Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Coordenação de Pós-Graduação em Ciência da Computação

Inferring Passenger-Level Bus Trip Traces from Schedule, Positioning and Ticketing Data: Methods and Applications

Tarciso Braz de Oliveira Filho

Thesis submitted to Coordenação de Pós-Graduação em Ciência da Computação in partial fulfillment of the requirements for the degree of Master of Computer Science at Universidade Federal de Campina Grande.

Area of Concentration: Computer Science

Line of Research: Intelligent Transportation Systems

Nazareno Ferreira Andrade

(Advisor)

Campina Grande, Paraíba, Brazil

O48i Oliveira Filho, Tarciso Braz de.

Inferring passenger-level bus trip traces from schedule, positioning and ticketing data: methods and applications / Tarciso Braz de Oliveira Filho. – Campina Grande, 2019.

55 f. : il. color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro Engenharia Elétrica e Informática, 2019.

"Orientação: Prof. Dr. Nazareno Ferreira Andrade".

Referências.

1. Intelligent Transportation Systems. 2. Public Transportation. 3. Automatic Fare Collection. 4. Automatic Vehicle Location. 5. Origin-Destination Matrix. 6. Transit Usage Performance Evaluation. I. Andrade, Nazareno Ferreira. II. Título.

CDU 004.891.3:004.62:656.121(043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA SEVERINA SUELI DA SILVA OLIVEIRA CRB-15/225

"INFERRING PASSENGER-LEVEL BUS TRIP TRACES FROM SCHEDULE, POSITIONING AND TICKETING DATA: METHODS AND APPLICATIONS"

TARCISO BRAZ DE OLIVEIRA FILHO

DISSERTAÇÃO APROVADA EM 27/02/2019

NAZARENO FERREIRA DE ANDRADE, Dr., UFCG
Orientador(a)

CLÁUDIO ELÍZIO CALAZANS CAMPELO, PhD., UFCG
Examinador(a)

KEIKO VERÔNICA ONO FONSECA, Dra., UTFPR
Examinador(a)

CAMPINA GRANDE - PB

Abstract

As a result of the recent and fast rise in urban population, mobility has emerged as one of the most problematic and fast-evolving urban problems of the 21st century. With the advent of the Internet of Things, gigabytes of data are generated every day by Public Transportation Systems around the world, including bus GPS/speed records, and passenger boarding registries. Although this data has the potential to help improve mobility, the vast amount, dynamicity and diversity of data produced by different systems with different goals and constraints poses difficulties to integrate and analyze it and help the system's users, operators and administrators. This study addresses this problem, more specifically the one of using bus schedule data, raw GPS and smart card records to reconstruct trips at passenger-level. We use data from the Curitiba bus system in Brazil to devise an analysis pipeline that combines and extends consolidated heuristics found in literature. Experiments demonstrate the utility of the proposed solution in two applications scenarios: a) the estimation of an Origin-Destination Matrix for Public Transport users, which was validated by a comparison to a recent Origin-Destination Survey performed in the city; and b) an analysis of the (in)efficiency of passenger itinerary choice, conducted by contrasting the estimated itinerary choice (extracted from trip reconstruction) to the set of available and feasible itineraries at the time of boarding.

Keywords: Intelligent Transportation Systems, Public Transportation, Automatic Fare Collection, Automatic Vehicle Location, GTFS, Origin-Destination Matrix, Transit Usage Performance Evaluation

Resumo

Como resultado do recente e rápido crescimento da população urbana, a mobilidade tem emergido como um dos problemas urbanos mais complexos e de rápida evolução no século XXI. Com o advento da Internet das Coisas, *gigabytes* de dados são gerados diariamente por Sistemas de Transporte Público ao redor do mundo, incluindo registros de GPS e velocidade dos ônibus, além de registros de embarque de passageiros. A despeito desses dados possuírem o potencial de auxiliar na melhoria da mobilidade, a enorme quantidade, dinamicidade e diversidade de dados produzidos por diferentes sistemas com diferentes objetivos e restrições, impõe dificuldades para a integração e análise do mesmo com o fim de ajudar os usuários, operadores e administradores do sistema. Esse estudo aborda esse problema, mais especificamente o de utilizar dados de programação dos ônibus, dados brutos de GPS e dados de cartão de embarque para reconstruir viagens de ônibus a nível de passageiro. São utilizados dados do sistema de ônibus de Curitiba no Brasil para conceber um processo de análise que combine e extenda heurísticas consolidadas encontradas na literatura. Experimentos demonstram a utilidade da solução proposta em dois cenários de aplicação: a) a estimação de uma Matriz de Origem-Destino para usuários de Transporte Público, que foi validada através de uma comparação com uma Pesquisa Origem-Destino realizada recentemente na cidade; e b) uma análise da (in)eficiência da escolha de itinerário do passageiro, realizada contrastando o itinerário escolhido estimado (extraído da reconstrução da viagem) com o conjunto de itinerários disponíveis e viáveis no momento do embarque.

Palavras-Chave: Sistemas de Transporte Inteligentes, Transporte Público, Coleta Automática de Tarifa, Localização Automática de Veículos, GTFS, Matriz de Origem-Destino, Avaliação da Performance do Uso do Transporte

Acknowledgements

To my advisor, Nazareno, thank you for all guidance and support during this whole Masters course. Your brilliant mind and creative ideas were undeniably important for the success of this research. Thank you for putting your confidence in me and not closing doors, but always opening gates. However, the thing I am most grateful for is your humanity, comprehension and patience in the tougher times of this journey. I will never forget that. If one day I become a professor, I want to act in the same way towards my students.

To Matheus, my fellow researcher and coworker, thank you for your willingness in helping me with the experiments, discussing ideas and bringing great insights. Hope you grow and have a very successfull career.

To Talita, my fellow researcher, thank you for having the initial idea and sharing it with us.

To UTFPR professors Keiko Fonseca and Nadia Kozievitch, and student Paulo Diniz, thank you for faccilitating the access to the data and how to use it. Special thanks to Professor Keiko for being so solicitous and answering my e-mails so quickly and even while on vacation. Your support surely made a difference to the results of this.

To URBS and IPPUC, thank you for providing access to all the data and answering questions regarding its understanding and usage.

To my fellow coworkers from Analytics lab, thank you for the support and comprehension when I needed to be away from work to invest time in this research work.

To my friends of Missão Federal Christian Fellowship, thank you for your true friendship and support through all my Masters course. I am very priviledged to be able to serve God alongside you at UFCG.

To my family, Cláudia (mom), Tarciso (dad) and Thiago (brother). Thank you for all the support and for fighting with me through this journey. Mom, thank you for your selfless love which was evidenced so many times and in so many ways throughout these years, for celebrating my victories and crying my defeats. You taught me the definition of love. Dad, thank you for making it all possible with long hours of work driving that truck through the roads of this country. Your tireless energy encourages me. Thiago, thank you for making me smile when I could not see the positive things of life. Your simplicity and peace of mind teach me a lot; and thank you for helping mom on fridays when I had to go to the university - I know it was not easy.

To God, my creator, supporter, redeemer and Lord - What shall I render to the LORD for all his benefits to me? (Psalm 116.12) - thank you for giving me life, strength and intelligence to traverse this journey. You were with me all the time, through the goods and bads, the sunny and stormy days. You give meaning to everything in this short and tough life. To you I dedicate this work.

Contents

List of Acronyms

AFC - *Automatic Fare Collection*

API - *Application Programming Interface*

AVL - *Automatic Vehicle Location*

BRT - *Bus Rapid Transit*

BULMA - *BUs Line MAtching*

BUSTE - *BUs Stop Ticketing Estimation*

C40 - *Cities Climate Leadership Group*

GPS - *Global Positioning System*

GTFS - *General Transit Feed Specification*

IoT - *Internet of Things*

MAE - *Mean Absolute Error*

OD - *Origin-Destination*

OTP - *Open Trip Planner*

SLA - *Service Level Agreement*

URBS - *Urbanização de Curitiba S. A.*

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Today, more than fifty percent of world's population live in cities. Around eight percent of these people live in cities with more than ten million inhabitants [3], and by 2030 it is expected that the number of people living in urban areas reaches five billion [6]. Having this many people sharing cities infrastructure, services and goods creates a complex environment with many challenges to the urban planners and administrators [8], being Urban Mobility among the most defying ones.

Every day, millions of people around the globe use Public Transportation to move around cities, with purposes varying from work or study to leisure. In the big metropoles, people cover large distances and spend a reasonable amount of time in transit. The 2016 report on the scenario of the Public Transportation in the world released by Moovit [13] (trip planning app used daily by 350 million passengers from all over the world), shows that approximately one third of users commute for more than 2 hours daily in big cities as São Paulo, Mexico City and London. Moreover, almost 40% of the users wait more than 20 minutes per day at the bus station. The Moovit Public Transport Index[1] also reports, for several cities where the app is used around the world, the average distance people ride in a single trip within the city, going from 3.6 km in Campina Grande/Brazil to 11.2 km in Hong Kong, for example.

With the recent advances in Computer Hardware and Software, more specifically the Internet of Things (IoT), many Intelligent Transportation Systems were developed. Such sys-

[1] https://moovitapp.com/insights/en/Moovit$_{I}nsights_{P}ublic_{T}ransit_{I}ndex - countries$

tems combine, among other things, information technology, sensors, and the conventional transportation system infrastructure with the objective of improving the use, functioning and management of the system, by supporting safety, mobility and working towards transportation system's efficiency [11]; being available to serve passengers, operators and managers. Some examples of such systems in context of Public Transportation Systems are: Automatic Fare Collection (AFC), Automatic Vehicle Location (AVL), and APC (Automatic Passenger Counting). The data generated by Intelligent Transportation Systems is much valuable, as it can be used to create insights which will helps users make a better use of the system, and administrators evaluate and make improvements on its operation.

Due to the large scale of many such systems, and the many sensors/devices installed on buses and terminals, vast amounts of data are generated every day, consisting of, but not limited to bus GPS and speed records, and passenger boarding information. Such data is collected by different systems with differing objectives. For instance: AFC systems collect information about passenger boarding to be used by the transportation consortium to prove its share of the city passengers, usually recording the passenger card id, bus route and vehicle, but not being concerned about the boarding location. AVL systems, in turn, focus on the bus trajectory traces to assess the routes compliance with the predefined trajectory shapes, mainly recording bus route and vehicle, and the time series of their location geographic coordinates, not keeping track of the trip stops timestamps. If one wants to know the boarding location of passengers in a city, it will be necessary to merge the data from the above systems, which clearly have no direct link to each other. Thereby, given the vast amount of data and the diversity of their source, collection intent, and nature, it becomes hard to integrate and analyze such data,

This study aims at integrating Public Transportation Systems data (shedule specification, bus location and passenger boarding) in order to perform Passenger Trip Inference, which consists in inferring, for each passenger boarding record, the itinerary followed, comprising its origin and destination location and time, along with any bus transfers performed during the trip. With passenger trip itineraries inferred, a number of valuable analyses can be performed, for instance: bus crowding estimation, which can assist both passengers in their itinerary choices, and transit agencies/city managers in monitoring Service Level Agreements (SLAs) and giving insights for the improvement of the system. The last two chapters of this thesis are dedicated to describe applications performed on top of inferred passenger trips data.

All the experiments and analysis described in this work are performed in the context of the Public Transportation System of Curitiba, a 1.8M-inhabitant city in southern Brazil, widely known for its pioneering in the use of technologies to assist urban mobility planning and operation.

1.2 Objectives

1.2.1 General

Develop a computational method to integrate Schedule, Positioning and Ticketing data in order to infer bus passenger trips, thus enabling analytical tasks for city planners.

1.2.2 Specific

- Combine and extend current state-of-the-art heuristics to integrate Schedule, Positioning and Ticketing data capable of inferring trip passenger trips in the Public Transportation System of a large city;

- Validate developed methods by estimating an Origin-Destination Matrix and comparing it to the results of Curitiba's most recent Origin-Destination Survey; and

- Validate the current method by evaluating the inefficiency of the trip itinerary choices performed by bus passengers throughout a large city.

Chapter 2

Background

In this chapter, we describe the data sources and formats most commonly used and available for Public Transportation analysis, discuss the related work and gaps in literature, which will be addressed by the current study; and present the materialization of the data sources in the context of Curitiba.

2.1 Transport Data Sources and Formats

Considering bus systems, three data sources are most often available in different cities: (i) routes and schedules for bus operation, (ii) automatic vehicle location either at a moment or historically, and (iii) automatic fare collection data informing when transit users boarded and sometimes left vehicles.

2.1.1 Transit Routes and Schedule

The *de facto* standard for the description of transit routes and schedule is the General Transit Feed Specification[1] (GTFS), which defines formats for files to be provided by an operator or authority to describe transit supply at three levels. First, *routes* describe meta-information such as route name, transit mode, textual description, and the operator of the different services. Next, *Predefined bus tracks* capture variations of a service over a given route represented by shape linestrings. The shape linestrings are of two types: *complementary shapes*

[1] https://developers.google.com/transit/gtfs/reference/

that must join other shapes to form a complete shape (a shape whose start and end points are the same, i.e., it returns to the starting point); and *circular shapes* that describe a complete shape.

The third level described in GTFS is the *stop times*, which describe the time at which, during a trip, a bus is expected to stop at reference locations. In addition to this description, the location and meta-information of bus stops are typically also specified, and the GTFS feed (instance) from a city normally specify system operation in different situations, such as weekdays and weekends, or public holidays.

Among the data sources considered in this work, GTFS is by far the most often available. Since public transport systems must adapt and evolve according to multiple factors, GTFS information in a city must be dynamic, but not real time, usually being updated at every semester. Naturally, there is often some delay for the information available in the GTFS feed to reflect operational changes.

2.1.2 Vehicle Location

Automatic Vehicle Location (AVL) systems typically track the position of the fleet providing public transport in a city using the Global Positioning System[2] (GPS). GPS devices on buses send data to a server that is commonly able to construct a real-time view of the system, as well as to create a historical record of vehicle movement.

The data made available to transportation system authorities normally contains, for each message sent by a vehicle, a timestamp, the vehicle id, geographical coordinates, and sometimes the route associated to the ongoing trip. The periodicity of data transmission from the vehicles to the server often uses a value in the order of dozens of seconds. In the city we consider in this work, this amounts to around 100MB of GPS data in an ordinary day.

In our experience with multiple large cities in Brazil and Europe, AVL data normally contains no reference for the shape or schedule of the ongoing trip the vehicle is performing. In other words, there is no key to directly associate a bus on a trip with a trip in the GTFS schedule or with a trajectory among those that comprise a route. This association is further complicated by the fact that sometimes vehicles deviate from their prescribed trajectories. This may happen for example due to a traffic change or to an emergency. Such unexpected

[2] https://www.gps.gov/

trajectories are also coupled with two other sources of error: frequent missing data due to network transmission faults, and imprecision in vehicle coordinates due to instrument measurement error.

