Title
On learning Game-Theoretical models with Application to Urban Mobility

Author
Thai, Jerome

Publication Date
2017

Peer reviewed|Thesis/dissertation

On learning Game-Theoretical models with Application to Urban Mobility

by

Jérôme Thai

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Alexandre Bayen, Chair
Professor Laurent El Ghaoui
Professor Alexei Pozdnoukhov

Fall 2017

On learning Game-Theoretical models with Application to Urban Mobility

Abstract

On learning Game-Theoretical models with Application to Urban Mobility

by

Jérôme Thai

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Alexandre Bayen, Chair

Modeling real-world processes as convex optimization or variational inequality problems is a common practice as it enables to leverage powerful mathematical tools for the study of such processes. For example, in economics, knowing the consumer utility function enables to adjust prices to achieve some demand level. In control, a low complexity controller requires less computation for little performance loss. In transportation science, the selfish behavior of agents (from shorted path routing) leads to an aggregate cost in the network worse than the system's optimum, and which can be analytically quantified. Taxation schemes can be designed to incentivize system optimal drivers' decisions.

In the first part of our work, we briefly review fundamental results in convex optimization, variational inequality theory, and game theory. We also focus on the selfish routing game, which is a popular game-theoretical framework to model the urban transportation network. In particular, we study the impact of the increasing penetration of routing apps on road usage. Its conclusions apply both to manned vehicles in which human drivers follow app directions, and unmanned vehicles following shortest path algorithms. To address the problem caused by the increased usage of routing apps, we model two distinct classes of users, one having limited knowledge of low-capacity road links. This approach is in sharp contrast with some previous studies assuming that each user has full knowledge of the network and optimizes his/her own travel time. We show that the increased usage of GPS routing provides a lot of benefits on the road network of Los Angeles, such as decrease in average travel times and total vehicle miles traveled. However, this global increased efficiency in urban mobility has negative impacts as well, which are not addressed by the scientific community: increase in traffic in cities bordering highway from users taking local routes to avoid congestion.

In the second part, we explore the ability of low complexity game-theoretical models to accurately approximate real transportation systems. For example, system mischaracterizations in selfish routing can cause taxes designed for one problem instance to incentivize inefficient behavior on different, yet closely-related instances. Hence, we want to be able to measure the quality of the learned model. In the present work, we present a statistical framework for the fitting of equilibrium models based on measurements of edge flows using the (standard)

empirical risk minimization principle, by choosing the fit giving the lowest expected loss (the distance between the observed and predicted outputs) under the empirical measure. Hence, for the class of models of interest, it is critical to be able to have theoretical guarantees on the quality of the fit. We then present a computational methodology for imputing the map of an equilibrium model, and propose a statistical hypothesis test for validating the trained model against the true one.

In the third part, we explore existing work for estimating link and route flows, and we propose two novel frameworks for traffic estimation. In the first framework, we focus on estimating the highway traffic, which is modeled as a discretized hyperbolic scalar partial differential equations. The system is written as a switching dynamical system, with a state space partitioned into an exponential number of polyhedra in which one mode is active. We propose a feasible approach based on the interactive multiple model (IMM), and apply the k-means algorithm on historical data to partition modes into clusters, thus reducing the number of modes. In the second framework, we develop a convex optimization methodology for the route flow estimation problem from the fusion of vehicle count and cellular network data. The proposed approach is versatile: it is compatible with other data sources, and it is model agnostic and thus compatible with user equilibrium, system-optimum, Stackelberg concepts, and other models. The framework is validated on the I-210 corridor near Los Angeles, where we achieve 90% route flow accuracy with 1033 traffic sensors and 1000 cellular towers covering a large network of highways and arterials with more than 20,000 links.

To Evelyne, Alphonse, and Joël

Contents

List of Figures

List of Tables

Acknowledgments

Doing a Ph.D. has been a real roller coaster for me, both on a professional and on a personal level. I felt like what seems to be a much longer period of my life has been condensed in just five years. These years have been filled with both rewarding and difficult moments, resulting in a very enriching and fulfilling experience. During these years, I also had the chance to meet amazing people, and I apologize in advance to anyone whom I neglected to mention.

How did this odyssey started ?

With two emails to my advisor, Prof. Alex Bayen. I sent the first one back in 2010, when I was looking for a research internship required to complete my curriculum at Ecole Polytechnique. This led me to work in the supervision of Mohammad, a former PhD student in Bayen's lab, on the Floating Sensor Network project. Our collaboration led to the publication of a conference paper while I was completing my Master's degree at Columbia University. As I was preparing my application for Ph.D. programs in the Fall of 2011, another email to Alex during an impromptu visit to Berkeley (I was still living in NYC), reconnected me with him. Based on my positive experience with the lab, he convinced me to apply to the PhD program in the EECS department at UC Berkeley and return to his group, which I ultimately did.

I firmly believe that life is mostly determined by the people you meet along the way. Alexandre Bayen convinced me to embark on this wonderful adventure. His energy, enthusiasm, and encouragements had a profound and positive impact on what I value, what I believe is possible, and how I decide to pursue it. He helped me to navigate through graduate school and produce great work. I will be forever grateful for his mentorship. Then, I would like to thank Mohammad Rafiee, with whom I collaborated during my research internship. Without his energy and drive, I would not have been able to publish an article during my research internship, and may not have returned to Berkeley for the PhD program.

I would also like to thank the faculty I collaborated with during my time at Berkeley. With their constant support, they contributed to make Berkeley a great environment to learn from the best. In particular, many thanks to Claire Tomlin for her two phenomenal classes in control theory: Linear System Theory and Nonlinear Systems. I am also grateful that Claire welcomed me to her lab meetings and I thoroughly enjoyed learning more about the research of her PhD students and postdoctoral students. Thank you Laurent El Ghaoui for taking me as his teacher assistant for his Convex Optimization class. I still fondly remember our discussions on convex optimization. Laurent's advise were extremely helpful during the first two years of my PhD. I would also like to thank Jean Walrand for letting me be a teacher assistant for his class: Random Processes in Systems. I had a great time teaching the recitation sessions and learned a lot through the process. I also really enjoyed brainstorming with Jean on the pricing and the dispatching of rides in peer-to-peer transportation. Jean's suggestions to adopt a machine learning approach on these topics were instrumental to the success of my internship at Lyft. Many thanks to Alexei Pozdnoukhov for introducing me

to machine learning methods applied to transportation systems and urban networks. I am also grateful that I had the chance to attend the Theoretical Statistics class by Martin Wainwright and Michael Jordan. I consider it as one of the best classes I took in my graduate life. This course inspired me to explore the statistical properties of the selfish routing game.

Berkeley also offered me the opportunity to collaborate with amazing students. I have fond memories spending all-nighters launching simulations, producing graphics, and writing articles with Chenyang. I enjoyed discussing with Walid on diverse topics on gradient descent algorithms for solving convex optimization programs. Timothy was also a great mentor during the first year of my PhD. He pushed me to hone my programming skills, which was instrumental to build computational frameworks for analyzing Mobility-as-a-Service systems and selfish routing games with heterogeneous users. It was exciting to collaborate with Cathy on the Megacell project. I am also grateful for her advise on getting summer internships in the Bay Area. Last but not least, I would like to thank Kene, Skander, and Rim for collaborating with me on diverse research topics and articles.

I would like to thank the friends with whom I share some great memories during my time at Berkeley. I enjoyed practicing thrilling sports with some of them: cycling up mountains, surfing, mountain biking with Charles, Jon, and Carlos, skiing and snowboarding with George and Sungmin, and going on motorcycle rides with Romain. It was great to hang out around campus, and occasionally discuss about algorithms, statistics, and preparing for job interviews with Chedly, Sara, and Aurélien.

During my last semester as a student at Berkeley, in the Fall of 2017, I enjoyed hanging out at the lab with Jesse and Kathy, and watching Black Mirror episodes and music videos with them.

I also have fond memories having lunch and dinner with Zoé, Alex, and Yves, whom I met at Safeway, as I was buying two bottles of Prosecco to celebrate my Lyft offer. It was fun to prepare desserts for them, and follow their adventures as they created their startup Grape2Glass (`grape2glass.us`), with the learn2launch program at Berkeley.

My PhD experience would have been quite different without Elizabeth, a fellow PhD student in the Art History department at Berkeley, who accompanied me for five years. I have fond memories of us working until ungodly hours at my lab, and going to conferences together in places around the world, including Washington DC, Osaka, and Antwerp.

Finally, I would like to thank my parents, Evelyne and Alphonse, for accompanying me throughout my education. With their emotional and financial support, I was able to navigate successfully through the French preparatory program with two years at Lycée Louis-le-Grand and a third year at Lycée privé Sainte-Geneviève (switching to "Ginette" was one of the best decisions of my life). For my fourth year at Polytechnique, my parents also supported my decision to pursue a Master's degree at Columbia, which ultimately led me to apply to PhD programs at American universities.

Chapter 1

Introduction

The topic of this work is the study of the selfish routing game seen as a regression model encoding the relationship between the traffic demand (the explanatory variables or inputs) and the resulting equilibrium flow (the dependent variables or outputs).

The selfish routing game follows the well-known Wardrop equilibrium conditions [166]. It models agents selecting the shortest route (in terms of time or cost) on a transportation or communication network with congestion. This framework enables me to leverage convex optimization and variational inequality theory in order to derive convenient analytical results, and design efficient algorithms for computing the equilibria. Hence, selfish routing games have been studied extensively in the literature. For instance, it is known as the Wardrop equilibrium in the operations research literature and in economics [47], the traffic assignment in transportation science [131]. The routing game has been used by urban planners for many applications including estimating travel demand and designing toll strategies. In the work presented in this thesis, I tackle a topical issue, which is the impact of the use of GPS apps on traffic patterns.

Since the selfish routing game is commonly used for evaluating urban projects, it is natural to assess the viability of such a model at predicting traffic patterns. I assume that the edge cost functions are learned from N observations of traffic flows using the empirical risk minimization principle. This consists in choosing the edge cost functions from a family of candidate functions that gives the lowest expected loss under the empirical measure defined by N observations. In this work, the loss is the distance between the observation and the output predicted by the learned model. It is then useful to study the behavior of the out-of-sample loss, which is a measure of the prediction accuracy of the model. In fact, the estimation of the prediction capability lies at the heart of techniques concerned with model selection, see, *e.g.*, Chap. 7 in [78].

I also present computational techniques for estimating the edge cost functions from observations of traffic patterns. Since the equilibrium flow described by the selfish routing game can be interpreted as the solution of a convex optimization problem, these techniques belong to a more general framework of learning problems known as *inverse optimization* [85], [93], [19], [150].

The last topic of our work is the problem of estimating the traffic flow on the urban network, which I rely on to fit our trained model. While there is an abundance of literature in transportation science aiming at estimating the movement of traffic, I focus on two approaches: 1) the estimation of traffic flows on highways using a hybrid systems framework, and 2) the estimation of traffic flows on an urban network using the fusion of loop detector and cellular data.

1.1 The selfish routing game

In 1952, Wardrop stated the user equilibrium condition, also known as the Wardrop equilibrium. It states that the travel times in all paths that are used (with positive flow on it) are equal, and they are less or equal than those that would be experienced by a single vehicle on any unused route. A traffic flow satisfying the Wardrop principle is referred to as "user equilibrium" (UE) flow, since each user cannot improve its travel time by unilaterally changing its route.

