei Wang, John P Attanucci, and Nigel H M Wilson. Bus Passenger Origin-Destination Estimation and Related Analyses Using Automated Data Collection Systems. *Journal of Public Transportation*, 14(4):131–150, 2011.

[12] Xiao-lei Ma, Yin-hai Wang, Feng Chen, and Jian-feng Liu. Transit smart card data mining for passenger origin information extraction. *Journal of Zhejiang University SCIENCE C*, 13(10):750–760, 2012.

[13] Marcela A. Munizaga and Carolina Palma. Estimation of a disaggregate multimodal public transport Origin-Destination matrix from passive smartcard data from Santiago, Chile. *Transportation Research Part C: Emerging Technologies*, 24:9–18, 2012.

[14] Jason Gordon, Harilaos Koutsopoulos, Nigel Wilson, and John Attanucci. Automated Inference of Linked Transit Journeys in London Using Fare-Transaction and Vehicle Location Data. *Transportation Research Record: Journal of the Transportation Research Board*, 2343:17–24, 2013.

[15] Li He and Martin Trépanier. Estimating the Destination of Unlinked Trips in Transit Smart Card Fare Data. *Transportation Research Record: Journal of the Transportation Research Board*, 2535:97–104, 2015.

[16] Azalden Alsger, Behrang Assemi, Mahmoud Mesbah, and Luis Ferreira. Validating and improving public transport origin-destination estimation algorithm using smart card fare data. *Transportation Research Part C: Emerging Technologies*, 68:490–506, 2016.

[17] Aditya Krishna Menon, Chen Cai, Weihong Wang, Tao Wen, and Fang Chen. Fine-grained OD estimation with automated zoning and sparsity regularisation. *Transportation Research Part B: Methodological*, 80:150–172, 2015.

[18] E.J. Candes and M.B. Wakin. An Introduction To Compressive Sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008, arXiv:1307.1360v1.

[19] Tian Li, Dazhi Sun, Peng Jing, and Kaixi Yang. Smart Card Data Mining of Public Transport Destination: A Literature Review. *Information*, 9(1):18, 2018.

[20] DS Navick and PG Furth. Estimating passenger miles, origin-destination patterns, and loads with location-stamped farebox data. *Transportation Research Record: Journal of ...*, 107-113(02):2466, 2002.

[21] Catherine Seaborn, John Attanucci, and Nigel H M Wilson. Using Smart Card Fare Payment Data To Analyze Multi- Modal Public Transport Journeys in London. *Transportation Research Record: Journal of the Transportation Research Board*, 2121.-1:55–62, 2009.

[22] Google. General Transit Feed Specification, 2005.

[23] Jason B. Gordon, Haris N. Koutsopoulos, and Nigel H.M. Wilson. Estimation of population origin–interchange–destination flows on multimodal transit networks. *Transportation Research Part C: Emerging Technologies*, 90(January):350–365, 2018.

[24] P. Kumar, A. Khani, and Q. He. A robust method for estimating transit passenger trajectories using automated data. *Transportation Research Part C: Emerging Technologies*, 95, 2018.

[25] Steve Robinson, Baskaran Narayanan, Nelson Toh, and Francisco Pereira. Methods for pre-processing smartcard data to improve data quality. *Transportation Research Part C: Emerging Technologies*, 49:43–58, 2014.

[26] Kenneth Perrine, Alireza Khani, and Natalia Ruiz-Juri. Map-Matching Algorithm for Applications in Multimodal Transportation Network Modeling. *Transportation Research Record: Journal of the Transportation Research Board*, 2537(2537):62–70, 2015.

[27] Moshe E Ben-akiva. Alternative Methods to Estimate Route-Level Trip Tables and Expand On-Board Surveys. *Transportation Research Record 1037, TRB, National Research Council, Washington, D.C.,*, pages pp. 1–11.

[28] Ennio Cascetta and Sang Nguyen. A unified framework for estimating or updating origin/destination matrices from traffic counts. *Transportation Research Part B*, 22(6):437–455, 1988.

[29] Rabi Mishalani, Yuxiong Ji, and Mark McCord. Effect of Onboard Survey Sample Size on Estimation of Transit Bus Route Passenger Origin-Destination Flow Matrix Using Automatic Passenger Counter Data. *Transportation Research Record: Journal of the Transportation Research Board*, 2246:64–73, 2011.

[30] Gary A. Davis. Estimating Freeway Demand Patterns and Impact of Uncertainty on Ramp Controls. *Journal of Transportation Engineering*, 119(4):489–503, 2006.

[31] M J Maher. Inferences on trip matrices from observations on link volumes: a Bayesian statistical approach. *Transportation Research Part B: Methodological*, 17(6):435–447, 1983.

[32] Baibing Li. Markov models for Bayesian analysis about transit route origin-destination matrices. *Transportation Research Part B: Methodological*, 43(3):301–310, 2009.

[33] Martin L. Hazelton. Statistical inference for transit system origin-destination matrices. *Technometrics*, 52(2):221–230, 2010.