2.1.3 Fare Collection

A third data source often available for public transportation operators is that from Automated Fare Collection (AFC) systems. These systems collect data from boarding and sometimes alighting of passengers on each vehicle. There are two major types of AFC systems being adopted: flat fare and distance-based fare. For the flat-fare buses, passengers are required to tap their smart cards over a card reader when they get on the bus, but it is not necessary for check-out scans (Entry-Only). For the distance-based AFC system, passengers have to tap their smart cards for both boarding and alighting. The most common fare collection system is the Entry-Only.

For each card tap, AFC records contain at least a timestamp, card id, and vehicle id. Because some of this data may lead to identifying a transport user, this data is normally not publicly available. Also, in our experience with two AFC systems, there is no explicit link between the AFC data and the GTFS specification. Therefore, in order to understand which trip in a schedule had more passengers, one must integrate fare data and schedule information with vehicle location data.

2.2 Challenges in Passenger Trip Inference

2.2.1 Data Integration

Integrating the three data sources described in the previous section allows for a number of important analyses and applications. For example, it is possible to create detailed origin-destination matrices [17; 14; 15; 19], evaluate the adherence of operation to schedule, perform behavior analysis [21; 9], and plan the public transportation system [20; 7].

Nevertheless, there are a number of nontrivial challenges in integrating the three data sources. First, the different data sources do not explicitly reference each other and it is therefore impossible to make straightforward data merges. This creates a difficulty if one intends to evaluate alternative trip options from the schedule (GTFS data) for a passenger that boarded (AFC data) at a given vehicle at a given time (AVL data), for example.

This difficulty is amplified by the fact that the system generating the data is dynamic, and by measurement errors at different levels. If a bus trajectory changes or traffic mandates a trip to deviate from its prescribed trajectory, this is reflected with different delays in vehicle location and schedule data, creating a mismatch. AVL faults and errors also complicate matters.

It is worthwhile to mention that the challenges discussed here are amplified if one is performing a citywide analysis instead of the analysis of a controlled and limited portion of the transportation system, as is sometimes the case (eg. [16]). Dealing with the citywide system includes a number of different types of routes and schedules, and multiplies corner cases and exceptions. Also related to scale, the sheer volume of the data poses challenges for its analysis. In the 1.8M-citizen city analyzed in this work, a month of vehicle location and fare collection data amounts to approximately 5GB. Efficiently analyzing such data volume for larger periods or with detailed methods calls for parallel algorithms.

2.2.2 Large scale of long term data

Most studies only focus on a small period of time, going from few weeks [14] up to a month [15]. Two possibilities for that are: 1) this kind of data is hard to obtain, specially AFC data; 2) as there are a number of computationally costly operations involved in this kind of processing, analyzing a big amount of data may become infeasible, if computational resources are scarce. In our work, we made use of parallel frameworks and implementations to perform a long-term analysis.

2.2.3 Special Boarding Records

Curitiba, as many large cities, has special bus stops, which have a different operation: Terminals and Tube Stations. Terminals are larger and there are few in the city, set in strategic places to serve as hub, connecting city areas. Tube Stations are small, accommodating fewer people, being more popular throughout the city, and connecting less bus lines. Figure 2.1

shows a tube station in Curitiba.

In both of them, the passenger can either get in from the outside, tapping the smart card on the station reader, or get in from a bus which stops at this terminal. Once in the station, the user can board at any bus without being required to tap in the smart card again.

The first station entrance case creates a challenge for our study, as the boarding event happens, but there is no route associated to it, differing from vehicle boarding events. The former case is also challenging because this transfer does not leave any trace, and increases the number of possible destinations the passenger has. *Munizaga et. al.* deal with the former case by choosing a bus from the set of bus lines which go through the station and have a bus stop within walking distance of the passenger's next boarding [14]. This study uses the same heuristics with a few modifications.

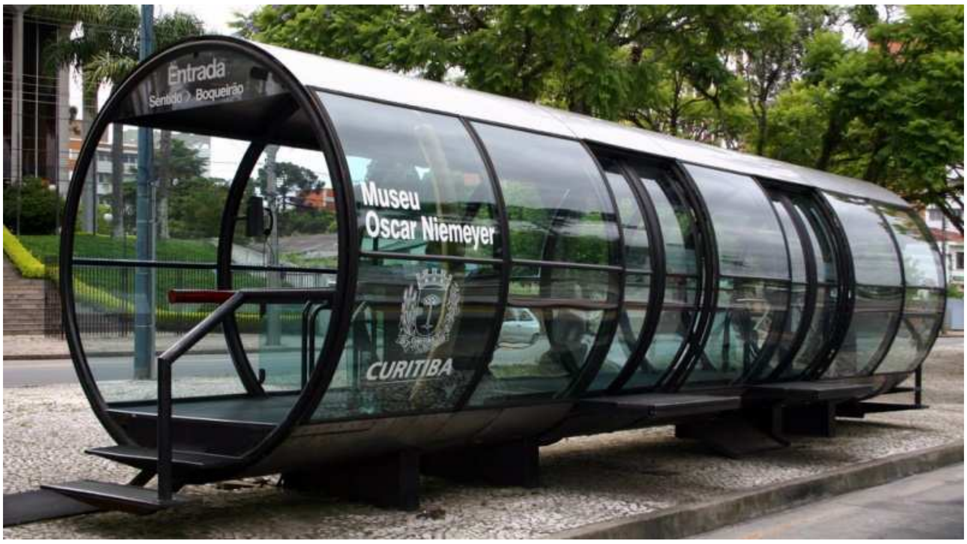


Figure 2.1: Photo of a Tube Station in Curitiba. Image downloaded from: http://conexaoplaneta.com.br/blog/curitiba-disputa-titulo-de-capital-mundial-do-design-2018/ in February 2019.

2.3 The Curitiba Bus System

This section details the transport system of Curitiba to contextualize our analyses.

Besides its relevance as one of Brazil's largest cities[3], Curitiba is also part of the C40

[3] https://www.statista.com/statistics/259227/largest-cities-in-brazil/

Cities Climate Leadership Group[4][5], a group of 90+ world mega-cities engaged into develop innovative data-driven actions to combat climate change, while increasing the health, well-being and economic opportunities of their urban citizens, working in initiatives such as Energy & Buildings and Transportation & Urban Planning.

The city of Curitiba is a reference in Brazil for its bus system, which has pioneered a number of innovations. One of such innovations is the implementation of the Bus Rapid System (BRT) in the 1970s, which defines a separate corridor for bus circulation, improving bus transit speed, carrying capacity and reliability [5; 10]. A more recent innovation is an open transportation data service API that provides updated GTFS and real-time AVL data for citizens[5]. The municipal urban planning agency (URBS) also gracefully provided anonymized AFC data to be used for research purposes in the context of the Brazil-Europe joint research project in which this work was conducted.

The bus system of Curitiba has a fleet of 1,290 vehicles that serve 1.5M passenger trips on a daily basis. The service performs over 23,000 bus trips a day and covers the metropolitan area of Curitiba including nearby districts.

[4] https://www.c40.org

[5] http://www.curitiba.pr.gov.br/dadosabertos/consulta/?grupo=8

Chapter 3

Materials and Methods

3.1 Datasets

3.1.1 GTFS

In this study, we used the GTFS feeds from the first and second semester of 2017, as our AFC data encompasses the months of April to July. Table 3.1 shows a numeric description of each feed used in this study. As we can see, the number of routes, stops and shapes increased from the first to the second feed. In addition, some routes, stops and shapes might have changed or been deactivated/removed. Thus, it is important to use the right GTFS feed when analyzing the movement of buses and passengers around the city. Throughout the experiments described in this work, we selected the correct GTFS feed based on the date of the data to be analyzed: for data regarding events before June 1st 2017, we use the 2017.1 (first semester), and from June 1st on, the 2017.2 (second semester) GTFS feed.

	First semester	**Second semester**
Routes	269	272
Bus Stops/Stations	6,927	6,932
Shapes	616	661

Table 3.1: Summary of the GTFS feeds used.

3.1.2 Bus Position Records

We have vehicle location data from the buses of Curitiba for the whole year of 2017, but for the purposes of this experiment, we only use data from April to July, as we are constrained by the AFC data provided by the Transportation Agency URBS. The AVL data used in the experiment is described in Table 3.2. Table 3.3 describes some fields available in AVL data and provides examples for each one. Figure 3.1 depicts the GPS traces of a trip of a vehicle from route 022.

Year	Month	Days Available
2017	April	28 (except the 21st and 22nd)
2017	May	30 (except the 4th)
2017	June	30 (all)
2017	July	31 (all)

Table 3.2: Number of days available per month in the original AVL dataset

Vehicle ID	Route	Latitude	Longitude	Timestamp
KB603	030	-25.453355	-49.214538	05/01/2017 06:56:31
JA010	711	-25.47364	-49.35012	05/01/2017 06:53:52
BC281	160	-25.43101	-49.272046	05/01/2017 12:17:40
MN600	918	-25.414796	-49.340435	05/01/2017 19:53:58

Table 3.3: Example of AVL data provided by URBS.

3.1.3 Boarding Data

The AFC data used consists of the timestamp, card id and vehicle id of each boarding using a smart card in Curitiba. The URBS website reports that in 2017 (the year of the data used in this study) 60% of boardings are paid using cards[1]. An important limitation in this data is that there are no records of passenger alighting. The boarding data used henceforth in this

[1] https://www.urbs.curitiba.pr.gov.br/transporte/estatisticas/uso_cartoes

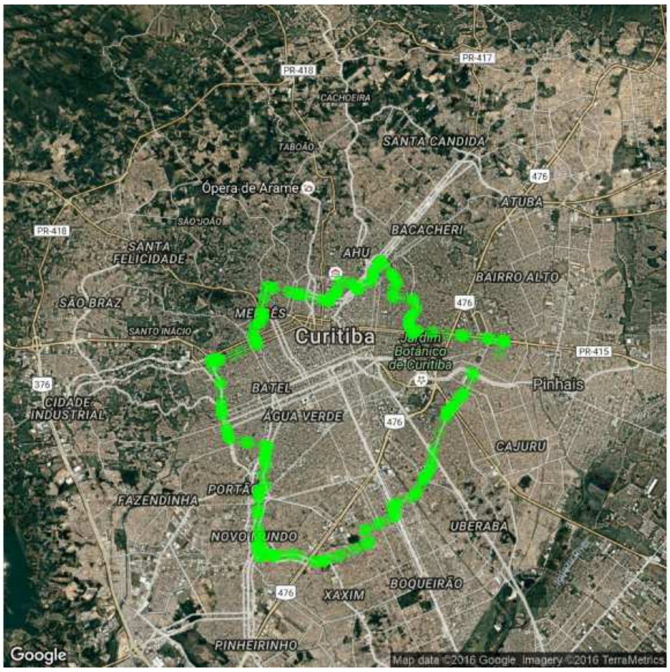


Figure 3.1: GPS traces for Route 022

study encompasses 64 days between April and July 2017, as described in Table 3.4. Such days are not necessarily consecutive, although we have AVL data for almost all of them, as described above.

Year	Month	Days Available
2017	April	1 (April 30th)
2017	May	29 (except days 3 and 31)
2017	June	17 (06/14-30)
2017	July	17 (07/01-17)

Table 3.4: Number of days available per month in the original AFC dataset

Table 3.5 describes some fields available in AFC data and provides examples for each one. Figure 3.2 shows a passenger using its smart card on a bus station in Curitiba.

The total volume of our data is 8.5 GB. As we use a combination of the data sources to perform the analysis, the data used in this work is limited by the joint availability of AVL and AFC data. Besides, we do not use sundays in our analyses, as there is no bus schedule for this weekday in the GTFS feeds. After these filterings, we end up with 54 days from May

Card Number	Route	Vehicle ID	Boarding Timestamp
12345	811	BA022	04/30/2017 14:04:55
23456	ARA	00989	04/30/2017 06:49:52
34567	511	EA170	04/30/2017 10:14:51
45678	706	JC311	04/30/2017 06:13:57

Table 3.5: Example of AFC data provided by URBS. Card Numbers displayed are ficticious for privacy.

Figure 3.2: Photo of passenger tapping URBS smart card on reader. Image downloaded from: https://paranaportal.uol.com.br/cidades/426-projeto-prefeitura-curitiba-cobrador/ in February 2019.

to June, as detailed in Table 3.6.

Year	Month	Days Available
2017	May	25 (except sundays and the 3rd and 31st)
2017	June	15 (06/14-30, except sundays)
2017	July	14 (07/01-17, except sundays)

Table 3.6: Number of days available per month in the joint dataset

3.2 Data Collection and Preparation

The GTFS data comprising the Curitiba Bus System is provided by Web Service[2], which provides data about public transportation in Curitiba, comprising information about lines, bus stops, itineraries, real-time vehicle positions (GPS) and timetables.

AVL data was initially collected through a scraper using the URBS API, until URBS launched an Open Data Portal[3] where they make available static files containing all the information provided by their API for a day in the past. This data amounts to 100MB in an ordinary day. Each record sent by a vehicle contains a timestamp, vehicle id, route id as well as latitude and longitude coordinates. The data contains vehicle location every 20 seconds.

The AFC data is not public due to privacy issues. It amounts to 1.2GB and was gently provided by URBS to us for research purposes. We added the boarding_id column to the raw data as a unique identifier in order to be able to trace each boarding record throughout the Origin-Destination Matrix Estimation Pipeline.

3.3 Data Integration

It is important to note that the above datasets do not have clear or direct points of intersection, so that one could easily integrate their information to perform an analysis of the Transportation System. In order to overcome this challenge, we use two Spark-based Entity-Matching techniques: BULMA and BUSTE. They were mainly proposed by fellow researchers, with

[2] http://www.curitiba.pr.gov.br/dadosabertos/consulta/?grupo=8

[3] http://dadosabertos.c3sl.ufpr.br/curitibaurbs/

our collaboration in the design and validation phases. Both are briefly described in [2], and in more detail in [12].

3.3.1 Integrating GPS Records to GTFS Shapes

Integrating AVL (GPS information) and GTFS data demands identifying, for each vehicle in the AVL data, which of the shapes in the informed route the vehicle has followed. As mentioned in Section 2.1.2, the data made available by the transportation system authorities does not contain information about the route shape or trip identifier for a bus. Due to this missing link, it is infeasible to directly integrate GPS and GTFS data. It is thus necessary to use (i) a vehicle trip generated by a sequence of coordinates reported through GPS, and (ii) a route id, to identify a bus trajectory among a set of shape candidates.