Formally, the urban network is modeled as a directed graph in which edges represent road segments, and vertices represent intersections or nodes between two consecutive road segments. For each edge e, there is a cost function $c_e : \mathbb{R}_+ \to \mathbb{R}_+$ that models the relationship between the travel cost $c_e(x_e)$ and the volume of traffic x_e on thed edge e. Encoding the link flows into a vector $\mathbf{x} = (x_e)_{e\in\mathcal{E}} \in \mathbb{R}_+^{\mathcal{E}}$, the user equilibrium can be computed by solving the following optimization program [14]:

$$\min_{\mathbf{x}} \sum_e \int_0^{x_e} c_e(u)\, du \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{D} \tag{1.1}$$

where $\mathcal{D}$ is the domain of feasible link flows. The above formulation is also known as a potential game [140]. When the cost functions $c_e(\cdot)$ are continuous and non-decreasing, the above optimization program is convex. Section 2.5 in Chapter 2 explores in more details the representation of the routing game as an optimization program.

The Frank-Wolfe algorithm (a.k.a. the conditional gradient algorithm) is commonly used to solve the traffic assignment problem (1.1). With additional assumptions, the optimal solution $\mathbf{x}^\star \in \mathcal{D}$ to (1.1) is unique, and it can be shown that the iterates $\mathbf{x}_k$ of the Frank-Wolfe algorithm converges to $\mathbf{x}^\star$ in $\|\mathbf{x}^\star - \mathbf{x}_k\|_2^2 = \mathrm{O}\big(1/k\big)$, where k is the number of iterations. I refer the reader to Section 3.2 in Chapter 2 for more details on the convergence rate.

There is a heterogeneous extension of the selfish routing game which models drivers experiencing different travel costs because they may have, *e.g.*, different routing preferences. However, there is, in general, no potential formulation similar to (1.1) for the heterogeneous game. I encode the link flows into a vector $\mathbf{x} = (x_{e,t}) \in \mathbb{R}_+^{\mathcal{E}\times|T|}$, where t encodes the type of the driver and $|T|$ is the number of types of drivers. I also define the vector of edge costs $F : \mathbb{R}_+ \to \mathbb{R}_+$, where $F(\mathbf{x}) = (c_{e,t}(\sum_{t'\in[T]} x_{e,t'}))_{e,t}$, *i.e.* the cost of traveling edge e is a function of the total flow $\sum_{t'\in[T]} x_{e,t'}$ on edge e. The user equilibrium in the heterogeneous

case can be computed by finding $\mathbf{x}^\star \in \tilde{\mathcal{D}}$ such that

$$\langle F(\mathbf{x}^\star),\, \mathbf{x} - \mathbf{x}^\star \rangle \geq 0, \quad \forall\, \mathbf{x} \in \tilde{\mathcal{D}} \tag{1.2}$$

where $\tilde{\mathcal{D}}$ is the domain of feasible edge flows. The above problem is known as a variational inequality problem. I refer the reader to the work of [60], [142] for more details on the variational inequality problem. I briefly present the heterogenous game in Section 2.6 in Chapter 2. In her work [76], Hammond also suggests a variant of the Frank-Wolfe algorithm for solving (1.2), and conjectures that the proposed algorithm solves the variational inequality problem. I refer to Section 3.4 in Chapter 3 for more details on the variant of the Frank-Wolfe algorithm proposed by Hammong to solve the variation

1.2 Impact of GPS-enabled routing apps on mobility

Routing games have been extensively studied in transportation settings, see [131] and the references therein. Applications include the design of strategies, such as taxation schemes [65], [91].

In this work, I apply the discussed game-theoretical framework to study the impact of app use on traffic patterns. With the widespread access to traffic information via use of routing apps, urban and suburban areas in the US have seen a recent rise in "cut-through" traffic and related congestion patterns. This phenomenon can be spontaneous when they are caused by a natural response of routing apps to special events such as big events (concerts, games etc.) or accidents. This phenomenon can also be a trend caused by the progressive increase in traffic demand, accompanied by a shift of traffic from highways to cities bordering them. To capture this phenomenon, I take into account motorists' different access to information (full access to traffic conditions via routing apps as opposed to incomplete information in the absence of app routing). For this purpose, I use the heterogeneous routing game framework. I differentiate routed users, who follow the shortest routes (in terms of travel time) using GPS devices from non-routed users, who have limited knowledge of the road network and of current travel times, hence favor high-capacity roads for 'perceived' benefits such as safety and low travel times.

I illustrate the potential effect of navigation apps rerouting traffic to urban areas on a benchmark network in Los Angeles. It is shown in the left panel of Figure 1.1, where I consider I-210 (illustrated in red) and a couple of arterial roads along it (illustrated in yellow). Formally, I define $\mathcal{E}^{\text{lo}}$ the set of low-capacity edges (the arterial roads), and $\mathcal{E}^{\text{hi}}$ the set of high-capacity edges (the I-210 corridor). I encode the travel time on edge e with a function $t_e(x_e) \in \mathbb{R}_+$, where $x_e \in \mathbb{R}_+$ is the total flow on edge e (sum of the flows of routed users and non-routed users). The cost for non-routed users are modeled with the following cognitive cost model

$$c_e^{\text{nr}}(x_e) = \begin{cases} C\, t_e(x_e) & \text{if } e \in \mathcal{E}^{\text{lo}} \\ t_e(x_e) & \text{if } e \in \mathcal{E}^{\text{hi}} \end{cases} \tag{1.3}$$

where non-app users are made to pay a multiplicative "cognitive cost" C for accessing arterial roads. On the other hand, the cost for routed users is $c_e^{\mathrm{r}}(x_e) = t_e(x_e)$ for all $e \in \mathcal{E}$, where $\mathcal{E} := \mathcal{E}^{\mathrm{lo}} \sqcup \mathcal{E}^{\mathrm{hi}}$ is the disjoint union of low and high-capacity edges.

I apply the heterogeneous game framework to illustrate the impact of the number of app users on traffic patterns. The highway capacity is 6000 veh./hr, the OD demand is set to 20,000 veh./h., and the cognitive cost C is set to 3,000 on the arterial (or low-capacity) roads. When all users are non-routed, only the I-210 is used because there is a high cognitive cost of travelling residential roads bordering the highways. As the percentage of app users is progressively increased from 0% to 5%, travel time on the I-210 decreases rapidly, while the travel time on arterial roads remains close to the free-flow travel time because only a few app users are rerouted. Then, the travel time along path 2 and path 1 increase quickly due to the congestion effect. In Chapter 4, I study the impact of routing apps on a larger network in Los Angeles, composed of 28,376 arcs and 14,617 nodes extracted from OpenStreetMap.

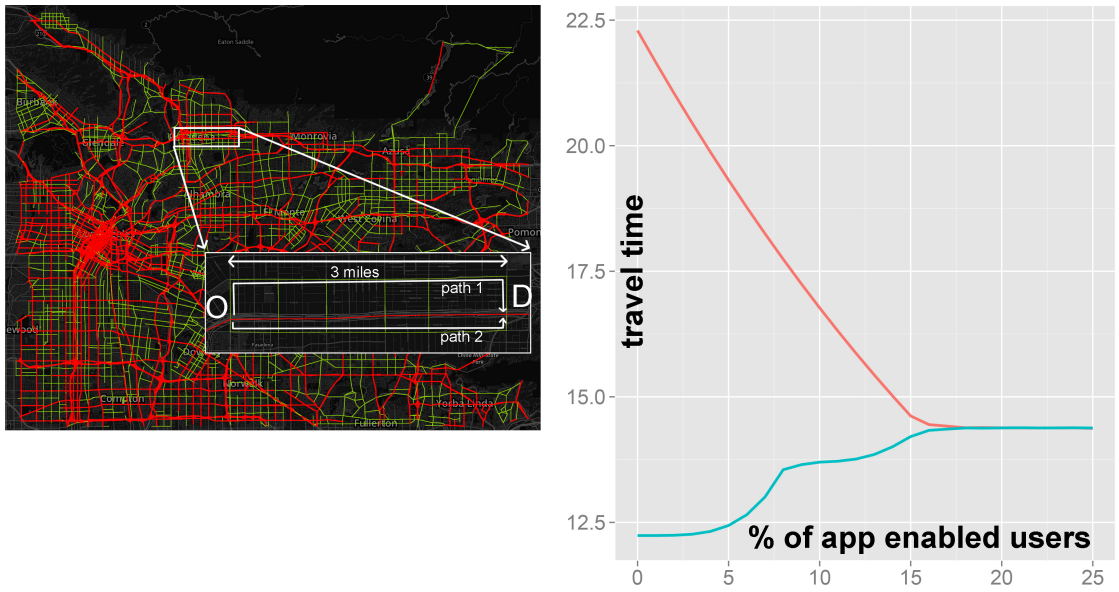

Figure 1.1: Left panel: Benchmark network along the I-210 corridor. Right panel: travel time along the I-210 in red, and travel time along the shortest path going through arterial roads in blue. Best viewed in color.

1.3 Statistics of learning the edge cost functions in selfish routing games

The selfish routing game has been commonly used for the evaluation of urban projects. However, the analysis of such models heavily relies upon having acccess to edge cost functions that yield equilibrium flows with good predictive accuracy. Estimating the edge cost functions

is difficult since they may represent some combination of the actual travel time, the tolls, and disutility from environmental factors, which are not directly observable. However, the equilibrium flows induced by the selfish routing of agents is in practice easily observable through the sensing infrastructure. This setting spurred the recent study of a class of learning problems based on estimating the edge cost functions that generate the observed equilibrium flows [85, 94, 19, 150].