[34] Henk J. Van Zuylen and Luis G. Willumsen. The most likely trip matrix estimated from traffic counts. *Transportation Research Part B: Methodological*, 14(3):281–293, 1980.

[35] Nancy L Nihan and Gary A Davis. Application of Prediction-Error Minimization and Maximum Likelihood to Estimate Intersection O-D Matrices from Traffic Counts., 1989.

[36] Heinz Spiess. A Maximum Likelihood Model For Estimating Origin-Destination Matrices. *Transportation Research Board*, 21B(5):395–412, 1987.

[37] Y Vardi. Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data. *Journal of the American Statistical Association*, 91(433):365–377, 1996.

[38] Pramesh Kumar, Alireza Khani, and Gary A. Davis. Transit Route Origin–Destination Matrix Estimation using Compressed Sensing. *Transportation Research Record: Journal of the Transportation Research Board*, page 036119811984589, 2019.

[39] Robert Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[40] Metro Transit. About Metro Transit, 2019.

[41] Jing Quan Li. Match bus stops to a digital road network by the shortest path model. *Transportation Research Part C: Emerging Technologies*, 22:119–131, 2012.

[42] Paul Newson Krumm and John. Hidden-Markov-Map-Matching-Through-Noise-and-Sparseness-ACM-SIGSPATIAL-2009-final, 2009.

[43] Royal Navy. The Principles of Navigation: The Admiralty Manual of Navigation Volume 1, 2008.

[44] Frank van Diggelen. GNSS accuracy: Lies, damn lies, and statistics. *GPS World*, 18(1):26–32, 2007.

[45] Borhan M. Sanandaji and Pravin P. Varaiya. Compressive Origin-Destination Matrix Estimation. pages 2–9, 2014, 1404.3263.

[46] Sanjay Chawla, Yu Zheng, and Jiafeng Hu. Inferring the Root Cause in Road Traffic Anomalies. 2012.

[47] Yin Zhang, Zihui Ge, Albert Greenberg, and Matthew Roughan. Network anomography. page 1, 2008.

[48] Emmanuel Candès and Terence Tao. Decoding by Linear ProgrammingEmmanuel Candes†. *IEEE Trans. Inf. Theory*, 51(12):4203–4215, 2005.

[49] Trevor Park and George Casella. The Bayesian Lasso. 103(482):681–686, 2008.

[50] E.J. Candes and M.B. Wakin. An Introduction To Compressive Sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

[51] Michael Grant and Stephen Boyd. CVX: Matlab Software for Disciplined Convex Programming, version 2.1, 2014.

[52] M Lapsley and B D Ripley. RODBC: ODBC database access, 2005.

[53] John Douclas Hunt. A Logit Model of Public Transport Route Choice. *ITE Journal*, December(December):26–30, 1990.

[54] Zhan Guo and Nigel Wilson. Modeling Effects of Transit System Transfers on Travel Behavior: Case of Commuter Rail and Subway in Downtown Boston, Massachusetts. *Transportation Research Record: Journal of the Transportation Research Board*, 2006:11–20, 2007.

[55] Sebastián Raveau, Zhan Guo, Juan Carlos Muñoz, and Nigel H M Wilson. Route Choice Modelling on Metro Networks. *Conference on Advanced Systems for Public Transport*, (56 2):1–13, 2012.

[56] R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, 2017.

Appendix A

Notations

Table A.1: Notations used in this thesis

Variable	Definition
n	Index/row number in the AFC data
t	Time of the tag
r	Bus route number of the tag
δ	Direction of bus route
θ	Geographical coordinates of the tag
$\mathcal{GC}$	Great circle distance
α	Buffer distance for finding possible boarding stops
ϵ	Buffer distance for finding possible alighting stops
τ	Buffer time for finding possible trips
k	Index for different boarding stops
l	Index for different trips
m	Index for different alighting stops
S_n	List of possible boarding stops for tag n
$\mathcal{T}_{nk}$	List of possible trips for tag n and boarding stop k
Δ_{kl}	Absolute difference between tag time t_n and trip time $t_{tr_{kl}}$
A_{nkl}	List of possible alighting stops for tag n, boarding stop k and trip l

Continued on next page

Variable	Definition
$\mathcal{IV}_{klm}$	In-vehicle travel time for trip l with boarding stop k and alighting stop m
w_{klm}	Walking distance from alighting stop m for trip l with boarding stop k to the next tag location θ_{n+1}
$\|M\|_1$	l_1 norm of matrix M, $\|M\|_1 = \sum_{ij} \|M_{ij}\|$
$\|M\|_\infty$	l_∞ norm of matrix M, $\|M\|_\infty = \max_{i,j} \|M_{ij}\|$
N	Noise matrix
L_f	Lipshitz constant
$\langle , \rangle$	Inner product
N	Set of stops/stations along a transit route
i	Index for transit stop
b_i	Number of passengers boarding at stop i
a_i	Number of passengers alighting at stop i
X	Origin destination flow matrix
x	Vector form of OD matrix X
$\mathcal{A}$	A Linear map on vector x