This problem is addressed using an unsupervised technique named BULMA (BUs Line MAtching), proposed by Braz et al. [2], which reduces the search space by using blocking strategies to partition the input GPS data into blocks that can be processed in parallel, and then, for each block, selects the correct shape by a) selecting candidate shapes among the ones described in GTFS, b) determining which of the candidate shapes best fits the given GPS trajectory, and c) labelling the trips with their respective matched shapes, associating each GPS point with its closest point in the matched shape. Figure 3.3 describes the execution of BULMA on a bus trajectory with 3 candidate shapes.

BULMA receives as input all bus GPS records of a given day and the city GTFS, and outputs the same GPS entries with their respective matched shape closest point and labelled trip. Problematic bus GPS traces (for which BULMA could not find matching shape points with confidence) are marked for future consideration in the processing pipeline. The execution of BULMA on the whole experiment dataset (64 days - about 6GB of GPS data) on a Spark cluster with 1 master and 8 slave nodes (node configuration: 2 VCPUs, 4GB RAM and 60GB of storage) took aproximately 5.5h.

3.3.2 Integrating Bus Trips to Boarding Records

Having the bus trips inferred by BULMA, the next challenge on the way to inferring passenger trips origins and destinations is to match the boarding records to the bus trips data.

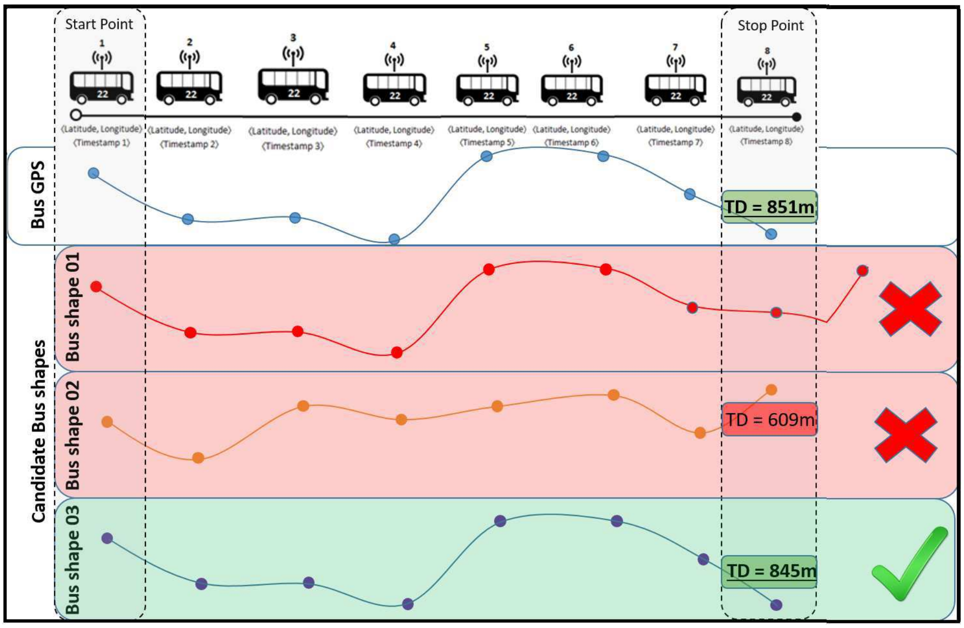

Start Point
Stop Point
1
2
3
4
5
6
7
8
22
(Latitude, Longitude) (Timestamp 1)
(Latitude, Longitude) (Timestamp 2)
(Latitude, Longitude) (Timestamp 3)
(Latitude, Longitude) (Timestamp 4)
(Latitude, Longitude) (Timestamp 5)
(Latitude, Longitude) (Timestamp 6)
(Latitude, Longitude) (Timestamp 7)
(Latitude, Longitude) (Timestamp 8)
Bus GPS
TD = 851m
Candidate Bus shapes
Bus shape 01
Bus shape 02
TD = 609m
Bus shape 03
TD = 845m

Figure 3.3: Example of BULMA execution for route 022 containing three candidate shapes.

The AFC data, as the AVL data, has no direct link to GTFS, as mentioned in Section 2.1.3. Thus, the information of the bus stop where the passenger boarded is not directly or easily available.

We overcome this challenge by using another Entity-Matching technique: BUSTE (BUs Stop Ticketing Estimation), also proposed in [2], which integrates BULMA's output (bus trips) with ticketing data, by 1) reading the bus trips and filtering out its missing values, 2) performing a time interpolation over the shape points (as only matched GPS records have time associated), 3) matching the bus stops to the interpolated shape based on distance, and 4) matching the boarding records to the stop whose time difference is shorter.

BUSTE builds on BULMA output by integrating the stops and ticketing data, assembling a dataset which describes the city's bus trips and their associated passenger boardings over a period of time. Such dataset is very useful for a number of applications, the Origin-Destination Matrix estimation being one of them. The execution of BUSTE on the whole experiment dataset on the same Spark cluster as the BULMA execution took aproximately 3h.

Both Entity-Matching applications are Open Source, being available at Github[4]. They

[4] https://github.com/eubr-bigsea/EMaaS

were implemented in Java using Apache Spark[5] framework to enable parallel processing.

3.3.3 Enhance BUSTE output with special station boardings

As explained in Section 2.2.3, there are a number of boarding records which are not associated to a bus route and vehicle, but instead, refer to a smart card reader machine, which is located at either a Terminal or a tube station. This kind of record corresponds to about 50% of records in our traces, showing that this type of station is heavily used by passengers in their commute in Curitiba. To avoid losing such proportion of the data in our analyses, we perform an extra step in our analysis pipeline, which deals with this nuance, as described next.

Our heuristics to integrate the non-vehicle boardings has three steps. First, we filter GPS-Ticketing matches performed by BUSTE whose match time difference is larger than 30 minutes, assuming those matches are inconsistent and thus, untrustworthy. This value is the maximum acceptable time difference between two sequential buses of the same line at a stop in Curitiba, defined by contract. Next, using a reference table gently provided by Paulo Diniz which he built for his Master Thesis [4], we use the card reader machine codes found in the special boarding records to match passengers to bus station ids. Finally, the GTFS stops table is used to add the geographical coordinates for each bus station or parent station (in the case of terminals) to the record. After the addition of non-vehicle boardings, the number of boarding records in the dataset rose from 5.5 million to 9 million, jumping from 29% to 48% of the total number of boardings in the original dataset (18.6 million records).

3.4 Tools and Resources used

All the code implemented for the experiments is Open Source, being available on Github[6]. Most of the code was implemented in Python and R, using well-known data analysis libraries, such as: Pandas[7]/NumPy[8] (Python) and dplyR (R). We also made use of Spark for parallel processing.

[5] https://spark.apache.org/

[6] https://github.com/analytics-ufcg/people-paths

[7] https://pandas.pydata.org/

[8] http://www.numpy.org/

As part of our analyses, we needed to have a routing service to compute the itinerary alternatives between two points A and B, using the bus schedule described in the GTFS feed. The most widely known tool which offers such service is Google Directions API[9]. However, this API is not free of charge, and costs around 5.00 USD per 1000 queries per month[10].

In order to reduce research costs, we looked for Open Source routing tools, and found Open Trip Planner[11], a Java-based project started in 2009 that provides passenger information and transportation network analysis services, being supported and used by public agencies, startups, researchers, and so on.

A local OTP server setup only requires a GTFS feed and OpenStreetMap[12] data for the geographical region the GTFS feed refers to. As our experiments generate in the order of millions of queries, we set up a Docker Swarm to orchestrate 20 OTP server containers under an nginx[13] proxy server that acts as a load balancer.

An OTP routing query from origin o to destination d at time t returns a set of itinerary alternatives composed of one or more legs, which can be of two modes: walk and bus legs. Each leg comprises a start and end time, start and end location and expected duration. Bus legs also have route and start/end stop ids.

[9] https://developers.google.com/maps/documentation/directions/start

[10] https://developers.google.com/maps/documentation/directions/usage-and-billing

[11] http://www.opentripplanner.org/

[12] https://www.openstreetmap.org

[13] https://www.nginx.com/

Chapter 4

Passenger-Level Trip Inference

This chapter describes the methods and heuristics implemented to infer trip traces (boarding and alighting location and time, and bus transfers) after the preprocessing steps performed to integrate the data sources. The goal is to build the Passenger Trip Traces data set, which comprises a detailed description of the passengers trips around the city of Curitiba. This data set is very useful for a number of applications, as will be described and exemplified in the next chapters.

4.1 Input Data for Trip Inference

After the preprocessing pipeline there are three data sets that we consider for trip inference. Two of these have already been described:

- **Bus Trips** - comprises all the tuples (route, vehicle, trip, stop, timestamp) found in the enhanced BUSTE dataset, which represent the observed operation of all city buses in a period. This data set amounts to 23.7 million observations from 64 days between April and July.

- **Geolocated Boardings** - comprises all the tuples (passenger, boarding_id, route, vehicle, stop, timestamp) found in enhanced BUSTE dataset, which represent the matched boarding events (at buses or terminals/tube stations) for all passengers in a single day. This data set amounts to 9 million observations.

The third dataset implies in an additional preprocessing step, and is called **Inexact Passenger Trips** data. This pre-processing is needed for destination estimation. Recall that our card data contains only boardings, and no alightings. To infer aligthings, our method is grounded on two standard assumptions about trip destination inference found in literature [1]:

1. the destination of a trip performed by a passenger is most probably located close to its next trip origin;

2. the destination of the last trip of a day is most probably located close to the origin of the first trip of that day.

The rationale behind these assumptions is that commuters usually perform a chain of trips during the day, to go from home to some activity place (work, school, gym, supermarket, etc.), maybe moving between activity places, and then getting back home. Based on that, it can be presumed that the destination of a trip is located somewhere close to the origin of its next trip; and the destination of the last trip of a day is the first origin (traveling back home).

With these considerations, we create a data set of (Origin,Next-Origin) - (o,n) pairs. In our data set, such pairs can be created for each passenger with more than one trip. At the same time, there is no data for our heuristics to estimate the destination of passengers who have a single trip in a day, and trips from such passengers are therefore removed from the data. After this filtering step, there are approximately 7 million (o,n) pairs remaining in the data set, which implies 2 million single-trip-in-a-day passenger trips were discarded.

4.2 Destination Estimation

Using the data sets built in the previous phases, in order to infer the trip traces of a passenger trip between an origin o and an unknown destination d, which is close to the next origin n, one needs to identify itinerary alternatives, according to schedule, that could have been chosen to execute the given trip; to pair these scheduled itineraries to the actual Bus Trips data, obtaining a set of observed alternative itineraries; and finally to select from this set of itineraries the one which was most likely chosen by the passenger. Each of these steps is described in more detail below.

4.2.1 Find trip itinerary alternatives

Having the Origin/Next-Origin boarding pairs, we need to find out how the user went from an origin o to a next-origin n using the public transportation system. To solve this problem, we need to be able to compute the possible itineraries the user could have taken which satisfy both the geographical and time constraints imposed by the consecutive boarding records.

We accomplish the above task resorting to Open Trip Planner[1] (OTP). OTP is an open source platform for multi-modal and multi-agency journey planning that operates in our experiments using GTFS data from the municipality of Curitiba and street network data from Open Street Map. For a given origin and destination, OTP employs Multiobjective A and the Tung-Chew heuristic algorithm [18] to find the shortest itineraries considering the bus system, including connections, and walking, subject to some practical restrictions on number of connections and walking distances. We opted to use OTP because it is free software and thus we were able to deploy our own routing service. Alternatives such as Google Directions API[2] incur in prohibitive costs for number of queries our experiments demand.

An OTP routing service was deployed with two routers, one for each GTFS feed, as described in Section 3.1.1. As those feeds have no bus trips scheduled for Sundays, it is not possible for OTP to perform routing for trips on this day of week, and so we remove Sundays from our analysis, as described in Table 4.1.

Month	Sundays Removed
April	1 (April 4th)
May	4 (May 7th, 14th, 21st, and 28th)
June	2 (June 18th and 25th)
July	2 (July 2nd and 9th)
Total	9

Table 4.1: Sundays removed from data set per month

The core of our heuristic consists in processing the (o,n) pairs in the inexact passenger trip data by performing the following steps for each (o,n) pair:

[1] http://www.opentripplanner.org/

[2] https://developers.google.com/maps/documentation/directions/usage-and-billing

1. Select the proper GTFS feed (and thus OTP router) according to the trip date;

2. Query the proper router for at most 10 possible itineraries between o's location as origin and n's as destination. Additionally, o's start time and bus route taken (in the case of vehicle boardings) are passed as constraints, as well as a maximum walking distance for walking legs of 1km (in literature, this threshold varies from 400m [22] to 2km [17]);

3. For each leg of each itinerary alternative returned, add a row to a result data set with the leg's start/end location and time, and the transportation mode used, be that a bus route or walking.

Each OTP query takes approximately 0.2 seconds. This poses another challenge to our experiment, as it performs about 7 million queries, which would take 16 days to run. Using a cluster infrastructure which runs several OTP server replicas in parallel provided a considerable speedup, running all queries in 2 days.

4.2.2 Infer Passenger Trip Traces

After the previous step, we have, for each (o,n) pair, a set of itinerary alternatives the passenger could have taken to go from o to n, mixing walk and bus legs. Next, we need to match those scheduled itineraries to executed itineraries, and then select the alternative that the passenger most probably chose.

For that, we consider that a trip was performed as one of the available alternatives based on the trips that were indeed performed at that moment by the bys system. This is done using the Bus Trip dataset and a penalty score that guides our choice of which performed trip is most likely the alternative taken by the passenger. The steps of this process are:

1. Discard (o,n) pairs for which OTP returned no itineraries;

2. Match all origin boardings (vehicle and terminal) to the first bus leg of the OTP itinerary alternatives using the *boarding id*, *route* (when available) and *origin stop id* information;

3. Add new (o,n) pairs to the Inexact Passenger Trips data set to account for each alternative itinerary subsequent leg (walk or bus transfer)

4. Match OTP itineraries bus legs origin and destination to bus trips in the Bus Trips data set using the route and location as matching keys. Matches with start/end time difference longer than 60 minutes are discarded (notice OTP legs origin/destination can be matched to multiple Bus Trips records).

5. For each OTP itinerary leg (walk or bus), choose the best match among the set of matches produced in the previous step (when applicable), by:

   (a) filtering out the matches whose actual start time is earlier than chosen previous leg's end time (Notice that the previous leg can be a walk leg);

   (b) selecting the leg whose start time is earlier (the first bus which gets at the bus stop).

6. Add bus stops/stations geographic coordinates to the legs data using the appropriate GTFS feed.

7. For each trip, evaluate all actual itinerary alternatives remaining in the data set, assigning them a penalty score P , defined as:

$$P = 2 \times start\_diff + itinerary\_duration + 10 \times num\_transfers,$$

   where $start\_diff$ is the time difference between the boarding timestamp and the itinerary alternative actual start time, $itinerary\_duration$ is the actual duration on the itinerary alternative (found after the above matching/choice steps), and $num\_transfers$ is the number of bus transfers planned in the itinerary alternative. This penalty score will be used in the evaluation of the itinerary alternatives when selecting the itinerary most likely chosen by the passenger.