Empirical risk minimization is a standard decision-theoretic framework for learning the edge cost functions. It consists in choosing the ones giving the lowest expected loss, where the loss is a measure of how much the model deviates from empirical data. Formally, I assume the vector of congestion functions $F = (c_e(\cdot))_{e\in\mathcal{E}}$ belongs to a family $\{F_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$ of strongly monotone maps from $\mathbb{R}_+^{\mathcal{E}}$ to itself, where K is the number of populations in the selfish routing game. I also suppose that the domain of feasible flows $\mathcal{D}(\mathbf{d})$ depends on the random demand vector $\mathbf{d} \in \mathbb{R}_+^K$. Then, the user equilibrium $\mathbf{x}^\star(\mathbf{d})$ is a function of the demand $\mathbf{d}$ such that, for each $\mathbf{d}$, it is the solution of the convex program

$$\min_{\mathbf{x}} \sum_e \int_0^{x_e} c_e(u)\, du \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{D}(\mathbf{d})$$

Empirical risk minimization then consists in finding the parameters $\boldsymbol{\theta} \in \Theta$ that minimize the *empirical risk*

$$R_N(\boldsymbol{\theta}) := \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}^\star(\mathbf{d}_i) - \mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d}_i)\|$$

where $\{\mathbf{d}_1, \cdots, \mathbf{d}_N\}$ is a family of N samples of the demand vector $\mathbf{d}$, and $\{\mathbf{x}^\star(\mathbf{d}_i)\}_{i\in[N]}$ are the N observations of equilibrium flows to which I want to fit my model. An important question is whether or not, and at which rate, the empirical risk $R_N(\boldsymbol{\theta})$ approaches the *population risk*, defined by

$$R(\boldsymbol{\theta}) := \mathbb{E}_{\mathbf{d}}[\|\mathbf{x}^\star(\mathbf{d}) - \mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d})\|]$$

The population risk gives a measure of the prediction capability of the trained model, which is a better measure of the quality of the model. Let me define the loss function $\ell_{\boldsymbol{\theta}} : \mathbb{R}_+^K \to \mathbb{R}_+$:

$$\ell_{\boldsymbol{\theta}}(\mathbf{d}) := \|\mathbf{x}^\star(\mathbf{d}) - \mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d})\|$$

The parametric loss function belongs to the following function class, called the *loss class*:

$$\mathcal{L} := \{\mathbf{d} \in \mathcal{D} \mapsto \ell_{\boldsymbol{\theta}}(\mathbf{d}) \,|\, \boldsymbol{\theta} \in \Theta\}$$

Then, the empirical risk and the population risk are given by $R_N(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N \ell_{\boldsymbol{\theta}}(\mathbf{d}_i)$ and $R(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{d}}[\ell_{\boldsymbol{\theta}}(\mathbf{d})]$ respectively, and I am interested in studying the behavior of the following quantity

$$\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}} := \sup_{\boldsymbol{\theta}\in\Theta} |R(\boldsymbol{\theta}) - R_N(\boldsymbol{\theta})|$$

where $\mathbb{P}_N$ is the *empirical distribution* assigning mass $1/N$ to each of the samples $\mathbf{d}_1, \cdots, \mathbf{d}_N$, and $\mathbb{P}$ is the distribution of the random demand vector $\mathbf{d}$. The above quantity measures the absolute deviation between the sample average and the population average. I also note that $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}}$ is a random variable since it is a function of the N random samples $\{\mathbf{d}_1, \cdots, \mathbf{d}_N\}$.

In Chapter 5, I combine sensitivity results in optimization theory [48], [174] and results in approximation theory [54], [132], [37] in order to obtain a tail bound on the distribution of $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}}$. With the additional assumption that the edge cost functions I want to estimate belong to a family of L-Lipschitz and c-strong-monotone, this enables me to get the number N of data samples needed so that the probability of having $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}} \leq \epsilon$ is at least $1 - \delta$

$$\sqrt{N} \geq \frac{\sqrt{|\mathcal{E}|}\Big(60 + (L-c)\sqrt{2\log(\frac{1}{\delta})}\Big)}{\epsilon}\mathcal{J}(c, L)$$

where $\mathcal{J}(c, L)$ is a function that depends on the Lipschitz constant L and strong monotonicity constant c of my restricted family of candidate edge cost functions:

$$\mathcal{J}(c, L) := \frac{c\,(L-c)}{L^2\big(1 - \sqrt{1 - \frac{c^2}{L^2}}\big)}$$

Doing an asymptotic analysis on $\mathcal{J}(c, L)$ enables me to show, independently of the sample distribution, that the sample size required to have a small uniform deviation $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}}$ with high probability, grows linearly in L/c when L/c goes to infinity, and linearly in $1 - c/L$ when c/L approaches 1.

1.4 Two frameworks for estimating traffic flow on the highway and arterial networks

The framework of fitting a selfish routing model to the traffic flows $\mathbf{x} = (x_e)_{e\in\mathcal{E}}$, where $\mathcal{E}$ is the set of edges representing the road segments in the network, assumes that I have the sensing infrastructure that enables me to measure the traffic flows. Among the types of data that are the most commonly used by the transportation community are data obtained from loop detectors, which are induction loops placed beneath the road. Vehicles induce currents into the loop which are counted by an electric meter. Video cameras coupled with a video processor are also used to count vehicles passing a specific location. However, the cost of deploying and maintaining the sensing infrastructure is expensive, hence traffic sensors are sparse.

In Chapter 9, I model the state of the traffic flow on highways using discretized hyperbolic scalar partial differential equations. Specifically, I discretize the so-called *Lighthill-Whitham-Richards* (LWR) equation [109, 136] with a triangular flux function using a Godunov scheme [100, 106, 146]. The resulting partial differential equations have been widely used in the in the scientific community for modelling traffic, they also known as the *Cell Transmission Model*

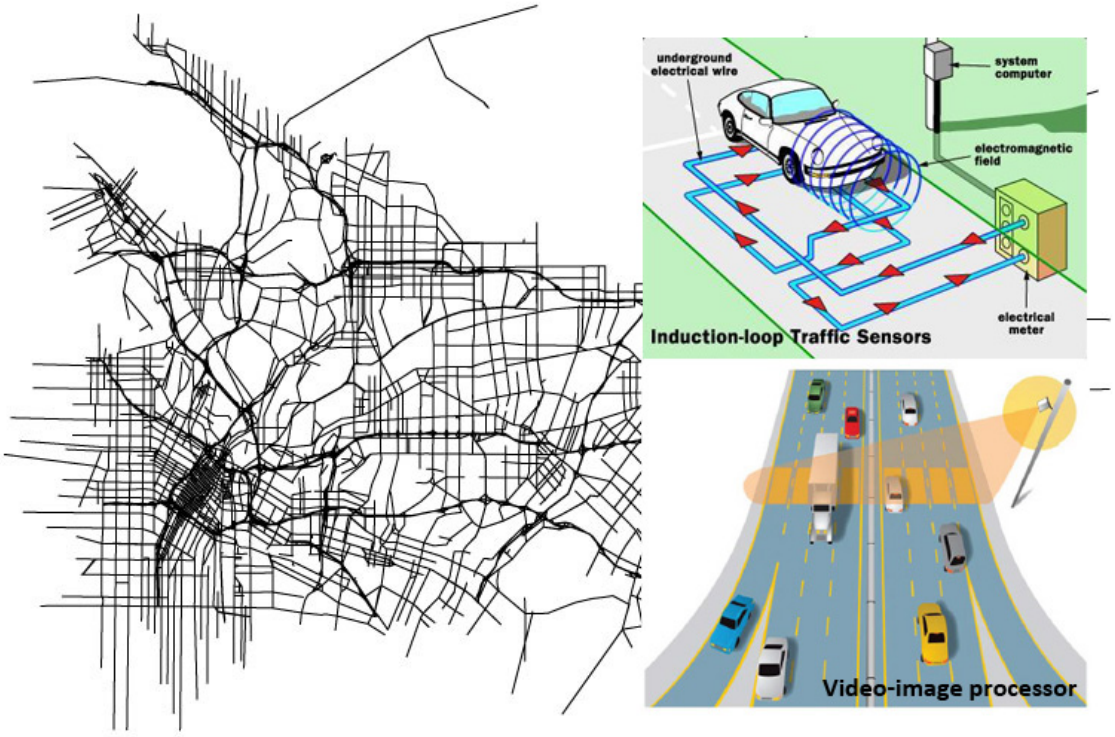

Figure 1.2: Induction-loop traffic sensors and video cameras in order to measure traffic flow on highways. In the background, the network of Los Angeles obtained from OpenStreetMap is presented.

(CTM) [50, 51] in the transportation literature. Formally, I assume the stretch of highway is discretized into n cells, and its state at time step t is represented by a vector $\mathbf{x}_t \in [0, x_{\max}]^n$, where $x_{\max} \in \mathbb{R}_{>0}$ is the maximum accepted flow at any cell. Then, I show that there exists a partition of the hypercube $[0, x_{\max}]^n$ into a family of polyhedra $\mathcal{F} := \{\mathcal{P}_1, \cdots, \mathcal{P}_K\}$ such that if the state $\mathbf{x}_t$ belongs to a specific polyhedron $\mathcal{P} \in \mathcal{F}$, then the transition equation for the discretized LWR partial differential equations is fully specified by this polyhedron

$$\mathbf{x}_{t+1} = \mathbf{A}_{\mathcal{P}}\mathbf{x}_t + \mathbf{b}_{\mathcal{P}} \quad \text{if} \quad \mathbf{x}_t \in \mathcal{P}$$

Hence, I have a hybrid system switching between K linear modes. However, K being exponential in the number n of cells in the discretization of the LWR partial differential equation, I reduce the number of modes by clustering them with the *k-means* algorithm. Then, I propose a feasible estimation approach based on the interactive multiple model (IMM) [9].

My proposed method is applied to measurements from 29 PeMS stations along an 18-mile long stretch of I-880 and is compared to the Ensemble Kalman filter (EnKF), which is an algorithm commonly used in the traffic monitoring community [170]. The results are shown

in Figure 1.3. The recovered state from both the EnKF and my proposed algorithm are very similar, while there is a significant performance gain, as described in Chapter 9.

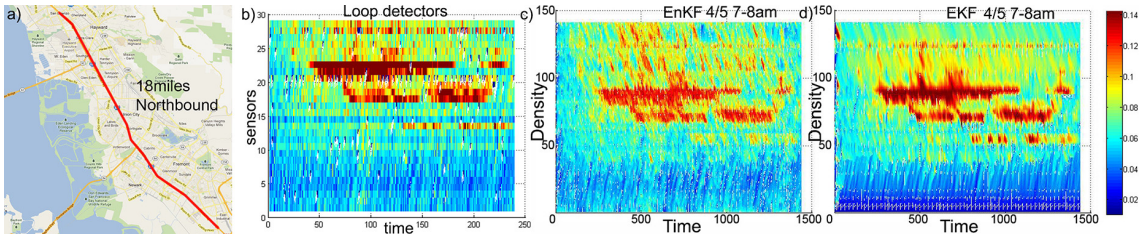

Figure 1.3: Based on measurements from 29 PeMS stations along an 18-mile long stretch of I-880 in the Bay Area, see the contour plot in b), I recover the traffic flow at a resolution of 198m using my hybrid Kalman filter algorithm, see plot in d). My results are comparable to the contour plot yielded by the Ensemble Kalman filter, considered a state-of-the-art algorithm for the estimation of non-linear multi-dimensional systems.

In Chapter 10, I partially address the shortage of traditional traffic monitoring sensors, such as loop detectors and video cameras, by leveragin the large penetration of mobile phones among the driving population and the ubiquitous coverage of service providers in urban areas. In the recent years, mobile phones have become an increasingly popular source of location data for the transportation community. In addition to dynamic probing by means of in-car GPS traces, location data are available directly from cellular communication network operators. A variety of phone to cell communication events such as *handovers* (HO), *location updates* (LU) and *call data records* (CDR) [160, 161] are being recorded by cellular network infrastructures, and this data has already been shown to be effective in studying urban environments [36, 90]. Since typical cellular networks in urban agglomerations include thousands of cells, HO/LU/CDR events can be used effectively to estimate traffic flow without requiring any additional infrastructure. Hence, I propose in Chapter 10 a framework for the fusion of cellular and loop data, which is illustrated in Figure 1.4.

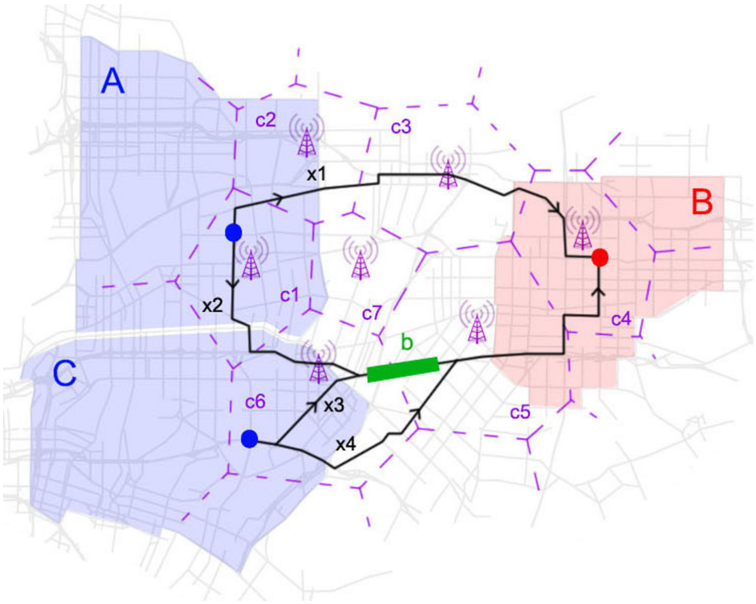

Figure 1.4: In this illustration of the cellular and loop data fusion, I have two origins A and C (the blue traffic regions and their centroid as blue dots) and one destination B (the red traffic region). I have routes going from A to B and routes going from C to B. The Voronoi partition of the cellular network based on the cell tower locations is depicted in purple dashed regions. I also measure the vehicle count on the green link from loop detectors. In Chapter 10, I propose a tractable framework merging these different sources of data in order to estimate the traffic flow in the urban network.

Part I

The impact of GPS-enabled shortest path routing on mobility: a game theoretic approach

Chapter 2

Convex optimization, variational inequality, and the selfish routing game

In the present chapter, I provide in a unified fashion the theoretical foundations and main techniques in game theory, convex optimization, and variational inequality theory, with an emphasis on the selfish routing game, and its extension to the heterogeneous setting. Specifically, I first characterize convex optimization programs and first-order optimality conditions for solutions to this class of mathematical programs. For further details, I refer the reader to, *e.g.*, [26]. Then, I present fundamental definitions in variational inequality theory, which are extensively covered in [61]. Lastly, I define the selfish routing game and its heterogeneous extension, in which drivers are assumed to experience different travel costs. The selfish routing game, also known as the traffic assignment problem, is studied in details in, *e.g.*, [131]. Heterogeneous games have been studied previously in, *e.g.*, [59, 91, 65, 114]. While the definitions and results presented in the present chapter are not novel, they are of didactic importance since they will be used throughout this work.

2.1 Convex optimization

We define central objects in convex optimization. We refer the reader to, *e.g.* [26], for a more complete treatment of the subject.

Convex sets

A set $\mathcal{X}$ in a Hilbert space $\mathcal{H}$ is convex if

$$\alpha\mathbf{x} + (1-\alpha)\mathbf{y} \in \mathcal{X}, \quad \forall\, \mathbf{x},\, \mathbf{y} \in \mathcal{X},\, \forall\, \alpha \in [0,1]$$

Convex functions

Let $\mathcal{X}$ be a convex set in a Hilbert space $\mathcal{H}$, and let $f : \mathcal{X} \to \mathbb{R}$ be a mapping. We now give definitions of convexity properties: f is *convex* on $\mathcal{X}$ if

$$f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1-\alpha)f(\mathbf{y}), \quad \forall\, \mathbf{x},\, \mathbf{y} \in \mathcal{X},\, \forall\, \alpha \in [0,1]$$

f is *strictly convex* on $\mathcal{X}$ if

$$f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) < \alpha f(\mathbf{x}) + (1-\alpha)f(\mathbf{y}), \quad \forall\, \mathbf{x} \neq \mathbf{y} \in \mathcal{X},\, \forall\, \alpha \in [0,1]$$

f is *strongly convex* on $\mathcal{X}$, if there exists a constant $c > 0$ such that

$$f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1-\alpha)f(\mathbf{y}) - \frac{c}{2}\alpha(1-\alpha)\|\mathbf{x}-\mathbf{y}\|^2$$

We note that strong convexity of f on $\mathcal{X}$ implies strict convexity of f on $\mathcal{X}$, which in turn implies convexity of f on $\mathcal{X}$, but both of these implications cannot be reversed in general. In addition, strong convexity of f is also equivalent to convexity of $f(\mathbf{x}) - \frac{c}{2}\|\mathbf{x}\|_2^2$, and implies that, see Section 9.1.2. in [26]

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \frac{c}{2}\|\mathbf{y}-\mathbf{x}\|_2^2, \quad \forall\, \mathbf{x},\, \mathbf{y} \in \mathcal{X} \tag{2.1}$$

Convex optimization programs

Given a convex set $\mathcal{X}$ in a Hilbert space $\mathcal{H}$ and a convex function $f : \mathcal{X} \to \mathbb{R}$, a convex optimization problem consists in finding $\mathbf{x}^\star \in \mathcal{X}$ that solves

$$\min\, f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{X} \tag{2.2}$$

i.e. $f(\mathbf{x}^\star) \leq f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$. Convex optimization problems are an important subclass of optimization problems that can be solved very efficiently using well-studied first-order optimization algorithms such as the gradient descent and conditional gradient algorithms .

Thus, modeling real-world processes as variational inequality problems or convex optimization problems is a common practice as it enables to leverage powerful mathematical tools for the study of such processes. For example, in economics, knowing the consumer utility function enables to adjust prices to achieve some demand level [94]. In many cases in control, a low complexity controller requires less computation for little performance loss [94, 163]. In transportation science, the selfish behavior of agents (from shorted path routing) leads to an aggregate cost in the network worse than the system's optimum, and which can be analytically quantified [137, 46]. Taxation schemes can be designed to incentivize system optimal drivers' decisions [65, 91].

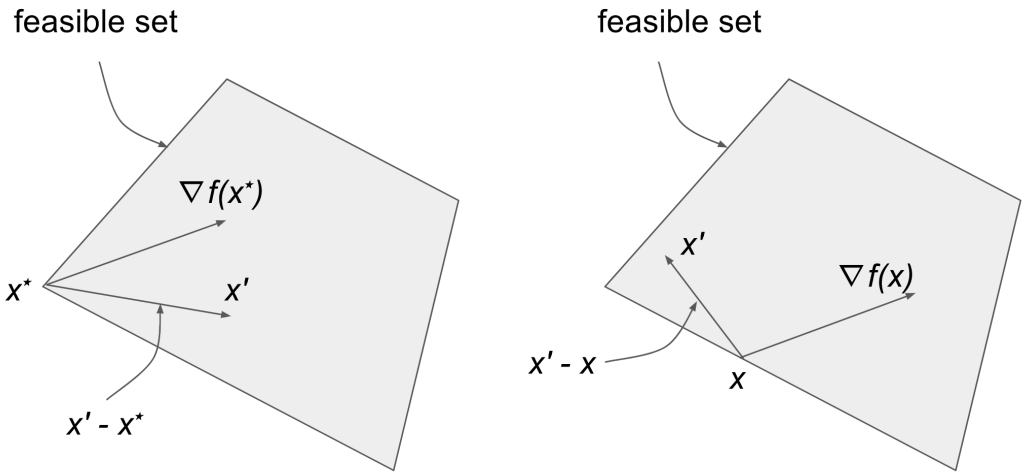

Figure 2.1: Geometrical interpretation of the first-order optimality condition. In the left figure, $\mathbf{x}^\star \in \mathcal{X}$ satifies the minimum principle (2.3) because $\nabla f(\mathbf{x}^\star)$ forms an acute angle with all the feasible directions $\mathbf{x}' - \mathbf{x}^\star$. The point $\mathbf{x}^\star$ is thus an optimal solution to the convex program (2.2). In the right figure, the feasible point $\mathbf{x}$ is not a solution to the convex program (2.2).

First-order optimality condition

Given a convex set $\mathcal{X}$ in a Hilbert space $\mathcal{H}$, if the function $f : \mathcal{X} \to \mathbb{R}$ is continuously differentiable on $\mathcal{X}$, then the first optimality condition for the convex program (2.2) is

$$\langle \nabla f(\mathbf{x}^\star), \mathbf{x} - \mathbf{x}^\star \rangle \geq 0, \, \forall \, \mathbf{x} \in \mathcal{X} \tag{2.3}$$

We refer to §4.2.3. in [27] for a proof. The condition means that the feasible region only lies in the half-space where the potential increases. Otherwise, we would have found a lower potential value by using the first-order Taylor expansion of $t \in [0, 1] \mapsto f(\mathbf{x}^\star + t(\mathbf{x} - \mathbf{x}^\star))$.

Proposition 2.1. *Let $\mathcal{D}$ be a compact convex subset of $\mathbb{R}^n$, let f be strongly convex with parameter c, and let $\boldsymbol{x}^\star$ be the unique solution to the VIP* (2.9)*. Then, for every $\boldsymbol{x} \in \mathcal{D}$,*

$$\|\boldsymbol{x} - \boldsymbol{x}^\star\|_2^2 \leq 2(f(\boldsymbol{x}) - f(\boldsymbol{x}^\star))/c$$

Proof. By definition of strong convexity (2.1),

$$\begin{aligned} f(\mathbf{x}) &\geq f(\mathbf{x}^\star) + \langle \nabla f(\mathbf{x}^\star), \mathbf{x} - \mathbf{x}^\star \rangle + \frac{c}{2} \|\mathbf{x} - \mathbf{x}^\star\|_2^2 \\ &\geq f(\mathbf{x}^\star) + \frac{c}{2} \|\mathbf{x} - \mathbf{x}^\star\|_2^2 \end{aligned}$$

where the second inequality is given by the fact that $\langle \nabla f(\mathbf{x}^\star), \mathbf{x} - \mathbf{x}^\star \rangle \geq 0$, from the first-order optimality condition (2.3). □

2.2 Variational inequality

Variational inequality problems constitute a broad class of problems that encompasses convex optimization problems, and is used in game theory. We refer the reader to, *e.g.*, [61], [142] for additional references on the variational inequality problem.