8. For each (o,n) pair, choose the itinerary alternative with the lowest penalty score

As Step 5 needs to be performed serially within each group of leg matches (the processing of each row depends on the result of processing the previous row), this step becomes computationally costly, and therefore was also executed on a parallel setup. Running 8 parallel threads, we were able to infer the trip traces for the data set in 38 hours on an Intel Core i7 with 8 VCPUs and 8GB RAM.

At the end of this step, we build our target data set, which we call the Passenger Trip Traces data set. This data contains, for each boarding record, the full inferred trip trace performed by the passenger, including walk, bus legs and transfers.

During our pipeline, with the filtering operations in each step, a number of boarding records was removed from the data set. In general, our approach is to prioritize confidence over completeness. This is the same approach followed by other researchers, such as Nunes et. al. [15]. Our final data set contains approximately 3.2 million boarding records, which represents 17.2% of the 18.6 million found in the initial data set, and slightly less than half of the trips of passengers that have multiple trips on a same day in our data set.

It is worthwhile stressing that the Passenger Trip Traces data set enables the development of applications that rely on fine grained information about the bus system at city scale. In the following chapters we describe first a comparison of the produced data against an approximate behchmark (an Origin-Destination Survey) and an example of a city-wide analysis enabled by this data.

Chapter 5

Origin-Destination Matrix Estimation

To the best of our knowledge, there is no ground truth against we can compare the Passenger Trips Traces our method infers. As there is no alighting data in the Curitiba system, there is no reference information about where passengers finish their trips measured at the passegner level. In this chapter, we use an alternative information gathered by the City of Curitiba as their reference for estimating mobility patterns to evaluate the precision of our method.

As part of their city planning, mobility agencies often employ *Origin-Destination surveys*. These surveys randomly sample residences in previously delimited sectors and presencially ask residents about their mobility patterns on a regular day. An important outcome of this survey is the Origin-Destination matrix (OD-Matrix) [16], which aggregates the estimates of trips between each pair of sectors used in the survey.

Because this data is notoriously valuable for planners, we use the OD-Matrix estimated for Public Transport in a recently conducted OD Survey in Curitiba as a reference against which we compare an OD-Matrix estimated through our Passenger Trip Traces.

5.1 Curitiba's 2016 Origin Destination Survey

In 2016, Curitiba Municipality, as part of their Urban Mobility Plan, contracted an Origin-Destination Survey, whose results are publicly available as open data[1]. The goal of the survey was to understand the mobility pattern of Curitiba and its Metropolitan Area. It was performed in Curitiba and 16 other municipalities in its Metropolitan Region. For the purposes

[1] http://www.ippuc.org.br/mostrarpagina.php?pagina=536&idioma=1&ampliar=n%E3o

of the research, over 45,000 people from 15,000 homes were interviewed, which provided information on about 76,000 trips. The survey cost was about 6 million BRL (Brazilian Real).

5.2 OD Matrices

OD-Matrices consist of square matrices where the columns and rows are labelled identically, and a cell $[r,c]$ represents the number/proportion of occurred trips starting on r and ending on c.

The first step to building the OD-Matrices is to split the evaluated area into regions (at any desired level: continent, city, neighborhood, etc.), and then aggregate the number of trips occurred between each pair of regions.

5.2.1 Zones and Macrozones in the OD Survey

The Survey defined a set of 181 zones in the Metropolitan area of Curitiba, which can be aggregated into 10 Macro Zones. In order to build the Zones Dataset, we join the table Zoneamento_Modelacao, available in the Matrizes database found in the survey deliverables[2], and the shapefile which defines the 181 zones, also provided as a deliverable. The final zones dataset is a geometric dataset, comprising zone ID, geometry (polygons) and metadata such as population density and municipality.

5.2.2 Estimating a Public Transport OD-Matrix from the Survey

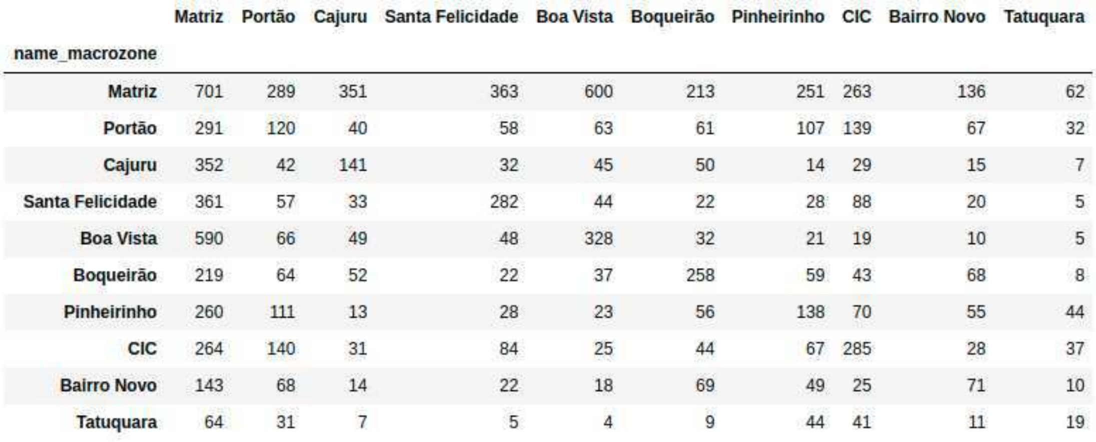
We first built the Survey OD-Matrix, which is being used as the ground truth. One of the deliverables of the OD Survey is the journeys data set, which describes the data obtained from the interviews. It contains data about 27 thousand interviewees and 72 thousand journeys (trips) performed by them. It is not limited to Public Transportation, but also includes private transportation as well as biking and walking. This data set is the input for the Survey OD-Matrix.

[2] http://www.ippuc.org.br/visualizar.php?doc=http://admsite2013.ippuc.org.br/arquivos/documentos/D536/D536_003_BR.zip

The first step to build the Survey OD-Matrix is to select only the trips whose transportation medium was the Municipal or Metropolitan Bus, amounting to 13 thousand trips, which represents about 18% of the total number of journeys. The resulting dataset contains the zone ID, macro zone ID, municipality and time for both origin and destination of the journey.

As the number of trips remaining in the data set is small relative to the population of the zones, we opt not to build OD-Matrices at the zone level. Otherwise there would be on average small samples from each zone, and there would be zones with no interview. Thus, we only build OD-Matrices at the Macrozone level.

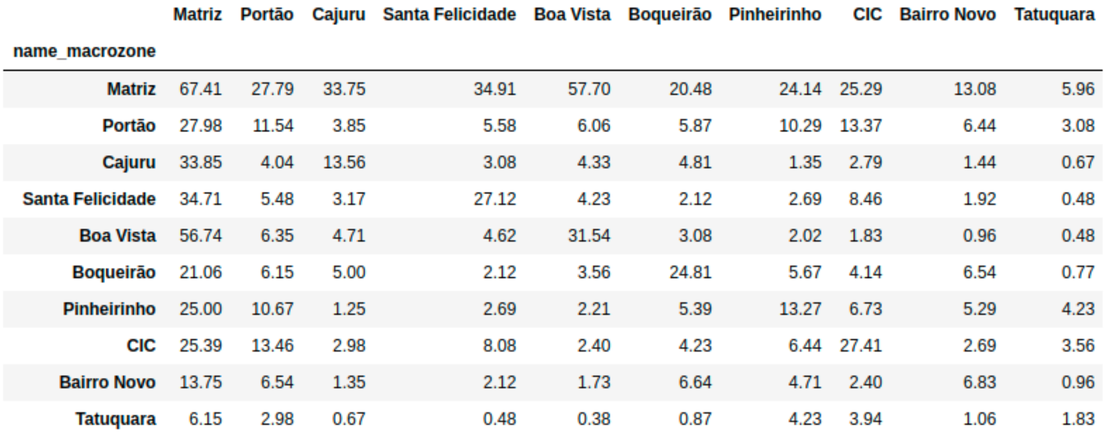
For that, we aggregate the trips by origin and destination Macrozone. We have also created a normalized version of the matrix, by transforming the raw trip counts into *proportions* of the total number of trips in the dataset and multiplying them by 1000. If cell $[r, c] = 50$, this means that, for every 1000 trips made in Curitiba on a day, 50 of them have their origin in Macrozone r and destination located in Macrozone c. This normalization enables the comparison to matrices generated from different data sources with different sample sizes, as is our case, and the chosen unit facilitates the information understanding. The Survey Macro-zones OD-Matrix and Normalized OD-Matrix are shown in Figures 5.1 and 5.2, respectively.



	Matriz	Portão	Cajuru	Santa Felicidade	Boa Vista	Boqueirão	Pinheirinho	CIC	Bairro Novo	Tatuquara
name_macrozone										
Matriz	701	289	351	363	600	213	251	263	136	62
Portão	291	120	40	58	63	61	107	139	67	32
Cajuru	352	42	141	32	45	50	14	29	15	7
Santa Felicidade	361	57	33	282	44	22	28	88	20	5
Boa Vista	590	66	49	48	328	32	21	19	10	5
Boqueirão	219	64	52	22	37	258	59	43	68	8
Pinheirinho	260	111	13	28	23	56	138	70	55	44
CIC	264	140	31	84	25	44	67	285	28	37
Bairro Novo	143	68	14	22	18	69	49	25	71	10
Tatuquara	64	31	7	5	4	9	44	41	11	19

Figure 5.1: Survey Macro Zones OD-Matrix

It is useful to bear in mind that if trip origin and destination were distributed uniformly across macrozones independent of their characteristics, then $[r_i, c_i] = 10.$ for all cells in the OD matrix (10x10 = 100). In contrast, the matrix in Figure 5.2 spans a range from **0.38 to 67.41**. A heatmap built from the OD Matrix values can be seen in Figure 5.3.



name_macrozone	Matriz	Portão	Cajuru	Santa Felicidade	Boa Vista	Boqueirão	Pinheirinho	CIC	Bairro Novo	Tatuquara
Matriz	67.41	27.79	33.75	34.91	57.70	20.48	24.14	25.29	13.08	5.96
Portão	27.98	11.54	3.85	5.58	6.06	5.87	10.29	13.37	6.44	3.08
Cajuru	33.85	4.04	13.56	3.08	4.33	4.81	1.35	2.79	1.44	0.67
Santa Felicidade	34.71	5.48	3.17	27.12	4.23	2.12	2.69	8.46	1.92	0.48
Boa Vista	56.74	6.35	4.71	4.62	31.54	3.08	2.02	1.83	0.96	0.48
Boqueirão	21.06	6.15	5.00	2.12	3.56	24.81	5.67	4.14	6.54	0.77
Pinheirinho	25.00	10.67	1.25	2.69	2.21	5.39	13.27	6.73	5.29	4.23
CIC	25.39	13.46	2.98	8.08	2.40	4.23	6.44	27.41	2.69	3.56
Bairro Novo	13.75	6.54	1.35	2.12	1.73	6.64	4.71	2.40	6.83	0.96
Tatuquara	6.15	2.98	0.67	0.48	0.38	0.87	4.23	3.94	1.06	1.83

Figure 5.2: Normalized Survey Macro Zones OD-Matrix

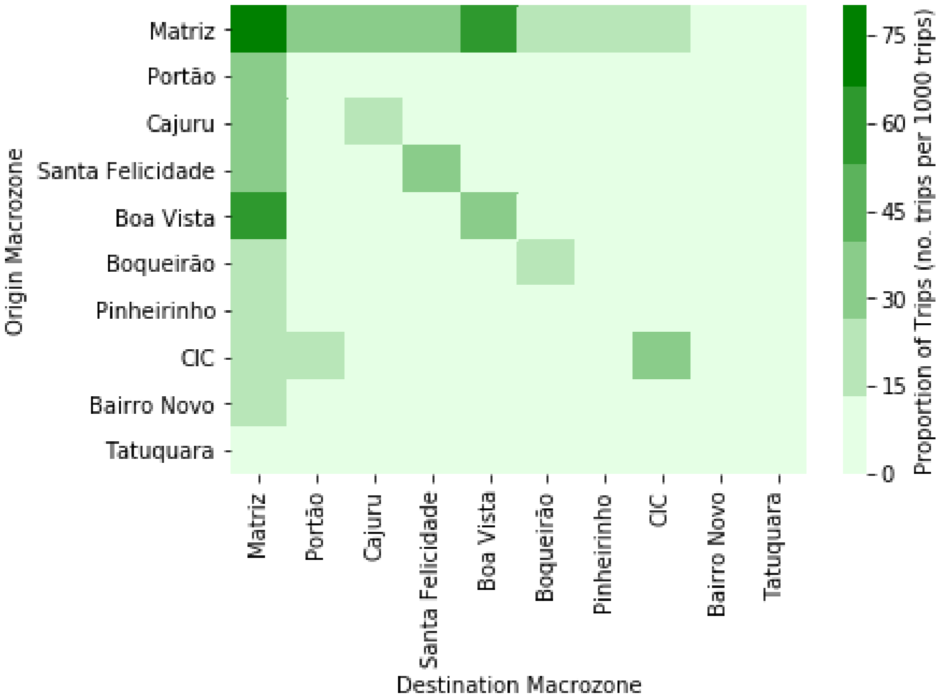


Figure 5.3: Heatmap of Normalized Survey Macro Zones OD-Matrix

5.2.3 Baseline OD-Matrices Construction

Next, we create baseline OD-Matrices to be compared to our method on reproducing the Survey results. These are meant as reference points for the error that would be seen if one is estimating the Public Transportation OD-Matrix only from census data, only from ticketing data (without passenger trip inference), or combining census and ticketing data. Accordingly, we consider 3 baselines:

1. Naive Population-Proportion Baseline (solely based on census data)

2. BUSTE Population-Proportion Baseline (solely based on ticketing data)

3. BUSTE Origin-Proportion Baseline (based on a combination of census and ticketing data)

For the following subsections, we consider that there is a square 10×10 matrix T that we must estimate, where each element $t_{o,d}$ represents the proportion of all trips that originate in macrozone o and have their destination in macrozone d, with the proportion unit being the number of trips per 1000 trips in the data. Additionally, there exist two vectors of 10 elements P and B. Element p_i of vector P contains a population estimate for macrozone i, and element b_i from B contain the number of boardings originating in B on a day, extracted from the ticketing data after processing with BUSTE.

Naive Population-Proportion Baseline

The Naive Population-Proportion Baseline considers that the number of origin and destinations that reside in a macrozone are proportional to its population. Thus, in this baseline the number of estimated trips $t_{o,d}$ from an origin macrozone o to a destination macrozone d is given by equation 5.1, where p_o and p_d are respectively the populations of macrozones o and d; and $\sum_{i=1}^{10} p_i$ corresponds to the total population of all macrozones.

$$t_{o,d} = \frac{p_o}{\sum_{i=1}^{10} p_i} \frac{p_d}{\sum_{i=1}^{10} p_i} \tag{5.1}$$

A heatmap of the resulting normalized OD-Matrix for this baseline is shown in Figure 5.4.