Monotonicity

Let $\mathcal{X}$ be a convex set in a Hilbert space $\mathcal{H}$, and let $F : \mathcal{X} \to \mathcal{H}$ be a mapping. We now give definitions of monotone properties in order of increasing strength: F is *monotone* if

$$\forall\, \mathbf{x},\, \mathbf{x}' \in \mathcal{X} \quad \langle F(\mathbf{x}) - F(\mathbf{x}'),\, \mathbf{x} - \mathbf{x}' \rangle \geq 0 \tag{2.4}$$

F is *strictly monotone* if

$$\forall\, \mathbf{x} \neq \mathbf{x}' \in \mathcal{X} \quad \langle F(\mathbf{x}) - F(\mathbf{x}'),\, \mathbf{x} - \mathbf{x}' \rangle > 0$$

F is *strongly monotone* if, for some parameter $c \in \mathbb{R}_{>0}$,

$$\forall\, \mathbf{x},\, \mathbf{x}' \in \mathcal{X} \quad \langle F(\mathbf{x}) - F(\mathbf{x}'),\, \mathbf{x} - \mathbf{x}' \rangle \geq c\|\mathbf{x} - \mathbf{x}'\|^2$$

Equivalence to convexity

Let $f : \mathcal{X} \to \mathbb{R}$ be a continuously differentiable potential. Then f is convex if and only if its gradient ∇f is monotone. To prove this, we observe that convexity of f is equivalent to convexity of its restriction to every line segment in $\mathcal{X}$, *i.e.* for all $\mathbf{x}$, $\mathbf{x}' \in \mathcal{X}$, the function $f_{\mathbf{x},\mathbf{x}'} : t \mapsto f\left((1-t)\mathbf{x} + t\mathbf{x}'\right)$ defined on $[0, 1]$, is convex. Since $f_{\mathbf{x},\mathbf{x}'}$ is differentiable, this is equivalent to having a non-decreasing derivative $f'_{\mathbf{x},\mathbf{x}'}(t) = \langle \nabla f\left(\mathbf{x} + t(\mathbf{x}' - \mathbf{x})\right),\, \mathbf{x}' - \mathbf{x} \rangle$, for all $\mathbf{x}$, $\mathbf{x}' \in \mathcal{X}$, which is finally equivalent to condition (2.4) of monotonicity. By extension, we can obtain equivalence between strict convexity of f and strict monotonicity of ∇f with similar arguments taken in the strict sense. Finally, the potential f is said to be strongly convex if its gradient is strongly monotone, hence the equivalence follows from definition of strong convexity. Note that, by Lemma 1.2.3 in [122], strong convexity of f is equivalent to, for all $\mathbf{x}$, $\mathbf{x}' \in \mathcal{X}$

$$f(\mathbf{x}') \geq f(\mathbf{x}) + \langle \mathbf{x}' - \mathbf{x},\, \nabla f(\mathbf{x}) \rangle + \frac{c}{2}\|\mathbf{x}' - \mathbf{x}\|^2$$

Variational inequality problem

Given a closed and convex set $\mathcal{X}$ in a Hilbert space $\mathcal{H}$, the variational inequality problem is:

$$\text{find } \mathbf{x} \in \mathcal{X} \text{ such that } \langle F(\mathbf{x}^\star),\, \mathbf{x} - \mathbf{x}^\star \rangle \geq 0,\, \forall\, \mathbf{x} \in \mathcal{X} \tag{2.5}$$

The variational inequality problem can be seen as a generalization of the first-order optimality condition for convex programs (2.3)

2.3 Uniqueness results

Uniqueness result for the variational inequality problem

Let $\mathcal{X}$ be a convex set in a Hilbert space $\mathcal{H}$, and let $F : \mathcal{X} \to \mathcal{H}$ be a mapping. If F is involved in a variational inequality problem

$$\text{find } \mathbf{x}^\star \in \mathcal{X} \quad \text{s.t.} \quad \langle F(\mathbf{x}^\star), \mathbf{x} - \mathbf{x}^\star \rangle \geq 0 \quad \forall\, \mathbf{x} \in \mathcal{X} \tag{2.6}$$

then strict monotonicity of F implies that there exists at most one solution to the variational inequality problem. To see this, assume that $\tilde{\mathbf{x}}$ is another solution, then $\langle F(\mathbf{x}^\star), \tilde{\mathbf{x}} - \mathbf{x}^\star \rangle \geq 0$ and $\langle F(\tilde{\mathbf{x}}), \mathbf{x}^\star - \tilde{\mathbf{x}} \rangle \geq 0$. Adding the two inequalities together, we obtain $\langle F(\mathbf{x}^\star) - F(\tilde{\mathbf{x}}), \mathbf{x}^\star - \tilde{\mathbf{x}} \rangle \leq 0$. By strict monotonicity, this is only possible if $\mathbf{x}^\star = \tilde{\mathbf{x}}$.

Uniqueness result for the convex optimization problem

If F is replaced by the gradient ∇f of a continuously differentiable potential $f : \mathcal{X} \to \mathbb{R}$, then (2.6) is a *necessary condition* for optimality for the optimization program

$$\min\ f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{X} \tag{2.7}$$

To see this, we instantiate condition (2.6) to

$$\langle \nabla f(\mathbf{x}^\star), \mathbf{x} - \mathbf{x}^\star \rangle \geq 0, \quad \forall\, \mathbf{x} \in \mathcal{X} \tag{2.8}$$

and note that this implies that the feasible region only lies in the half-space where the potential f increases. Otherwise, we would have found a lower potential value by using the first-order Taylor expansion of $t \in [0, 1] \mapsto f(x^\star + t(\mathbf{x} - \mathbf{x}^\star))$. It turns out that when f is convex, condition (2.8) is also sufficient for optimality in (2.7), see *e.g.* §4.2.3. in [27]. In addition, if f is strictly convex, then ∇f is strictly monotone from the analysis in Section 2.2, and the uniqueness of a solution to the variational inequality problem implies uniqueness of a solution to the convex constrained optimization program.

2.4 Existence results

The following analysis is adapted from Chapter 1 of [95]. Let $\mathcal{D}$ be a compact convex subset of $\mathbb{R}^n$, and let $F : \mathcal{D} \to \mathbb{R}^n$ be a continuous mapping. The existence of a solution to the variational inequality problem

$$\text{find } \mathbf{x}^\star \in \mathcal{D} \quad \text{s.t.} \quad \langle F(\mathbf{x}^\star), \mathbf{x} - \mathbf{x}^\star \rangle \geq 0 \quad \forall\, \mathbf{x} \in \mathcal{D} \tag{2.9}$$

can be proven using *Brouwer's fixed-point theorem.*

Projection operators

First, the projection $P_{\mathcal{D}}(\mathbf{x}) = \arg\min_{\mathbf{z}\in\mathcal{D}} \|\mathbf{x}-\mathbf{z}\|_2$ is well defined for all $\mathbf{x} \in \mathbb{R}^n$. Indeed, the program $\min_{\mathbf{z}\in\mathcal{D}} \|\mathbf{x}-\mathbf{z}\|_2$ admits a solution from continuity of the Euclidean norm $\|\cdot\|_2$ on the compact set $\mathcal{D}$, and a unique one from the strict convexity of $\|\cdot\|_2$, see Appendix 2.3. In addition, from differentiability and convexity of $\mathbf{z} \mapsto \|\mathbf{x}-\mathbf{z}\|_2^2$, a vector $\mathbf{y}$ is optimal for $\min_{\mathbf{z}\in\mathcal{D}} \|\mathbf{x}-\mathbf{z}\|_2^2$. *i.e.* $\mathbf{y} = P_{\mathcal{D}}(\mathbf{x})$ if and only if it satisfies the first-order optimality condition (2.8). Instantiating (2.8) to the present case, we obtain

$$\langle \mathbf{y}-\mathbf{x},\, \mathbf{z}-\mathbf{y}\rangle \geq 0 \quad \forall\, \mathbf{z}\in\mathcal{D} \tag{2.10}$$

Rewriting condition (2.10) as $\langle \mathbf{x}-\mathbf{y},\, \mathbf{z}-\mathbf{y}\rangle \leq 0$ for every $\mathbf{z}\in\mathcal{D}$, it means that the convex domain $\mathcal{D}$ lies in the half-space that is away from the direction $\mathbf{x}-\mathbf{y} = \mathbf{x} - P_{\mathcal{D}}(\mathbf{x})$.

Contraction property of projection operators

The characterization (2.10) of $\mathbf{y} = P_{\mathcal{D}}(\mathbf{x})$ implies that the projection operator $P_{\mathcal{D}}(\cdot)$ is 1-Lipschitz, thus continuous. To prove this, let $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$, and denote their projection by $\mathbf{y} = P_{\mathcal{D}}(\mathbf{x})$ and $\mathbf{y}' = P_{\mathcal{D}}(\mathbf{x}')$. Applying (2.10) to $\mathbf{y}$ and $\mathbf{y}'$ with $\mathbf{z} = \mathbf{y}'$ and $\mathbf{z} = \mathbf{y}$ respectively, we obtain

$$\begin{aligned}\langle \mathbf{y}-\mathbf{x},\, \mathbf{y}'-\mathbf{y}\rangle \geq 0 \;&\Rightarrow\; \langle \mathbf{y},\, \mathbf{y}-\mathbf{y}'\rangle \leq \langle \mathbf{x},\, \mathbf{y}-\mathbf{y}'\rangle \\ \langle \mathbf{y}'-\mathbf{x}',\, \mathbf{y}-\mathbf{y}'\rangle \geq 0 \;&\Rightarrow\; \langle \mathbf{y}',\, \mathbf{y}'-\mathbf{y}\rangle \leq \langle \mathbf{x}',\, \mathbf{y}'-\mathbf{y}\rangle\end{aligned}$$

Adding the two inequalities, we obtain

$$\|\mathbf{y}-\mathbf{y}'\|_2^2 \leq \langle \mathbf{x}-\mathbf{x}',\, \mathbf{y}-\mathbf{y}'\rangle \leq \|\mathbf{x}-\mathbf{x}'\|\,\|\mathbf{y}-\mathbf{y}'\|$$

Hence $\|P_{\mathcal{D}}(\mathbf{x}) - P_{\mathcal{D}}(\mathbf{x}')\| = \|\mathbf{y}-\mathbf{y}'\| \leq \|\mathbf{x}-\mathbf{x}'\|$

Equivalence of the VIP to a fixed point problem

By rewriting the equilibrium condition in (2.9)

$$\langle \mathbf{x}^\star - (\mathbf{x}^\star - F(\mathbf{x}^\star)),\, \mathbf{x}-\mathbf{x}^\star\rangle \geq 0 \quad \forall\, \mathbf{x}\in\mathcal{D}$$

we use characterization (2.10) to obtain that $\mathbf{x}^\star \in \mathcal{D}$ is a solution to the VIP if and only $\mathbf{x}^\star = P_{\mathcal{D}}(\mathbf{x}^\star - F(\mathbf{x}^\star))$. Finally, we note that the mapping $\mathbf{x} \mapsto P_{\mathcal{D}}(\mathbf{x} - F(\mathbf{x}))$ is continuous from the convex compact subset $\mathcal{D}$ to itself because it is the composition of two continuous functions. By Brouwer's fixed-point theorem, it admits a fixed point. This implies that the VIP (2.9) admits at least one solution.

2.5 The selfish routing game

Routing games were formulated by [166], and are extensively studied in transportation science, see *e.g.* [131].

Setting

We consider a non-cooperative game on a network represented by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ equipped with continuous, non-decreasing congestion functions $c_e(\cdot) : \mathbb{R}_+ \to \mathbb{R}_{>0}$ for each $e \in \mathcal{E}$. The set of players is partitioned in *populations* $\{\mathcal{X}_k\}_{k\in[K]}$. For each $k \in [K]$, players in $\mathcal{X}_k$ have available a set of simple paths $\mathcal{P}_k$ from a common source $s_k \in \mathcal{V}$ to a common sink $t_k \in \mathcal{V}$. For each population $\mathcal{X}_k$, we define $d_k \in \mathbb{R}_+$ its total flow, and $\boldsymbol{\mu}^k = (\mu_p^k)_{p\in\mathcal{P}_k} \in \mathbb{R}_+^{\mathcal{P}_k}$ its *path assignment*, which satisfies, for all $k \in [K]$

$$\boldsymbol{\mu}^k \in \Delta^k := \left\{ \mathbf{u} \in \mathbb{R}_+^{\mathcal{P}_k} \,:\, \textstyle\sum_{p\in\mathcal{P}_k} u_p^k = d_k \right\} \tag{2.11}$$

We denote $\mathcal{P}$ the disjoint union $\mathcal{P} = \sqcup_{k=1}^K \mathcal{P}_k$, thus $\mathbb{R}_+^{\mathcal{P}} = \Pi_{k=1}^K \mathbb{R}_+^{\mathcal{P}_k}$. Under population demand $\mathbf{d} = (d_k)_{k\in[K]}$, the path assignment can be summarized by $\boldsymbol{\mu} = (\boldsymbol{\mu}^1, \cdots, \boldsymbol{\mu}^K) \in \mathbb{R}_+^{\mathcal{P}}$ in the feasible set Δ defined as the product space of feasible population paths $\Delta := \Delta^1 \times \cdots \times \Delta^K$. In other words,

$$\Delta := \left\{ \boldsymbol{\mu} \in \mathbb{R}_+^{\mathcal{P}} : \textstyle\sum_{p\in\mathcal{P}_k} \mu_p^k = d_k,\, \forall\, k \in [K] \right\} \tag{2.12}$$

The path assignment determines the vector of *edge flows* $\mathbf{x} = (x_e)_{e\in\mathcal{E}} \in \mathbb{R}_+^{\mathcal{E}}$ such that each entry x_e is defined as $x_e = \sum_{k=1}^K \sum_{p\in\mathcal{P}_k : e\in p} \mu_p^k$, *i.e.* it is the sum of the flows of all paths going through edge e. In matrix form, it can be written compactly as $x_e = (\mathbf{M}\boldsymbol{\mu})_e$, where $\mathbf{M} \in \mathbb{R}^{\mathcal{E}\times\mathcal{P}}$ is an incidence matrix with entries defined as $M_{e,p} = \mathbf{1}_{e\in p}$.

For each edge e, we suppose that the edge flow incurs a cost $c_e(x_e)$ which only depends on the flow x_e on edge e. This assumption is common in transportation science and is sometimes referred as the the separability assumption, see [13] and [46]. We define the associated mapping

$$F \,:\, \mathbf{x} \in \mathbb{R}_+^{\mathcal{E}} \mapsto F(\mathbf{x}) = (c_e(x_e))_{e\in\mathcal{E}} \tag{2.13}$$

The cost of choosing a path p is the sum of edge costs along the path, *i.e.* $\sum_{e\in p} c_e(x_e)$. This can be written in matrix form as:

$$\textstyle\sum_{e\in p} c_e(x_e) = \sum_{e\in p} c_e((\mathbf{M}\boldsymbol{\mu})_e) = \sum_{e\in p} (F(\mathbf{M}\boldsymbol{\mu}))_e = C_p^T F(\mathbf{M}\boldsymbol{\mu}) \tag{2.14}$$

where $C_p \in \mathbb{R}^{\mathcal{E}}$, $p \in \mathcal{P}$ are the columns of the incidence matrix $\mathbf{M}$. Since $\sum_{e\in p} c_e(x_e) = C_p^T F(\mathbf{M}\boldsymbol{\mu})$ is fully determined by $\boldsymbol{\mu}$, we define $\ell_p(\boldsymbol{\mu}) := \sum_{e\in p} c_e(x_e)$ and we write $\ell(\boldsymbol{\mu})$ to denote the vector of path cost functions $(\ell_p(\boldsymbol{\mu}))_{p\in\mathcal{P}}$. Hence, the path costs can be written compactly as the vector of functions

$$\ell \,:\, \boldsymbol{\mu} \in \Delta \mapsto \mathbf{M}^T F(\mathbf{M}\boldsymbol{\mu}) \tag{2.15}$$

Equilibrium in routing games

We say that $\boldsymbol{\mu}^\star \in \Delta(\mathbf{d})$ is a Nash equilibrium for the routing game, or satisfies the *Wardrop conditions*, if, for all $k \in [K]$ and $p \in \mathcal{P}_k$

$$\mu_p^k > 0 \implies \ell_p^k(\boldsymbol{\mu}^\star) = \min_{q\in\mathcal{P}_k} \ell_q^k(\boldsymbol{\mu}^\star) \tag{2.16}$$

In other words, for every population k, some path $p \in \mathcal{P}_k$ is only used if it is of least cost in $\mathcal{P}_k$. This is equivalent to (5.3) below, see §3.2 in [131]

$$\langle \ell(\boldsymbol{\mu}^\star),\, \boldsymbol{\mu} - \boldsymbol{\mu}^\star \rangle \geq 0, \quad \forall\, \boldsymbol{\mu} \in \Delta \tag{2.17}$$

Since $\ell(\boldsymbol{\mu}) = \mathbf{M}^T F(\mathbf{M}\boldsymbol{\mu})$, substituting in (5.3) gives the condition $\langle F(\mathbf{M}\boldsymbol{\mu}^\star),\, \mathbf{M}(\boldsymbol{\mu} - \boldsymbol{\mu}^\star) \rangle \geq 0$, re-written as

$$\langle F(\mathbf{x}^\star),\, \mathbf{x} - \mathbf{x}^\star \rangle \geq 0, \quad \forall\, \mathbf{x} \in \mathcal{D} \tag{2.18}$$

where $\mathbf{x}^\star = \mathbf{M}\boldsymbol{\mu}^\star$ is the Nash equilibrium for the edge flows, and $\mathcal{D} = \mathbf{M}\Delta = \{\mathbf{M}\boldsymbol{\mu} : \boldsymbol{\mu} \in \Delta\} \subset \mathbb{R}_+^{\mathcal{E}}$ is the set of feasible edge flows, see *e.g.* §3.2.1. [131]. Note that $\mathcal{D}$ is compact convex since it is the image of $\mathbf{M}$ restricted to the compact convex set Δ. However, constructing the feasible set Δ requires enumerating all (potentially used) simple paths, the set $\mathcal{P}_k$, for each population $k \in [K]$, which may be intractable on large graphs. An alternative is to use a vertex-representation of the flow constraints, see *e.g.* §2.2.2 in [131].

Formulation as a potential game

If the congestion function $c_e(\cdot) : \mathbb{R}_+ \to \mathbb{R}_{>0}$ is non-decreasing and continuous for each $e \in \mathcal{E}$, then Beckmann et al. proved in [13] that the equilibrium edge flow always exists as a solution of the following convex program:

$$\min_{\mathbf{x}} \sum_{e \in \mathcal{E}} \int_0^{x_e} c_e(u)\, du \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{D} \tag{2.19}$$

Since the domain $\mathcal{D}$ is a closed and convex subset of $\mathbb{R}_+^{\mathcal{E}}$, and the potential function $\mathbf{x} \in \mathbb{R}_+^{\mathcal{E}} \mapsto \sum_{e \in \mathcal{E}} \int_0^{x_e} c_e(u)\, du$ is convex from our assumption that the congestion functions $c_e(\cdot)$ are non-decreasing and continuous for each $e \in \mathcal{E}$, the above problem is convex. As the domain $\mathcal{D}$ is a compact set, the objective function in the above program attains its optimum on $\mathcal{D}$. This proves the existence of an optimal solution to the above program.

To prove that an optimal solution of the program (2.19) is an equilibrium, we first formulate the (2.19) in terms of path flow vector $\boldsymbol{\mu} \in \mathbb{R}_+^{\mathcal{P}}$

$$\min_{\boldsymbol{\mu}} \sum_{e \in \mathcal{E}} \int_0^{(\mathbf{M}\boldsymbol{\mu})_e} c_e(u)\, du \quad \text{s.t.} \quad \mathbf{A}\boldsymbol{\mu} = \mathbf{d},\, \boldsymbol{\mu} \geq 0 \tag{2.20}$$

where $\mathbf{A} \in \mathbb{R}^{K \times \mathcal{P}}$ is an incidence matrix with entries defined as $A_{k,p} = \mathbf{1}_{p \in \mathcal{P}_k}$. We note that we have just rewritten in matrix form the domain of feasible path flows Δ defined in (2.12). The Karush–Kuhn–Tucker (KKT) conditions associated to the program (2.20) are

$$\ell(\boldsymbol{\mu}) - \mathbf{A}^T\boldsymbol{\lambda} = \boldsymbol{\pi} \tag{2.21}$$

$$\mathbf{A}\boldsymbol{\mu} = \mathbf{d},\, \boldsymbol{\mu} \geq 0 \tag{2.22}$$

$$\boldsymbol{\pi} \geq 0 \tag{2.23}$$

$$\boldsymbol{\pi}^T\boldsymbol{\mu} = 0 \tag{2.24}$$

A vector of path flows $\boldsymbol{\mu}^\star$ optimal for the program (2.20) must satisfy the above KKT conditions, which can be shown to be equivalent to the Wardrop conditions (2.16). The associated vector of edge flows $\mathbf{x}^\star = \mathbf{M}\boldsymbol{\mu}^\star$.

Formulation (2.19) of the routing game is a particular case of a *potential game* with contimuous player sets, because it admits a real-valued potential function encoding players' strategies, and local minimizers of the potential are Nash equilibria. For more details on potential games, we refer to [140] and [121]. More generally, if we denote by $F : \mathbb{R}_+^{\mathcal{E}} \to \mathbb{R}_+^{\mathcal{E}}$ the operator encoding the congestion $F(\mathbf{x}) = (c_e(\mathbf{x}))_{e\in\mathcal{E}}$ on each edge e, then the game defined in (2.18) is *potential* if and only if it satisfies the following *symmetry condition*

$$\frac{\partial c_e(\mathbf{x})}{\partial x_{e'}} = \frac{\partial c_{e'}(\mathbf{x})}{\partial x_e} \quad \forall\, e, e' \in \mathcal{E},\ \forall\, \mathbf{x} \in \mathbb{R}_+^{\mathcal{E}} \tag{2.25}$$

We refer to [46] for more details. The *separability assumption* (2.13) implies that

$$\frac{\partial c_e(\mathbf{x})}{\partial x_{e'}} = \frac{\partial c_e(x_e)}{\partial x_{e'}} = \begin{cases} c_e'(x_e) & \text{if } e = e' \\ 0 & \text{otherwise} \end{cases} \tag{2.26}$$

Hence, the *symmetry condition* (2.25) is satisfied, giving rise to the potential formulation (2.19).