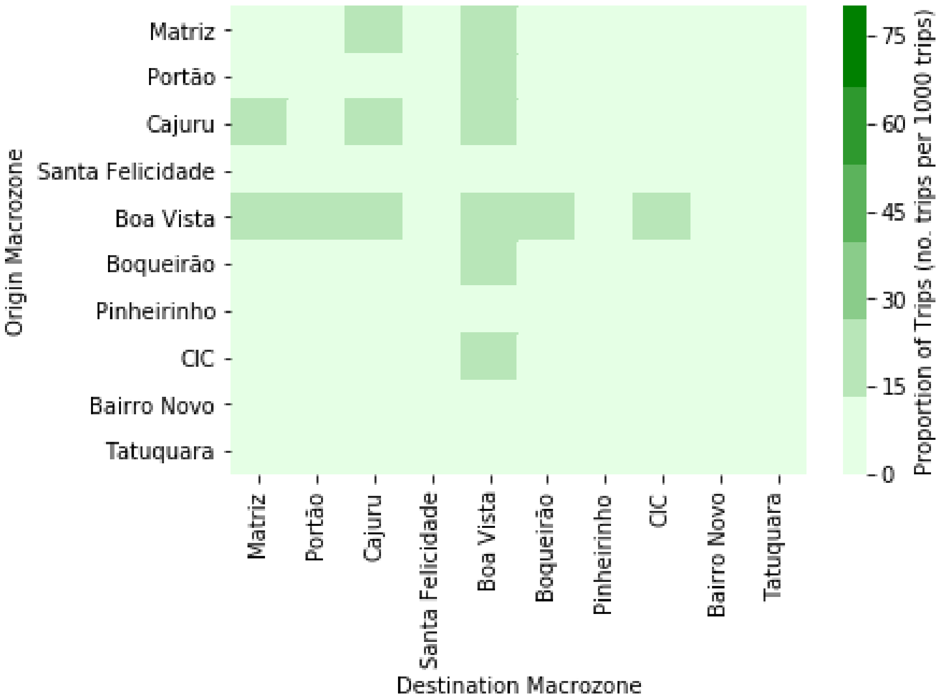


Figure 5.4: Heatmap of Naive Population-Proportion Macro Zones Baseline OD-Matrix

BUSTE Population-Proportion Baseline

BUSTE Population-Proportion Baseline is generated from the trips origin locations identified by BUSTE (Section 3.3.2). More detailedly, we perform a spatial join of trips origins to the zones dataset (Section 5.2.1) and group the origins by Macrozones. Thus, analogally to the previous baseline, $t_{o,d}$ is given by equation 5.2, where b_o is the number of BUSTE trips originated at macrozone o, p_d is the population of macrozone d; and $\sum_{i=1}^{10} b_i$ and $\sum_{i=1}^{10} p_i$ correspond respectively to the total number of BUSTE trip origins, and the total population of all macrozones.

$$t_{o,d} = \frac{b_o}{\sum_{i=1}^{10} b_i} \frac{p_d}{\sum_{i=1}^{10} p_i} \tag{5.2}$$

A heatmap of the resulting normalized OD-Matrix for this baseline is shown in Figure 5.5.

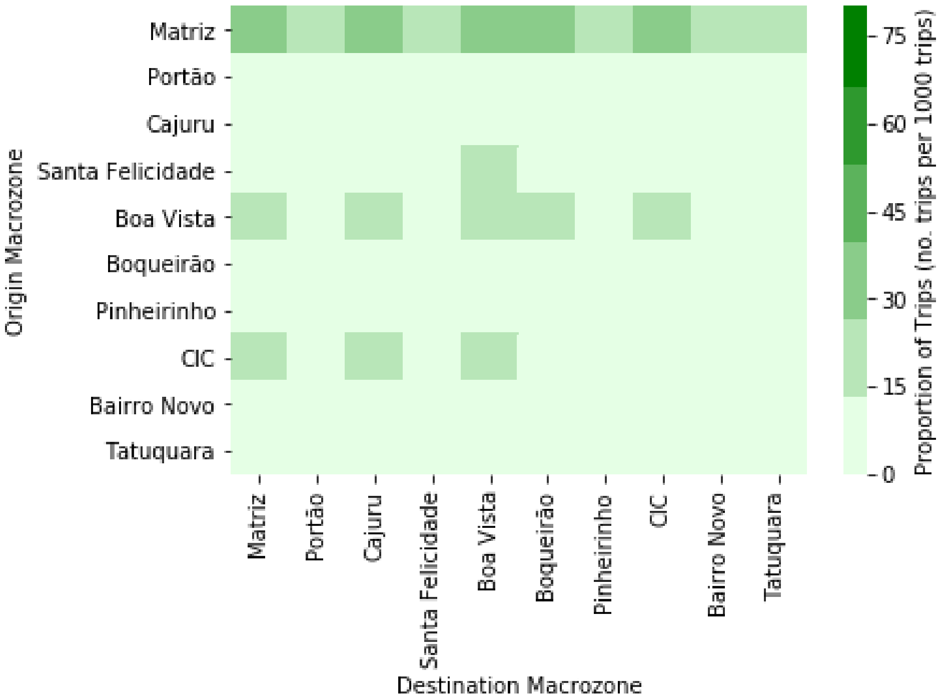


Figure 5.5: Heatmap of BUSTE Population-Proportion Macro Zones Baseline OD-Matrix

BUSTE Origin-Proportion Baseline

The third baseline is generated similarly to the second, but instead of the proportion of the population of the destination macrozone, we use the proportion of BUSTE trip origins occurred in the destination macrozone to estimate the number of destinations in each macrozone. Thus, analogally to the previous baseline, $t_{o,d}$ is given by equation 5.3, where b_o and b_d are respectively the number of BUSTE trips originated at macrozone o and the number of BUSTE trips originated at macrozone d; and $\sum_{i=1}^{10} b_i$ corresponds to the total number of BUSTE trip origins.

$$t_{o,d} = \frac{b_o}{\sum_{i=1}^{10} b_i} \frac{b_d}{\sum_{i=1}^{10} b_i} \tag{5.3}$$

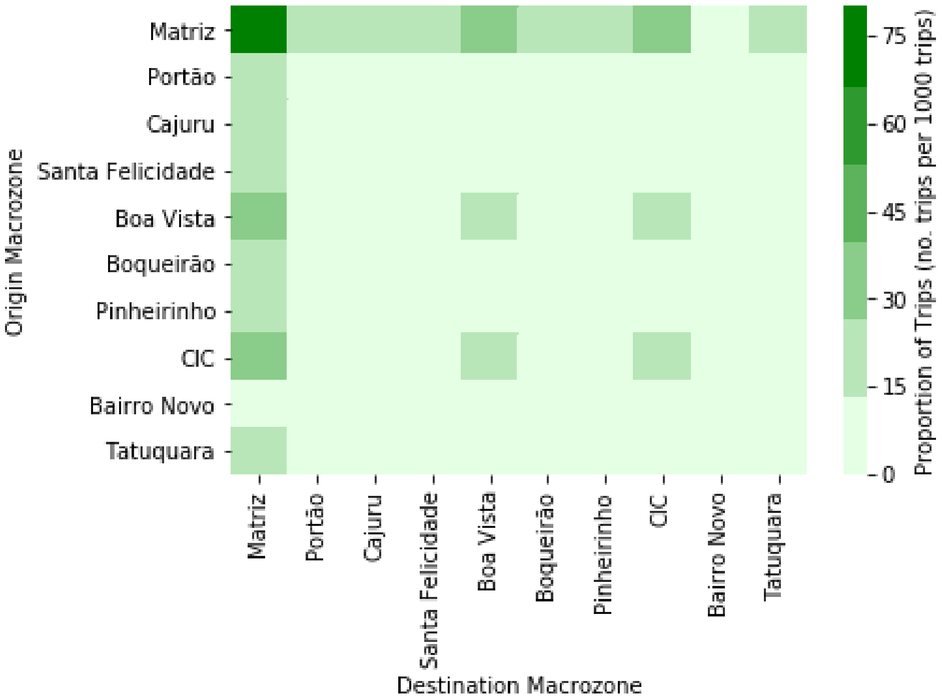


Figure 5.6: Heatmap of BUSTE Origin-Proportion Macro Zones Baseline OD-Matrix

Figure 5.6 shows the heatmap of the normalized OD-Matrix for this baseline.

5.2.4 Inferred Trips OD-Matrix Construction

To create the OD-Matrix using the Inferred Trips dataset, we:

1. Aggregate trips (o,d) pairs with multiple legs into a single record (where the trip origin is the first origin, and the trip destination is the last destination), to be able to compare with survey and baseline data.

2. Perform a spatial join of both trips origin and destination locations to zones dataset (by joining with a zone when the origin/destination location is within the zone polygon).

3. Generate the macro zones OD-Matrices (Original and Normalized) in the same way as we did for the above matrices.

Figure 5.7 displays a heatmap of the Normalized Macro-Zones OD-Matrix built from the Inferred Trips dataset, respectively.

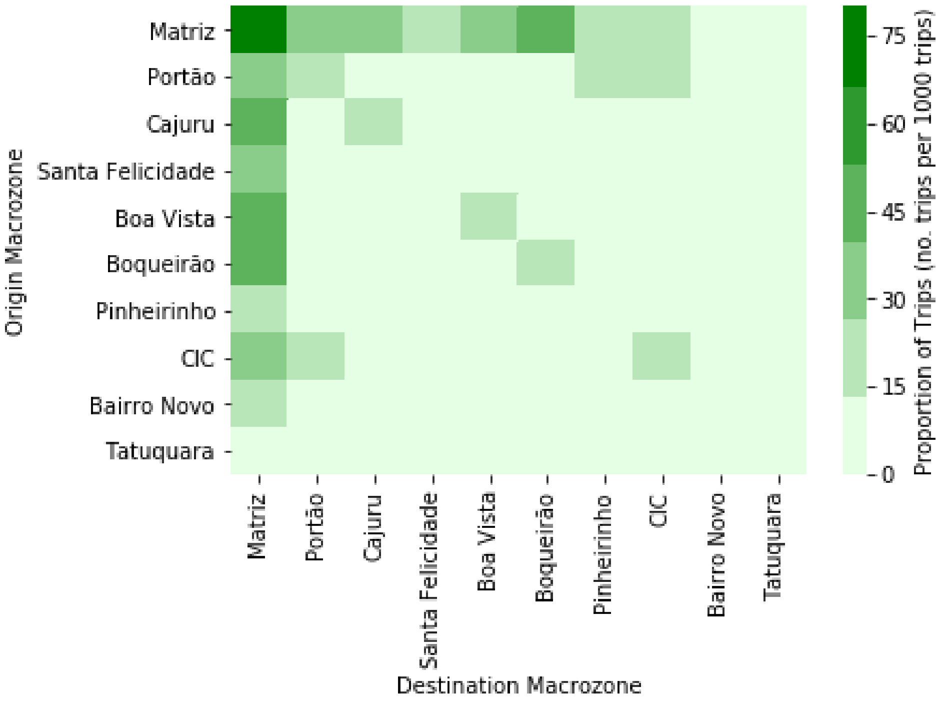


Figure 5.7: Heatmap of Inferred Trips Macro Zones OD-Matrix

5.3 OD-Matrices Comparison

The comparison of the matrices is performed using their normalized (proportional) versions. We subtract the Survey OD-Matrix from the estimated OD-Matrices (Baselines or Inferred), generating an Error OD-Matrix. We then analyze the error distribution and aggregate it in a single metric using MAE (Mean Absolute Error). Next, we show the comparison of each generated OD-Matrix to the results of the Survey.

5.3.1 Naive Population-Proportion Baseline

A heatmap of the Error Matrix obtained when comparing the Naive Population-Proportion Baseline OD-Matrix to the Survey OD-Matrix can be seen in Figure 5.8, and the error distribution histogram is shown in Figure 5.9. The MAE of the error matrix is 8.57 trips per 1000 trips in the data set.

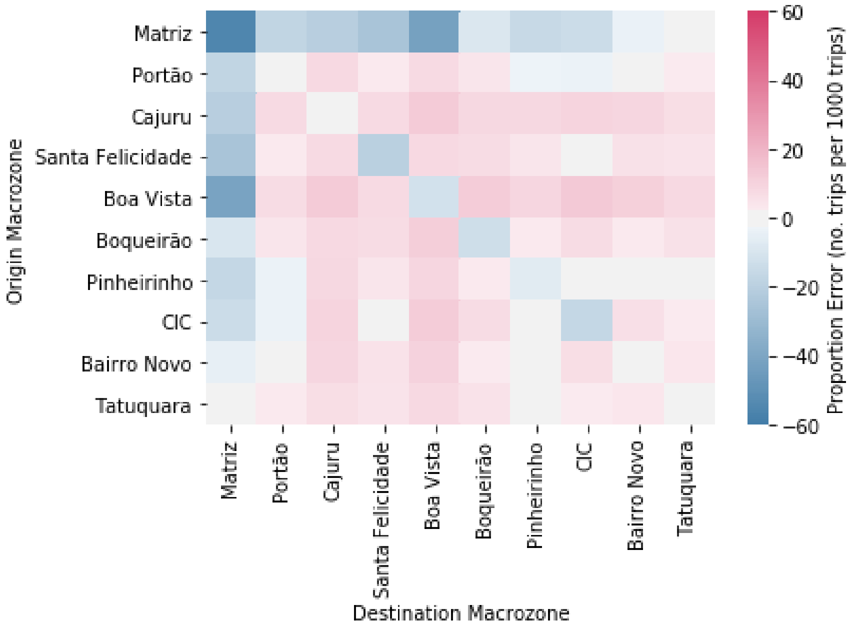


Figure 5.8: Heatmap of Comparison Error between Naive Baseline and Survey OD-Matrices

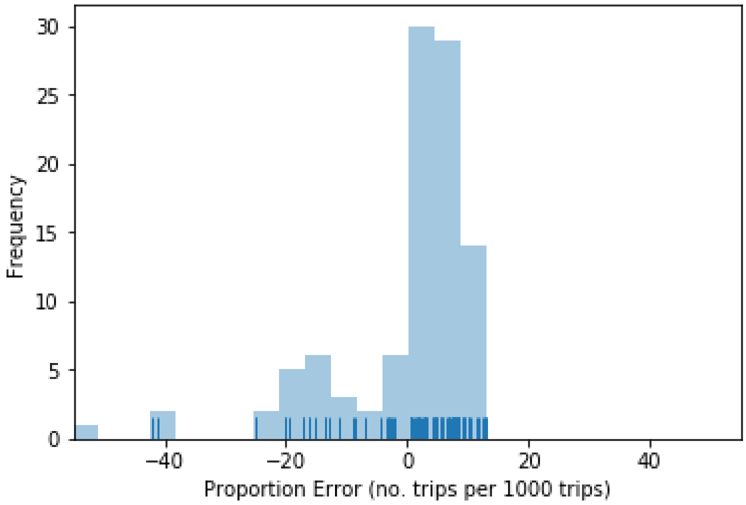


Figure 5.9: Naive Baseline OD-Matrix Comparison Error Distribution

5.3.2 BUSTE Population-Proportion Baseline

A heatmap of the Error Matrix obtained when comparing the BUSTE Population-Proportion Baseline OD-Matrix to the Survey OD-Matrix can be seen in Figure 5.10, and the error distribution histogram is shown in Figure 5.11. The MAE of the error matrix is 6.94 trips per 1000 trips in the data set.

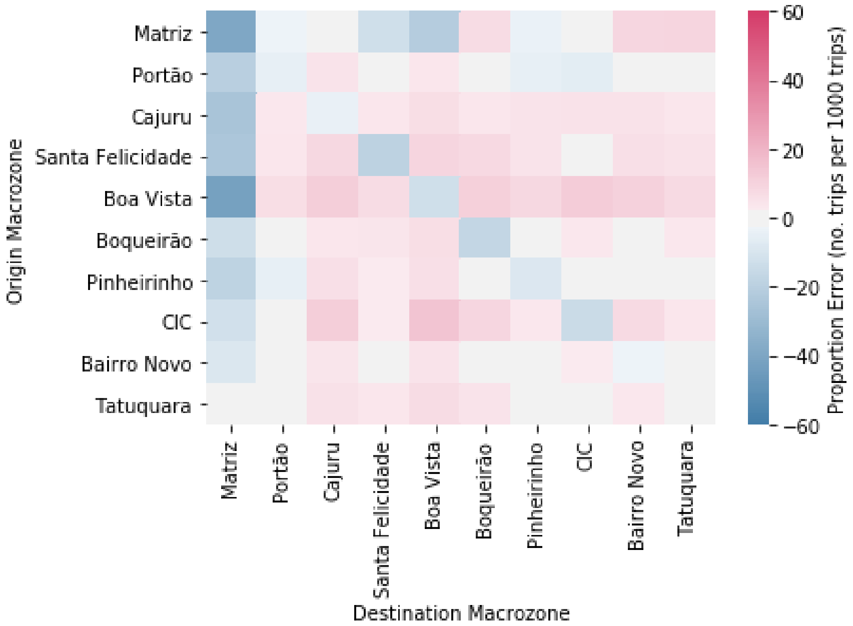


Figure 5.10: Heatmap of Comparison Error between Buste Population Proportion Baseline and Survey OD-Matrices

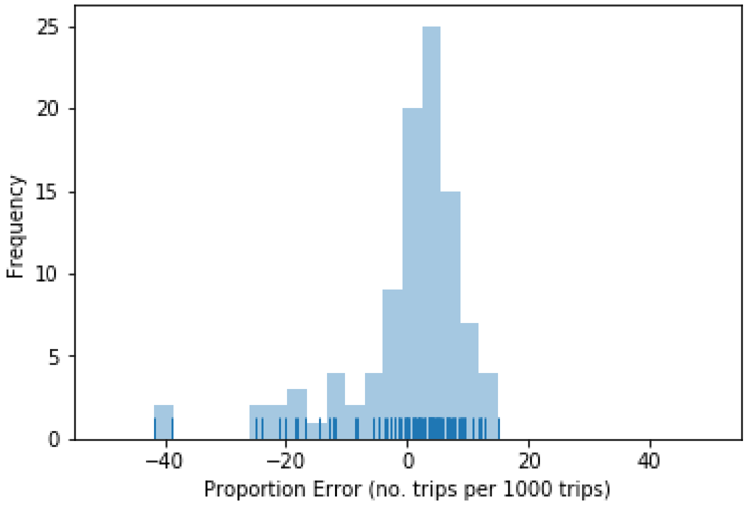


Figure 5.11: Buste Population Proportion Baseline OD-Matrix Comparison Error Distribution

5.3.3 BUSTE Origin-Proportion Baseline

A heatmap of the Error Matrix obtained when comparing the BUSTE Origin-Proportion Baseline OD-Matrix to the Survey OD-Matrix can be seen in Figure 5.12, and the error distribution histogram is shown in Figure 5.13. The MAE of the error matrix is 5.34 trips per 1000 trips in the data set.

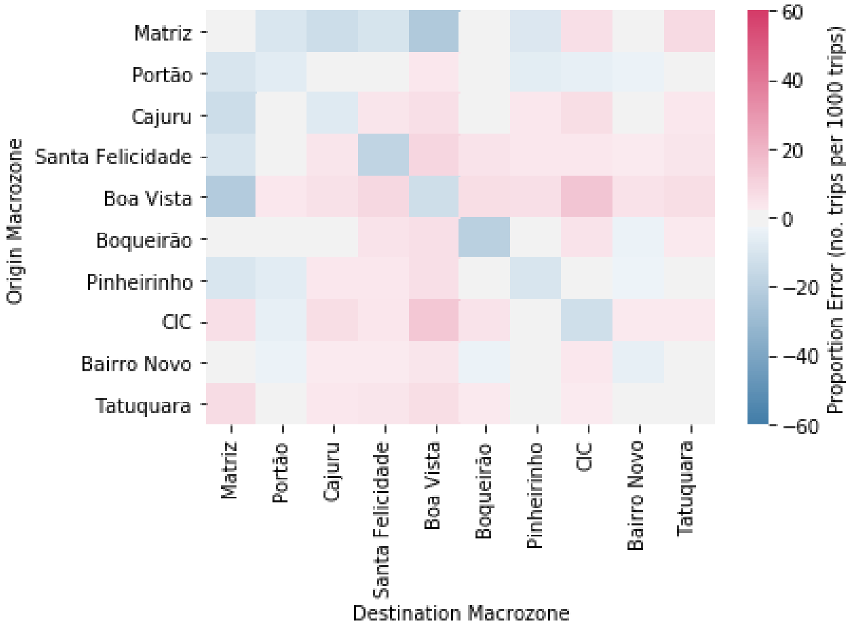


Figure 5.12: Heatmap of Comparison Error between Buste Origin Proportion Baseline and Survey OD-Matrices

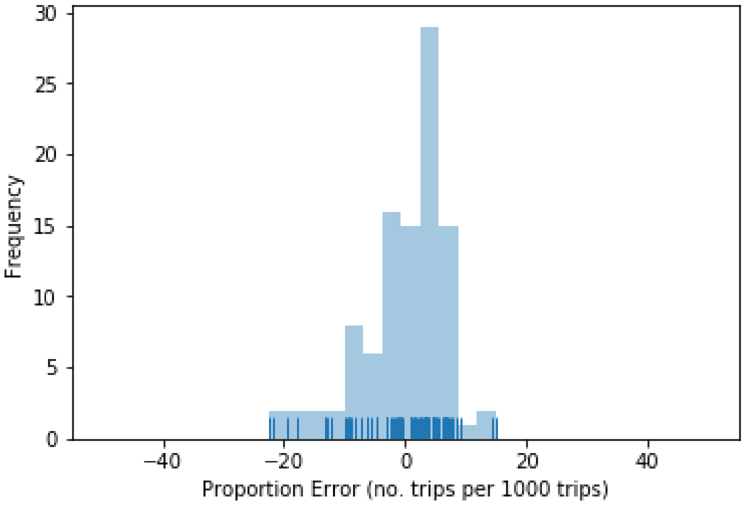


Figure 5.13: Buste Origin Proportion Baseline OD-Matrix Comparison Error Distribution

5.3.4 Inferred Trip Traces

A heatmap of the Matrix obtained when comparing the Inferred Trip Traces OD-Matrix to the Survey OD-Matrix can be seen in Figure 5.14, and the error distribution histogram is shown in Figure 5.15. The MAE of the error matrix is 3.18 trips per 1000 trips in the data set.

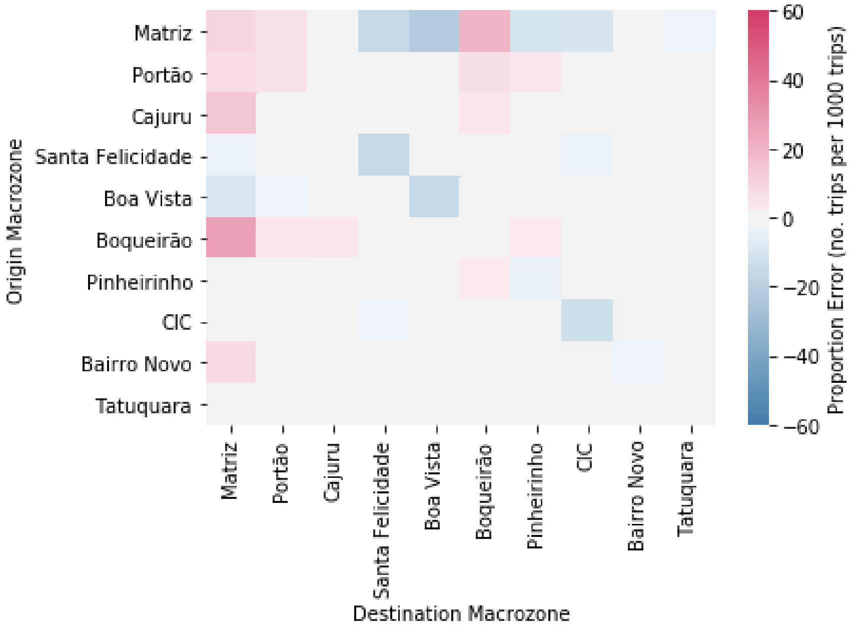


Figure 5.14: Heatmap of Comparison Error between Inferred Trips and Survey OD-Matrices

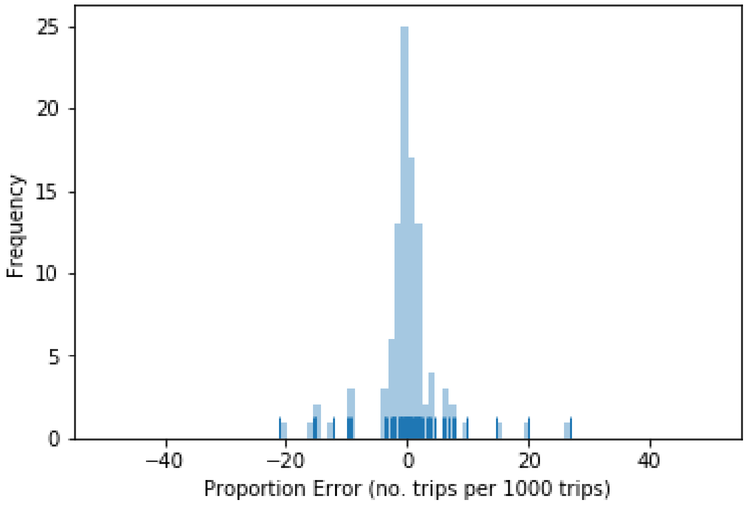


Figure 5.15: Inferred Trips OD-Matrix Comparison Error Distribution

5.4 Analysis of Results

As we can see, our method was able to reproduce the same results of the OD Survey, with an error of 3.18 trips per 1000 trips in the data set, being a reasonable option to create a profile of the Public Transportation System in a city. This error is 2.7 times lower than the error of the Naive Baseline, 2.2 times lower than the error of the Population-Proportion BUSTE Baseline, and 1.67 times lower than the error of the Origin-Proportion BUSTE Baseline. This result was achieved even with the loss of data during the process due to lack of confidence in the intermediate steps.

When analyzing the error distributions, we see that Naive and Population Proportion BUSTE Baselines have more spread error distributions, especially due to underestimation of trips. The Origin Proportion BUSTE Baseline has a more condensed distribution, concentrated around 0, with errors ranging from -22 to 14 trips per 1000 trips in the dataset. However, it still has many small errors, having 80% of the errors within the interval [-9.5,6.7], lifting the overall error up. The Inferred Trips error distribution is more condensed than the two first baselines, but has a slightly longer tail to the positive side (overestimation), with errors ranging from -21 to 26 trips per 1000 trips in the dataset. In spite of that, the errors are more tightly concentrated around 0, with 80% of the errors within the interval [-3.5,4.0], pulling the overall error down.

We have tested generating the OD Matrix from a few variations of the Inferred Trips data set, such as: only considering weekdays (monday through friday), and only considering the month of may (the month for which we have the largest amount of data). Their accuracy was no significantly different than the one using the whole data set.

Yet another analysis was performed on the Inferred Trips OD Matrix, in which we compute the MAE for the trips starting at each Macrozone, as shown in Figure 5.16. As we can see, the higher error is observed in the macrozone Matriz, followed by Boa Vista, Boqueirão and Santa Felicidade. If we consider the Median Error distribution per Macrozone, depicted in Figure 5.17, we see that most of the time, we overestimate the number of trips starting at macrozones Matriz and Boqueirão, and underestimate the number of trips starting at Macrozones Boa Vista and Santa Felicidade.