Gradient of the potential

Let us define the potential function $\phi : \mathbb{R}_+^{\mathcal{E}} \to \mathbb{R}$ such that, for all $\mathbf{x} \in \mathbb{R}_+^{\mathcal{E}}$,

$$\phi(\mathbf{x}) := \sum_{e\in\mathcal{E}} \int_0^{x_e} c_e(u)du \tag{2.27}$$

By definition, it is differentiable, with gradient given by

$$\nabla\phi(\mathbf{x}) = (c_e(x_e))_{e\in\mathcal{E}}$$

Let us define the potential function $f : \mathbb{R}_+^{\mathcal{P}} \to \mathbb{R}$ such that, for all $\boldsymbol{\mu} \in \mathbb{R}_+^{\mathcal{P}}$,

$$f(\boldsymbol{\mu}) := \sum_{e\in\mathcal{E}} \int_0^{(\mathbf{M}\boldsymbol{\mu})_e} c_e(u)du \tag{2.28}$$

We note that $f(\boldsymbol{\mu}) = \phi(\mathbf{M}\boldsymbol{\mu})$, where we recall that $\mathbf{M} = (\mathbf{1}_{e\in p})_{e,p}$ is the edge to path incidence matrix, and thus the gradient is

$$\nabla_{\boldsymbol{\mu}} f(\boldsymbol{\mu}) = \nabla_{\boldsymbol{\mu}}(\phi(\mathbf{M}\boldsymbol{\mu})) = \mathbf{M}^T\nabla_{\mathbf{x}}\phi(\mathbf{M}\boldsymbol{\mu}) = (\textstyle\sum_{e\in p} c_e((\mathbf{M}\boldsymbol{\mu})_e))_{p\in\mathcal{P}} = (\ell_p(\boldsymbol{\mu}))_{p\in\mathcal{P}} \tag{2.29}$$

Hence, the gradient of the potential with respect to the path flows is the vector of path costs.

2.6 The heterogeneous routing game

We have assumed in the previous sections that drivers experience the same cost when travelling along the edges of the network. However, drivers usually experience different travel costs because, *e.g.*, they operate different types of vehicles (in [62], the authors consider heavy-duty vehicles from cars), or have different routing preferences (in [149], the authors consider drivers who take shortcuts by travelling residential roads). Heterogeneous games have been studied previously [59, 91, 65, 114].

One possible formulation consists in indexing the different types of users with $t \in [T]$. Each population $k \in [K]$ is thus partitioned into types t, and population k of type t has a mass $d_{t,k} \in \mathbb{R}_+$ which is divided among paths $p \in \mathcal{P}_k$. The flow allocation is encoded by the vector $\boldsymbol{\mu}_t^k \in \mathbb{R}_+^{\mathcal{P}_k}$ which belongs to the feasible set $\Delta_t^k := \{\mathbf{u} \in \mathbb{R}_+^{\mathcal{P}_k} : \sum_{p\in\mathcal{P}_k} u_p^k = d_{t,k}\}$. We can aggregate these quantities per type and define $\boldsymbol{\mu}_t = (\boldsymbol{\mu}_t^k)_{k\in[K]} \in \mathbb{R}^{\mathcal{P}}$ which belongs to the feasible set Δ_t defined as:

$$\Delta_t := \Delta_t^1 \times \cdots \times \Delta_t^K = \{\boldsymbol{\mu}_t \in \mathbb{R}_+^{\mathcal{P}} \,:\, \textstyle\sum_{p\in\mathcal{P}_k} \mu_{t,p}^k = d_{t,k},\, \forall\, k \in [K]\} \tag{2.30}$$

The edge flow for type t is $\mathbf{x}_t = \mathbf{M}\boldsymbol{\mu}_t \in \mathbb{R}_+^{\mathcal{E}}$, where $\mathbf{M} \in \mathbb{R}^{\mathcal{E}\times\mathcal{P}}$ is the edge-path incidence matrix defined above. For each type $t \in [T]$, we assume that the cost of travelling edge e is a function $c_{t,e}(\sum_{t'\in[T]} x_{t',e})$ of the sum of the flows on edge e incurred by each type of agent. Note that the cost of travelling e is specific to the type t, and only depends on the flow on edge e (which is an extension of the separability assumption to the heterogeneous case). We define the operator $F_t : \mathbb{R}_+^{\mathcal{E}} \to \mathbb{R}_+^{\mathcal{E}}$ such that $F_t(\mathbf{x}) = (c_{t,e}(x_e))_{e\in\mathcal{E}}$, hence the edge costs for a driver of type t are encoded by $F_t(\sum_{t'\in[T]} \mathbf{x}_{t'})$. The cost of travelling path p is thus $\sum_{e\in p} c_{t,e}(\sum_{t'\in[T]} x_{t',e})$. We can define the vector of path costs $\ell_t((\boldsymbol{\mu}_{t'})_{t'\in[T]}) := (\ell_{t,p}((\boldsymbol{\mu}_{t'})_{t'\in[T]}))_{p\in\mathcal{P}}$ associated to a specific type of vehicles t in terms of path flows with the following operator, for all $t \in [T]$

$$\begin{array}{rccc} \ell_t : & \Pi_{t'\in[T]}\Delta_{t'} & \to & \mathbb{R}_+^{\mathcal{P}} \\ & (\boldsymbol{\mu}_{t'})_{t'\in[T]} & \mapsto & \mathbf{M}^T F_t(\mathbf{M}\sum_{t'\in[T]}\boldsymbol{\mu}_{t'}) \end{array} \tag{2.31}$$

The Wardrop conditions (2.16) can be extended to the heterogeneous case. We say that $(\boldsymbol{\mu}_t^\star)_{t\in[T]} \in \Pi_{t\in[T]}\Delta_t$ is a Nash equilibrium if

$$\mu_{t,p}^{\star,k} > 0 \implies \ell_{t,p}^k((\boldsymbol{\mu}_{t'}^\star)_{t'\in[T]}) = \min_{q\in\mathcal{P}_k} \ell_{t,q}^k((\boldsymbol{\mu}_{t'}^\star)_{t'\in[T]}), \quad \forall\, k \in [K],\, \forall\, p \in \mathcal{P}_k,\, \forall\, t \in [T]$$

This is equivalent to finding $(\boldsymbol{\mu}_t^\star)_{t\in[T]} \in \Pi_{t'\in[T]}\Delta_{t'}$ such that

$$\sum_{t\in[T]} \langle \ell_t(\textstyle\sum_{t'\in[T]}\boldsymbol{\mu}_{t'}^\star),\, \boldsymbol{\mu}_t - \boldsymbol{\mu}_t^\star\rangle \geq 0, \quad \forall\, (\boldsymbol{\mu}_{t'})_{t'\in[T]} \in \Pi_{t'\in[T]}\Delta_{t'} \tag{2.32}$$

We can write the above problem in terms of edge flows, which consists in finding $(\mathbf{x}_{t'}^\star)_{t'\in[T]} \in \Pi_{t'\in[T]}\mathbf{M}\Delta_{t'}$ such that

$$\sum_{t\in[T]} \langle F_t(\textstyle\sum_{t'\in[T]}\mathbf{x}_{t'}^\star),\, \mathbf{x}_t - \mathbf{x}_t^\star\rangle \geq 0, \quad \forall\, (\mathbf{x}_{t'})_{t'\in[T]} \in \Pi_{t'\in[T]}\mathbf{M}\Delta_{t'} \tag{2.33}$$

If we concatenate along the different types of drivers to form a general path flow vector $\tilde{\boldsymbol{\mu}} = (\boldsymbol{\mu}_t)_{t\in[T]}$, and define the general path cost function $\tilde{\ell}(\tilde{\boldsymbol{\mu}}) = (\ell_t(\sum_{t'\in[T]} \boldsymbol{\mu}_{t'}))_{t\in[T]}$, then (2.32) can be written in the form of a variational inequality problem, which consists in finding $\tilde{\boldsymbol{\mu}}^\star \in \Pi_{t'\in[T]}\Delta_{t'}$ such that

$$\langle \tilde{\ell}(\tilde{\boldsymbol{\mu}}^\star),\, \tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{\mu}}^\star \rangle \geq 0, \quad \forall\, \tilde{\boldsymbol{\mu}} \in \Pi_{t'\in[T]}\Delta_{t'} \tag{2.34}$$

If we define the general edge flow vector $\tilde{\mathbf{x}} = (\mathbf{x}_t)_{t\in[T]}$ and the general edge cost function $F(\tilde{\mathbf{x}}) = (F_t(\sum_{t'\in[T]} \mathbf{x}_{t'}))_{t\in[T]}$, the problem (2.33) can be formulated as a variational inequality problem, which consists in finding $\tilde{\mathbf{x}}^\star \in \Pi_{t'\in[T]}\mathbf{M}\Delta_{t'}$

$$\langle F(\tilde{\mathbf{x}}^\star),\, \tilde{\mathbf{x}} - \tilde{\mathbf{x}}^\star \rangle \geq 0, \quad \forall\, \tilde{\mathbf{x}} \in \Pi_{t'\in[T]}\mathbf{M}\Delta_{t'} \tag{2.35}$$

However, the heterogeneous routing game is in general not a potential game since the symmetry condition (2.25) in the heterogeneous case is not satisfied. We have, for all $e \in \mathcal{E}$, $\forall\, t, t' \in [T]$

$$\frac{\partial c_{t,e}(\sum_{u\in[T]} x_{u,e})}{\partial x_{t',e}} = c'_{t,e}(\textstyle\sum_{u\in[T]} x_{u,e})$$

Hence the symmetry condition is not satisfied in general unless the travel costs for each type of driver are the same modulo a constant term.

Chapter 3

Computational aspects

In general, the optimal solution of a convex optimization program (2.2) or a variational inequality problem (2.5) is unknown. Fortunately, with additional smooth assumptions on the convex potential f or on the operator F, it is possible to use gap functions to obtain certificate of how well $\mathbf{x} \in \mathbb{R}^n$ approximates an optimal solution $\mathbf{x}^\star$. These gap certificates are used as stopping criteria for iterative algorithms.

This chapter focuses on the Frank-Wolfe algorithm (*a.k.a.* the conditional gradient algorithm), which is a popular iterative descent algorithm for solving the traffic assignment problem (2.19). Specifically, exploiting the sparsity structure of the traffic assignment problem enables to reduce the problem of computing the search direction of the Frank-Wolfe algorithm to deriving shortest paths with weights equal to the travel costs at the current iteration. We also provide convergence rates on the Frank-Wolfe algorithm that extend the proof of Jaggi in [87].