In addition, our method i) is cheaper than the survey, as it does not need to employ

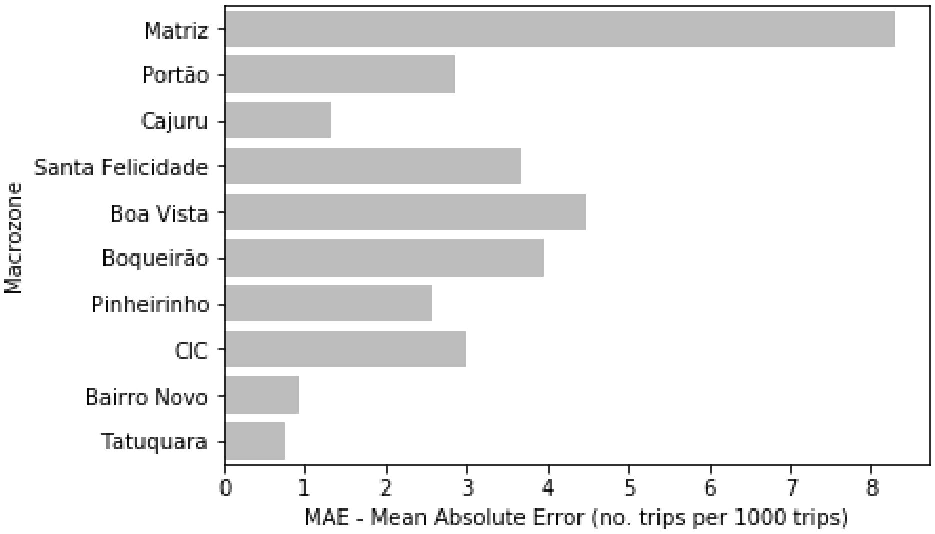


Figure 5.16: Inferred Trips OD-Matrix Comparison MAE By Macrozone

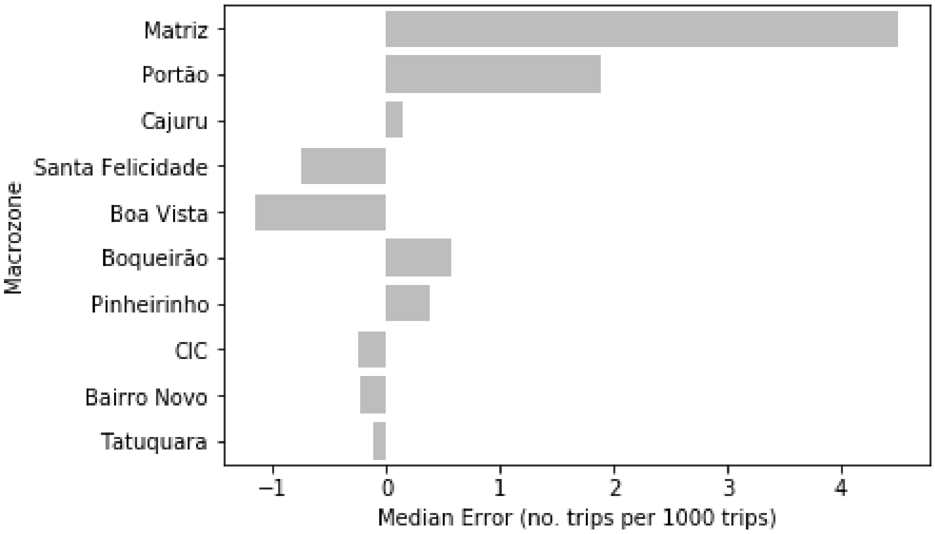


Figure 5.17: Inferred Trips OD-Matrix Comparison Median Error By Macrozone

interviewers to talk to citizens, ii) is faster, as it ran in a few days for a period of 3 months of data and is also highly parallelizable, being able to run as fast as there are computer resources to run it, iii) extends to a broader sample, as it takes all smart ticket boarding records in the city, and iv) can provide deeper and more specific analysis, going down to the level of the passenger trip, for instance.

An unexpected but relevant result of the OD Matrix validation is that the Naive Population Proportion Baseline, which uses solely the population proportion of the macrozone to estimate the OD Matrix, does not perform too poorly (MAE = 8.57 trips per 1000 trips), thus being an inexpensive option for transit planners with budgetary restrictions. However, it is important to keep in mind that this holds true for the case of Curitiba, and for the Macrozones OD Matrix, and not to apply the method thoughtlessly.

Chapter 6

Inefficiency Analysis

In the current chapter, we describe an Inefficiency Analysis experiment, whose goal is to demonstrate a city-wide analysis that is enabled by the passenger trip inference proposed in this work. The experiment uses trips inferred from the original data to evaluate how efficient are the itinerary choices performed by passengers using Public Transportation data.

Commuters who live in large cities around the world must make several decisions about their commuting each day: whether to pick route A or B to go from home to work, whether to leave a few minutes earlier from home or stay a few minutes more at work to get less transit, to take bus A which is at the stop right now but takes longer to get to destination or to wait for bus B which goes faster, and others. These decisions are often made based on their experience with the Transportation System or using transit apps, such as Google Maps[1], Moovit[2] or Here[3]. Much has been invested in research and development of solutions to assist these individuals in their daily choices. However, remains challenge to assess at the city scale whether the choice of passengers is presently providing passengers with the most efficient trips they can make.

The analysis shown in this chapter consists in examining passenger itinerary choice, and checking whether there was a more efficient itinerary he could have taken at that moment. Using the Inferred Trip Traces and Bus Trips data sets, whose construction and structure is explained in Chapter 4, as well as the city's GTFS feeds, we compute inefficiency metrics

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

[2] https://moovitapp.com/

[3] https://mobile.here.com/

and perform an evaluation of the passengers itinerary choices in the city of Curitiba.

6.1 Inefficiency Definition and Metrics

The goal of this experiment is to measure to what extent itineraries traveled by passengers in Curitiba are optimal with respect to the trip duration of alternatives available at their choice time.

More formally, at each moment a user goes from an origin to a destination, this choice is represented by (u, I), where u is the itinerary comprising one or more bus trips taken by the user, and $I = \{a_1, a_2, ..., a_n\}$ is the set of alternative itineraries available to the user for the same origin and destination at that moment, with $u \in I$. Let $d_s(a)$ be the duration of an itinerary alternative a according to the schedule, and $d_o(a)$ the duration occurred in practice according to Bus Trips data. The itinerary with shortest duration according to the schedule is $f_s(I) = \arg\min_{a \in I} d_s(a)$, while the shortest according to historical data is $f_o(I) = \arg\min_{a \in I} d_o(a)$.

The relative inefficiency in a user choice (u, I) as it occurred is then:

$$i_o = \frac{d_o(u)}{d_o(f_o(I))}. \tag{6.1}$$

Intuitively, this metric measures how worse was a trip taken by a user, relative to the best alternative that occurred: an inefficiency of 2 means that there was an alternative trip which could have been completed in half the time of that taken by the user. This inefficiency is the compound effect of suboptimal decisions taken by transit users or by deviations between schedule and operation by the bus system. For example, traffic may cause a delay in a bus which according to schedule would provide the fastest trip between two points, and render an alternative itinerary the best choice.

In addition to i_o, we use two other measures of inefficiency for a given choice situation. To isolate the inefficiency that would be experienced if the user followed the best choice according to schedule, we define the system inefficiency as:

$$i_s = \frac{d_o(f_s(I))}{d_o(f_o(I))}. \tag{6.2}$$

This metric is the ratio between the duration of the itinerary that should be the shortest according to the schedule and the itinerary that was the shortest among the alternatives.

Finally, we define the inefficiency according to the schedule as:

$$i_c = \frac{d_s(u)}{d_s(f_s(I))} \tag{6.3}$$

When $i_c = 1$, the itinerary chosen by the user was the best according to the schedule, and any inefficiency experienced is due to the system ($i_o = i_s$).

For simplicity, whenever we henceforth mention a itinerary that is shorter than an alternative, we mean that the former had a shorter duration than the latter.

6.2 Data Preparation

It follows from the previous definitions that to evaluate the inefficiency in an itinerary u chosen by a passenger in a situation (u, I), it is necessary for us to estimate $f_o(I)$. Searching for the itinerary with the shortest duration among all possible itineraries formed by combinations of bus trips and connections is an intractable task due to the combinatorial space it must traverse. We circumvent such difficulty by approximating $f_o(I)$ in a two-stage search: a first step finds the 10 itinerary alternatives $a_1, a_2, ..., a_{10}$ (when available) with the shortest expected itinerary duration between the origin and destination given the routes and schedule from GTFS. The second step uses the Bus Trips data set to search which itinerary in the set $I = \{a_1, a_2, ..., a_{10}\} \cup \{u\}$ had the shortest duration according to historical data.

Once again, we use Open Trip Planner (OTP), this time to search for $a_1, a_2, ..., a_{10}$. To implement the possibility of finding alternatives around the start time t of u, we consider alternative itineraries a_i starting in the interval $[t, t + 15mins]$. Such alternative itineraries are composed of one or more legs, whose transport mode can be either walking or transit. Thus, the itinerary duration is obtained by adding the bus legs duration found in historical data (BUSTE) to the estimated walking time. It is possible that an alternative itinerary is not found in the BUSTE data either because of our heuristics or because a bus trip did not occur. Only situations (u, I) where $|I| \geq 2$ are considered in our analysis.

A source of imprecision in our data is diminished by restricting our analysis to situations where both $d_o(u)$ and $d_o(f_o(I))$ are at least 10 minutes. The imprecision could be caused by situations where OTP proposes walking-only itineraries that substantially diverge from $d_o(f_o(I))$. Finally, we also only analyze situations where $d_o(u) \leq 120$ minutes (2 hours).

For this analysis, only vehicle-boarding events were considered (no terminal or bus station boarding), as the inferred traces for such trips are more trustworthy given the availability of route information in the first itinerary leg. We also filtered the data to comprise only weekdays (Monday to Friday), and remove from I the itineraries that lost bus legs along the processing (the scheduled itinerary/bus leg could not be matched to observed Bus Trips data). This sums up to over 4.6 million trips in 45 days, as described in Table 6.1.

Year	Month	Num. Days	Days Available
2017	May	21	(05/01-03,04,08-12,15-19,22-26,29-30)
2017	June	13	(06/14-16,19-23,26-30)
2017	July	11	(06/03-07,10-14,17)

Table 6.1: Number of days available per month considered for Inefficiency Analysis

At the end of the pre-processing step described above, we have two datasets: User Executed Trips Itineraries, which holds information for each inferred passenger trip itinerary; and Observed Schedule Suggested Itineraries, which holds information for each trip itinerary suggested by OTP and its match to historical data. For each of them, we keep the scheduled and observed duration.

Finaly, for each pair (u,I), we identify $f_s(I)$ - the fastest itinerary according to schedule; and $f_o(I)$ - the fastest itinerary actually observed, and use them to compute the inefficiency metrics i_o, i_s, and i_c.

6.3 Analysis Results

6.3.1 Overall Choice Efficiency

Figure 6.1 displays the distribution of relative inefficiencies i_o in the 1.2 million user itinerary choices in our data. It comprises both system and user inefficiencies. Overall, the observed inefficiency is low, but still short itineraries incur in lower inefficiencies compared to longer ones. In 75% of the situations where a user chose an itinerary for a 10-20 min travel, the user took the fastest possible itinerary ($i_o = 1$). In contrast, for 50+ min itineraries, one fourth of the time there was an inefficiency of at least 1.2. If $i_o = 1.2$ in a 50 min itinerary, this means the user traveled for 50 minutes while there was an alternative that could be taken at that moment and would lead to a 40 minutes trip. It is also apparent that for users traveling for 50 minutes or less, in most of the cases, the optimal option available is taken, whereas for longer itineraries, the majority of users could have taken a better itinerary.

The distribution of inefficiency values observed in the system operation irrespective of the user choice i_s is shown in Figure 6.2. Following the same trend as i_o, the itinerary expected to run faster according to schedule is outperformed by some other option most often when users are making longer itineraries. For all itinerary durations, in at least 47% of the choice situations, the shortest itinerary according to the schedule was the shortest executed ($i_s = 1$). For 10-20 min itineraries, this percentage amounts to 71%. This indicates that users who choose their itinerary based on the schedule will actually make the best choice in the case of short trips (lasting up to 30 minutes). In addition, longer trips may have more bus transfers, what may lead to greater inefficiency.

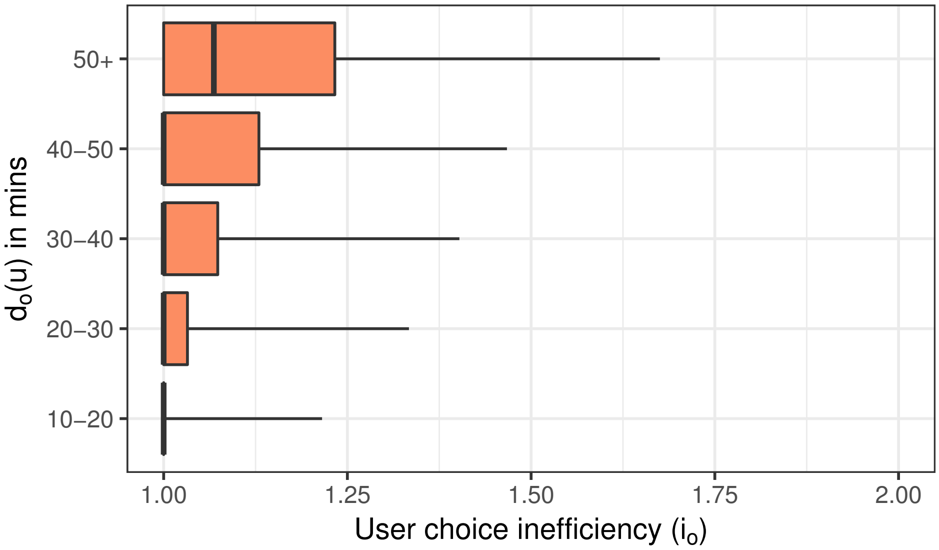


Figure 6.1: Distribution of experienced inefficiency i_o for all itineraries in the dataset according to their observed duration $d_o(u)$. Whiskers in the boxplot show 5th and 95th percentiles.

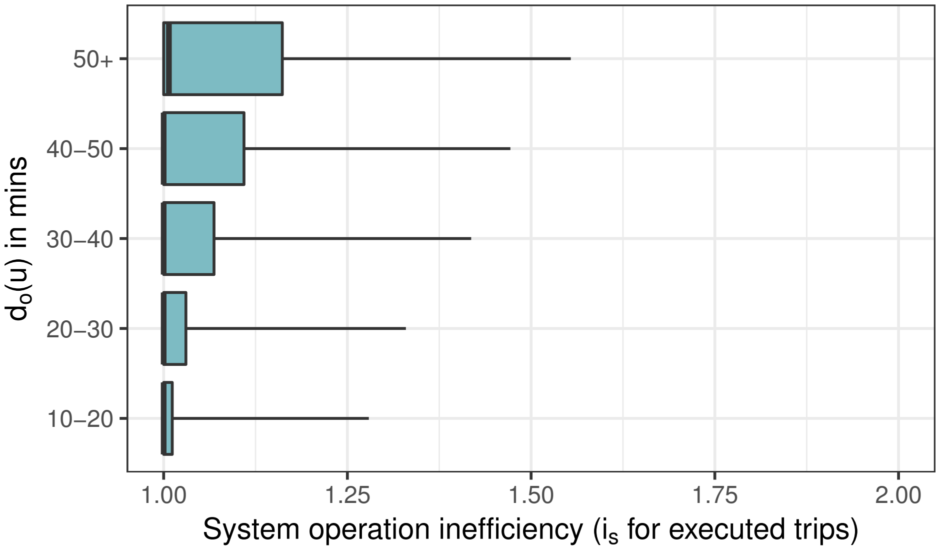


Figure 6.2: Distribution of system operation inefficiency i_s for all itineraries in the dataset according to their observed duration $d_o(u)$. Whiskers in the boxplot show 5th and 95th percentiles.

6.3.2 User Choice and the Schedule

It is possible to further examine how often users choose the itinerary that would be the shortest according to the schedule. In this case, $u = f_s(I)$ and thus $i_c = 1$. This is a situation that happens in 50-75$\%$ of the choice situations in our data. It is also worthwhile mentioning that when the user chooses the itinerary scheduled to perform shortest, $i_o = i_s$, all resulting inefficiency happens due to system operation. Figure 6.3 shows the inefficiency of these itineraries. In general, users experience little inefficiency when they take the itinerary most recommended by the schedule. In such cases, for 45 to 85$\%$ of the itineraries, $i_o = 1$.

Focusing on the 25-50$\%$ of the situations where users deviate from the itinerary prescribed by the schedule, there are three possible situations we can discern:

1. The chosen itinerary u led to no experienced inefficiency and $i_o = 1$ while that prescribed by the schedule had a non-negligible (at least 10$\%$) system inefficiency ($i_s > 1.1$);

2. The chosen itinerary u led to some experienced inefficiency ($i_o > 1$) while the one prescribed by the schedule was the best alternative ($i_s = 1$);

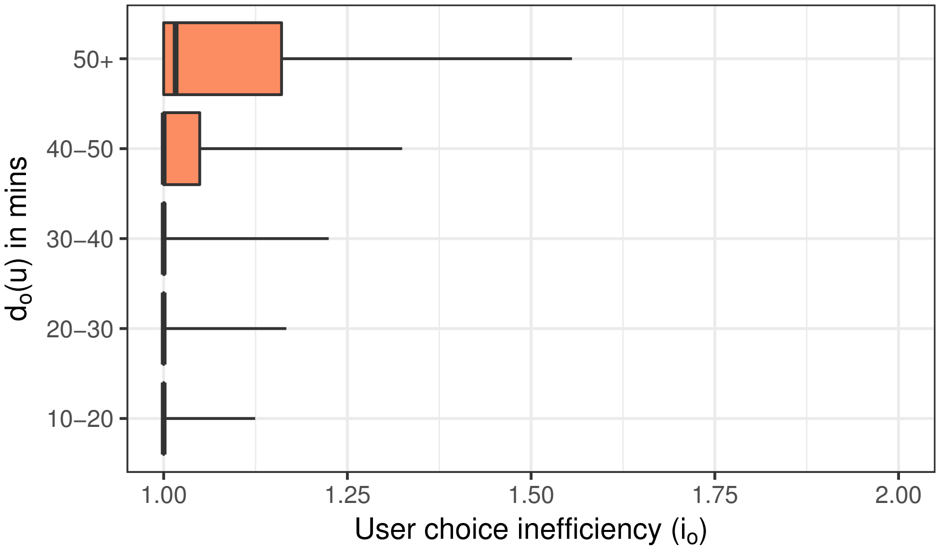


Figure 6.3: Distribution of experienced inefficiency i_o when users took the itinerary that should be the shortest according to schedule ($i_c = 1$). Whiskers show 5th and 95th percentiles.

3. Both i_o and i_s are greater than 1.1 and there was a non-negligible experienced inefficiency that is a compound of user and system inefficiencies we cannot discern.

Figure 6.4 displays the proportion of choice situations where the user deviated from the best choice according to the schedule that happen as described in cases *a* (deviation was better), *b* (schedule was better) and *c* (compound). Interestingly, not seldom (15-39%) users deviate from the schedule prescription and end up having the shortest travel time. For shorter itineraries, this happens more often. For 10-20 min itineraries it paid off to deviate from schedule in 40% of the observed choice situations where users did not follow the schedule. In the case of 50+ itineraries, deviation from schedule is usually not a good choice. This might also be due to the greater occurrence of transfers in longer trips, making it harder for the user to consider all the variables involved to make an efficient decision. Moreover, the Compound inefficiency is well represented (between 40 and 44%) among all trip lengths. A deeper analysis of such cases might bring some good insights about Curitiba's Transportation System in addition to helping the passengers make better decisions.

Taken together, our results point that there is limited room for significantly improving the travel time of users of shorter trips in the bus system of Curitiba. For longer itineraries, it

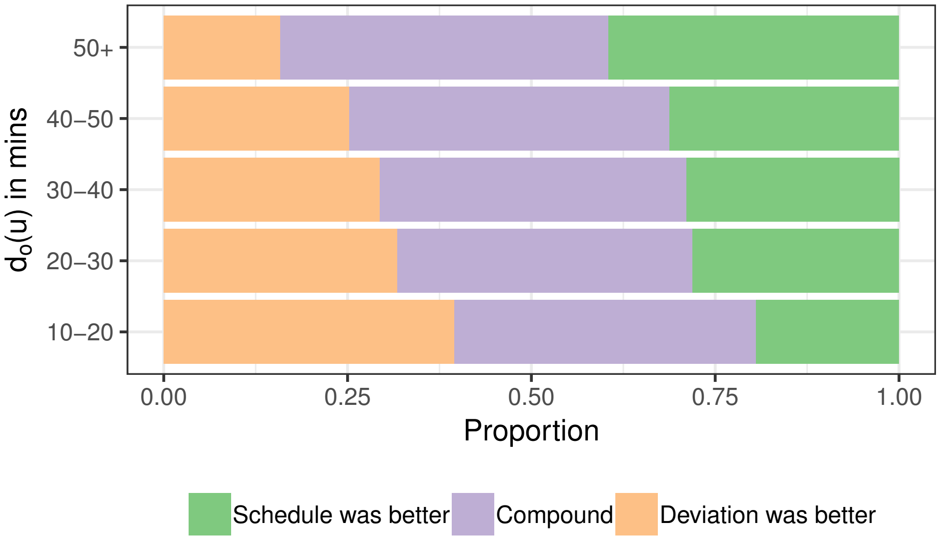


Figure 6.4: Number of trips in each trip length bin that are optimal ($i = 0$) or not ($i > 0$) according to our analysis.

happens often that travel time could be reasonably reduced if users took alternative options. These are indeed the situations where the system most diverges from the schedule, what suggests that helping the user decide would have an impact. Complementing these observations, our analysis points that users most often choose optimally according to the schedule. Even then, longer itineraries experience more inefficiency. Finally, in the situations where users deviate from schedule advice, it often happens that user choice is better than such advice, and the Compound (unexplained) inefficiency seems to be very prevalent in most scenarios, demonstrating the necessity for a deeper analysis of those cases.

Chapter 7

Discussion

This study proposed methods to integrate Public Transportation data sources, namely Schedule (GTFS, Positioning (AVL) and Ticketing (AFC), with the aim of inferring passenger trip traces for each boarding record present in the data. Having built the Inferred Passenger Trips data set, we designed and implemented two analysis which are now made possible: an estimation of an Origin-Destination Matrix from the inferred trips (validating with OD Survey results), and an analysis of user itinerary choice (in)efficiency based on the inferred trips and their respective itinerary alternatives. This chapter describes the global conclusions drawn from the research process, and points out room for improvement and future work ideas.

7.1 Implications

In order to carry out analyses on the Public Transportation passenger trips, a number of non-trivial, non-apparent and important steps need to be performed to integrate the different data sources, which were designed and operate separately and in most cases have no direct link to each other. This aspect reduces the generalizability of the integration process, as different cities will have data stored and organized in different ways, and also structure their transportation systems differently. Therefore, a method developed for a city should not be blindly applied to another. In the case of Curitiba, there are the tube stations, which serve as small terminals for a few bus routes, and are spread all over the city, making station boarding records account for about half of the total records.

As could be seen in Chapter 5, it is possible to estimate an Origin-Destination Matrix using the Inferred Trips data set. The comparison with 2016's Curitiba OD Survey showed that our estimated OD Matrix can accurately reproduce the Survey OD Matrix for the Public Transportation System, with an MAE of 3 trips per 1000 trips. This is an important result, demonstrating the ability of the developed method in generalizing the results when the original data set is not complete, as the smart card boardings correspond to 60% of the boardings in Curitiba. In addition, our OD-Matrix construction method has the following advantages: much lower cost; takes less time and human effort to be executed; can perform a system/city-wide analysis (not just using samples); can be performed continuously, providing a more accurate view of the system status; and is able to deliver a fine-grained analysis, going down to the aggregation level of the trip itinerary leg.

Another important outcome of this work is the use a routing engine, in our case OTP, to search for possible itineraries between points during the destination estimation process. As a routing engine developed on top of state-of-the-art path search algorithms, we can be sure to retrieve the best suggestions according to the system schedule, including itinerary alternatives with bus transfers. The fact that OTP is open-source enabled us to have our own deployment of the server with enough replicas to process our experiment queries on a reasonable time.

The Inefficiency Analysis reveals the overall observed inefficiency of user choices is low, but increases with the trip duration, meaning that passengers usually make the best itinerary choice (in terms of trip duration). The same applies for the system operation innefficiency, demonstrating the schedule reflects the reality of bus operation in terms of itinerary ranking. When analyzing the deviation from schedule, we observe that a reasonable amount of trips experience an unexplained innefficiency (neither the user chosen itinerary nor the schedule recommendation was the best choice), which suggests there is some room for improvement in such cases, and a deeper analysis can be carried out to understand and explain the scenario.

7.2 Limitations

There are also some limitations of the study which are worth being cited: a) the reference table to match the terminal boarding records smart card reader machine code to the bus station they are located at does not comprise all reader machines found in the data, hence we lost some records in this matching step; b) the lack of completeness of data between the different data sources, for instance: not all routes described in the GTFS feeds can be found in AVL data (one possible reason for that is the bus vehicles for such route are not GPS-equipped), and so this also contributes to the loss of data throughout the analysis; and c) the fact that Curitiba's Transport System has a feature called "Temporal Hub" (freely translated from Portuguese), which allows a passenger to make a bus transfer outside terminals without incurring into a new fare payment, given a time limit between the two boarding transactions, which is usually 2 hours. Because of this feature, some boarding records we considered to be new trips are actually just transfers of the same trip. This could be mitigated by applying a time difference threshold between origin and next orign boardings.

7.3 Future Work

The methods developed in this research proved to be effective in inferring passenger trip itineraries. However, a lot can be done to improve the methods results. A first step would be to run a parameter tunning experiment to select the best values for each parameter used accross the methods steps. The Step 5 from Section 4.2.2, which chooses the best leg matches for each step of the itinerary can also be optimized, making the whole process run faster. An important step would be to perform an error analysis of the results of Chapter 5 and check for patterns in the errors which can indicate bugs or points for improvement in the script. Another possible improvement is to include the single-transaction users in the analysis, by conducting a journey regularity analysis as has been done in other studies [1]. We can also analyze different cuts of the OD Matrix: per time of day, weekday vs. weekend; and generate OD Matrices for different geographic aggregation levels, such as: neighborhoods, zones, census tract/sector.

Moreover, the results of this study enable a variety of applications and analysis which demand large integrated public transportation data sets. One such application, already cited here, which seems promissing, is the bus crowding estimation, which can be performed on top of the Inferred Trip Traces data set. It is possible to estimate crowding for historical data, being able to analyze the bus crowding per route/region of the city along the time; and also train a machine-learning model to predict the crowding of a bus within a stretch of its trip.

Such model can be used to predict crowding for passenger itinerary alternatives in a transit app, for instance.

Bibliography

[1] James 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(02):183–187, 2002.

[2] T. Braz, M. Maciel, D.G. Mestre, N. Andrade, C.E. Pires, A.R. Queiroz, and V.B. Santos. Estimating Inefficiency in Bus Trip Choices from a User Perspective with Schedule, Positioning, and Ticketing Data. *IEEE Transactions on Intelligent Transportation Systems*, 19(11), 2018.

[3] Demographia. Demographia world urban areas, 2016.

[4] Paulo Diniz. Servicos telematicos em uma rede de transporte publico baseados em veiculos conectados e dados abertos. Master's thesis, Universidade Tecnologica Federal do Parana (UTFPR), Curitiba, Brazil, 2017.

[5] Dennis Dreier, Semida Silveira, Dilip Khatiwada, Keiko V.O. Fonseca, Rafael Nieweglowski, and Renan Schepanski. Well-to-Wheel analysis of fossil energy use and greenhouse gas emissions for conventional, hybrid-electric and plug-in hybrid-electric city buses in the BRT system in Curitiba, Brazil. *Transportation Research Part D: Transport and Environment*, 58:122–138, jan 2018.

[6] United Nations Population Fund. United nations population fund - urbanization, 2016.

[7] Antonio Gschwender, Marcela Munizaga, and Carolina Simonetti. Using smart card and GPS data for policy and planning: The case of Transantiago. *Research in Transportation Economics*, 2016.

[8] Björn Johnson. Cities, systems of innovation and economic development. *Innovation*, 10(2-3):146–155, 2008.

[9] Le-Minh Kieu, Ashish Bhaskar, and Edward Chung. A modified Density-Based Scanning Algorithm with Noise for spatial travel pattern analysis from Smart Card AFC data. *Transportation Research Part C: Emerging Technologies*, 58:193–207, 2015.

[10] Herbert Levinson, Samuel Zimmerman, Jennifer Clinger, and G. Rutherford. Bus Rapid Transit: An Overview. *Journal of Public Transportation*, 2002.

[11] Yangxin Lin, Ping Wang, and Meng Ma. Intelligent Transportation System(ITS): Concept, Challenge and Opportunity. *Proceedings - 3rd IEEE International Conference on Big Data Security on Cloud, BigDataSecurity 2017, 3rd IEEE International Conference on High Performance and Smart Computing, HPSC 2017 and 2nd IEEE International Conference on Intelligent Data and Securit*, pages 167–172, 2017.

[12] Demetrio Gomes Mestre. *Leveraging the entity matching performance through adaptive indexing and efficient parallelization.* PhD thesis, Universidade Federal de Campina Grande (UFCG), 2018.

[13] Moovit. Global cities public transit usage report, 2016.

[14] 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.

[15] Antonio A. Nunes, Teresa Galvao Dias, and Joao Falcao e Cunha. Passenger Journey Destination Estimation From Automated Fare Collection System Data Using Spatial Validation. *IEEE Transactions on Intelligent Transportation Systems*, 17(1):133–142, jan 2016.

[16] Mirai Tanaka, Takuya Kimata, and Takeshi Arai. Estimation of Passenger Origin-Destination Matrices and Efficiency Evaluation of Public Transportation. In *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, pages 1146–1150. IEEE, jul 2016.

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

[18] Chi Tung Tung and Kim Lin Chew. A multicriteria Pareto-optimal path algorithm. *European Jour. of Oper. Research*, 62(2), 1992.

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

[20] Menno Yap, S Nijenstein, and Niels Oort. Improving predictions of public transport usage during disturbances based on smart card data. *Transport Policy*, 61, 2018.

[21] J. Zhao, Q. Qu, F. Zhang, C. Xu, and S. Liu. Spatio-temporal analysis of passenger travel patterns in massive smart card data. *IEEE Transactions on Intelligent Transportation Systems*, 18(11):3135–3146, Nov 2017.

[22] Jinhua Zhao, Adam Rahbee, and Nigel H. M. Wilson. Estimating a Rail Passenger Trip Origin[U+2010]Destination Matrix Using Automatic Data Collection Systems. *Comp.-Aided Civil and Infrastruct. Engineering*, 22:376–387, 2007.