3.1 Gap functions

Let $\mathcal{D}$ be a compact convex subset of $\mathbb{R}^n$, and let $F : \mathcal{D} \to \mathbb{R}^n$ be a continuous mapping. We present gap functions in the context of the variational inequality problem given in (2.9)

$$\text{find } \mathbf{x}^\star \in \mathcal{D} \quad \text{s.t.} \quad \langle F(\mathbf{x}^\star), \mathbf{x} - \mathbf{x}^\star \rangle \geq 0 \quad \forall\, \mathbf{x} \in \mathcal{D}$$

A gap function must satisfy a *sub-optimality certificate* property for the variational inequality problem, namely, for every $\mathbf{x} \in \mathcal{D}$

$$g(\mathbf{x}) \geq 0 \quad \text{and} \quad g(\mathbf{x}) = 0 \;\Leftrightarrow\; \mathbf{x} \text{ solves the variational inequality problem} \tag{3.1}$$

We present properties on the gap functions introduced in [94, 19, 150], which are defined by, for each $\mathbf{x} \in \mathcal{D}$,

$$g^{'}(\mathbf{x}) = \max_{\mathbf{z}\in\mathcal{D}} \langle \mathbf{x}-\mathbf{z},\, F(\mathbf{x})\rangle \tag{3.2}$$

$$g^{''}(\mathbf{x}) = \min_{\boldsymbol{\nu}\in\mathbb{R}^p\,:\,\mathbf{A}^T\boldsymbol{\nu}\leq F(\mathbf{x})} F(\mathbf{x})^T\mathbf{x} - \mathbf{b}^T\boldsymbol{\nu} \tag{3.3}$$

$$g^{'''}(\mathbf{x}) = \min_{\substack{\boldsymbol{\nu}\in\mathbb{R}^p,\,\boldsymbol{\pi},\mathbf{s}\in\mathbb{R}^n_+:\\ \mathbf{s}=F(\mathbf{x})-\mathbf{A}^T\boldsymbol{\nu}}} \|(\alpha(\mathbf{s}-\boldsymbol{\pi}),\, \mathbf{x}\circ\boldsymbol{\pi})\|_1 \tag{3.4}$$

where the vector $\mathbf{x}\circ\boldsymbol{\pi} \in \mathbb{R}^n$ is the element-wise product between $\mathbf{x}$ and $\boldsymbol{\pi}$ (Hadamard product), and $\alpha \in \mathbb{R}_{>0}$ in (3.4) controls the weight of $\mathbf{s}-\boldsymbol{\pi}$ in the $2n$-vector $(\alpha(\mathbf{s}-\boldsymbol{\pi}),\, \mathbf{x}\circ\boldsymbol{\pi})$. In (3.3) and (3.4), we have assumed that $\mathcal{D}$ is polyhedral of the form $\mathcal{D} = \{\mathbf{x}\in\mathbb{R}^n_+ : \mathbf{A}\mathbf{x} = \mathbf{b}\}$, where $\mathbf{A}\in\mathbb{R}^{p\times n}$ and $\mathbf{b}\in\mathbb{R}^p$. Note that (3.4) is the sum of the absolute value of the residuals of the Kharush-Kuhn-Tucker (KKT) conditions, see [94]

$$g^{'''}(\mathbf{x}) = \min_{\substack{\boldsymbol{\nu}\in\mathbb{R}^p,\,\boldsymbol{\pi},\mathbf{s}\in\mathbb{R}^n_+:\\ \mathbf{s}=F(\mathbf{x})-\mathbf{A}^T\boldsymbol{\nu}}} \textstyle\sum_{i=1}^n \{\alpha\,|s_i-\pi_i| + |x_i\pi_i|\}$$

We note that $g^{'}$ satisfies the certificate property (3.1) directly from its definition. Adapting the proofs in [19, 150], we can show that $g^{''}$ and $g^{'''}$ are 'equivalent' to $g^{'}$ (similar to norm equivalence), and thus conclude that they also satisfy property (3.1).

Proposition 3.1. *Let $\mathcal{D}$ be compact and polyhedral of the form $\mathcal{D} = \{\boldsymbol{x}\in\mathbb{R}^n_+ : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\}$, and denote its diameter by $\lambda = diam_{\|\cdot\|_\infty}(\mathcal{D})$. Then, for every $\boldsymbol{x}\in\mathcal{D}$,*

$$g^{'''}(\boldsymbol{x}) \leq g^{'}(\boldsymbol{x}) = g^{''}(\boldsymbol{x}) \leq g^{'''}(\boldsymbol{x})\,\max(1,\,\lambda/\alpha)$$

Proof. To prove that $g^{'} = g^{''}$, we observe that $g^{'}$ can be rewritten as $g^{'}(\mathbf{x}) = \mathbf{x}^T F(\mathbf{x}) - \min_{\mathbf{z}\in\mathcal{D}} \mathbf{z}^T F(\mathbf{x})$. Since the minimization program involved in $g^{'}$ is linear and admits an optimal solution from compactness of $\mathcal{D}$, the strong duality theorem for linear program gives, for every $\mathbf{x}\in\mathcal{D}$

$$\min_{\mathbf{z}\in\mathbb{R}^n_+\,:\,\mathbf{A}\mathbf{z}=\mathbf{b}} \mathbf{z}^T F(\mathbf{x}) = \max_{\boldsymbol{\nu}\in\mathbb{R}^p\,:\,\mathbf{A}^T\boldsymbol{\nu}\leq F(\mathbf{x})} \mathbf{b}^T\boldsymbol{\nu}$$

from which we obtain $g^{'} = g^{''}$. To prove $g^{'''} \leq g^{''} \leq g^{'''}\max(1,\lambda/\alpha)$, we observe that $F(\mathbf{x})^T\mathbf{x} - \mathbf{b}^T\boldsymbol{\nu} = (F(\mathbf{x}) - \mathbf{A}^T\boldsymbol{\nu})^T\mathbf{x}$, for every $\mathbf{x}\in\mathcal{D}$. So we can rewrite,

$$g^{''}(\mathbf{x}) = \min_{\substack{\boldsymbol{\nu}\in\mathbb{R}^p,\,\mathbf{s}\in\mathbb{R}^n_+:\\ \mathbf{s}=F(\mathbf{x})-\mathbf{A}^T\boldsymbol{\nu}}} \mathbf{s}^T\mathbf{x}$$

To compare against $g^{'''}$, we re-write the latter as,

$$g^{'''}(\mathbf{x}) = \min_{\substack{\boldsymbol{\nu}\in\mathbb{R}^p,\, \mathbf{s}\in\mathbb{R}^n_+: \\ \mathbf{s}=F(\mathbf{x})-\mathbf{A}^T\boldsymbol{\nu}}} \min_{\boldsymbol{\pi}\in\mathbb{R}^n_+} \|(\alpha(\mathbf{s}-\boldsymbol{\pi}),\, \mathbf{x}\circ\boldsymbol{\pi})\|_1$$

We are left with comparing $\min_{\boldsymbol{\pi}\in\mathbb{R}^n_+} \|(\alpha(\mathbf{s}-\boldsymbol{\pi}),\, \mathbf{x}\circ\boldsymbol{\pi})\|_1$ and $\mathbf{s}^T\mathbf{x}$ for every $\mathbf{x}, \boldsymbol{\pi}, \mathbf{s} \in \mathbb{R}^n_+$, and $\alpha \in \mathbb{R}_{>0}$, since taking the infimum preserves the sense of an inequality. If we denote the vector $(\min(x_i,\, \alpha))_{i\in[n]}$ by $\bar{\mathbf{x}}$, it can be proven that $\min_{\boldsymbol{\pi}\in\mathbb{R}^n_+} \|(\alpha(\mathbf{s}-\boldsymbol{\pi}),\, \mathbf{x}\circ\boldsymbol{\pi})\|_1 = \mathbf{s}^T\bar{\mathbf{x}}$, which is less than $\mathbf{s}^T\mathbf{x}$, hence $g^{'''} \leq g^{''}$. Finally, proving $g^{''} \leq g^{'''}\max(1, \lambda)$ reduces to showing that $\mathbf{s}^T\mathbf{x} \leq \mathbf{s}^T(\max(1, \lambda/\alpha)\,\bar{\mathbf{x}})$. If $\lambda = \mathrm{diam}_{\|\cdot\|_\infty}(\mathcal{D}) \leq \alpha$, the inequality is an equality because $\max(1, \lambda/\alpha) = 1$ and $\bar{\mathbf{x}} = \mathbf{x}$. In the case when $\lambda = \mathrm{diam}_{\|\cdot\|_\infty}(\mathcal{D}) > \alpha$, then $\max(1, \lambda/\alpha) = \lambda/\alpha > 1$, and we prove that $\mathbf{x} \leq (\lambda/\alpha)\,\bar{\mathbf{x}}$ by observing that,

$$x_i \leq \min\left(\tfrac{\lambda}{\alpha}x_i,\, \lambda\right) = \tfrac{\lambda}{\alpha}\min(x_i,\, \alpha) = \tfrac{\lambda}{\alpha}\bar{x}_i \quad \forall\, i \in [n]$$

This completes our proof. □

In addition, when F is strongly monotone, the gap functions control the distance to the unique solution of the variational inequality problem.

Proposition 3.2. *Let $\mathcal{D}$ be a compact convex subset of $\mathbb{R}^n$, let F be strongly monotone with parameter c, and let $\boldsymbol{x}^\star$ be the unique solution to the variational inequality problem* (2.9)*. Then, for every $\boldsymbol{x} \in \mathcal{D}$,*

$$\|\boldsymbol{x} - \boldsymbol{x}^\star\|_2^2 \leq g^{'}(\boldsymbol{x})/c$$

Proof. By definition of $g^{'}(\cdot)$, we have for every $\mathbf{x} \in \mathcal{D}$, $\langle \mathbf{x} - \mathbf{x}^\star,\, F(\mathbf{x})\rangle \leq g^{'}(\mathbf{x})$ and $\langle \mathbf{x}^\star - \mathbf{x},\, F(\mathbf{x}^\star)\rangle \leq g^{'}(\mathbf{x}^\star)$. Adding the two, $\langle \mathbf{x}^\star - \mathbf{x},\, F(\mathbf{x}^\star) - F(\mathbf{x})\rangle \leq g^{'}(\mathbf{x}) + g^{'}(\mathbf{x}^\star)$. Observing that $g^{'}(\mathbf{x}^\star) = 0$ by optimality of $\mathbf{x}^\star$, and $\langle \mathbf{x}^\star - \mathbf{x},\, F(\mathbf{x}^\star) - F(\mathbf{x})\rangle$ is lower bounded by $c\|\mathbf{x}^\star - \mathbf{x}\|$ by strong monotonicity of F, we obtain the claimed bound. □

Combining Proposition 3.2 and Proposition 3.1, we obtain the following corollary:

Corollary 3.1. *Let $\mathcal{D}$ be compact and polyhedral of the form $\mathcal{D} = \{\boldsymbol{x} \in \mathbb{R}^n_+ : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\}$, and denote its diameter by $\lambda = diam_{\|\cdot\|_\infty}(\mathcal{D})$. Let F be strongly monotone with parameter c, and let $\boldsymbol{x}^\star$ be the unique solution to the variational inequality problem* (2.9)*. Then, for every $\boldsymbol{x} \in \mathcal{D}$,*

$$\|\boldsymbol{x} - \boldsymbol{x}^\star\|_2^2 \leq g^{''}(\boldsymbol{x})/c$$
$$\|\boldsymbol{x} - \boldsymbol{x}^\star\|_2^2 \leq g^{'''}(\boldsymbol{x})\max(1,\, \lambda/\alpha)/c$$

We note that the results of Proposition 3.1, Proposition 3.2, and Corollary 3.1 are applicable to the convex optimization program given in (2.7)

$$\min\ f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{X}$$

where strong monotonicity of F is replaced with strong convexity of the potential function $f : \mathcal{X} \to \mathbb{R}$, and in the definition of the gap functions $g^{'}$, $g^{''}$, $g^{'''}$ in (3.2), (3.3), and (3.4), the mapping F is replaced with the gradient of the potential ∇f. In addition, we have the following result:

Proposition 3.3. *Let $\mathcal{X}$ be a compact convex subset of $\mathbb{R}^n$, let f be a convex function, and let $\boldsymbol{x}^\star$ be any optimal solution to the convex optimization program* (2.7)*. The gap function $\tilde{g}(\boldsymbol{x}) := \max_{\boldsymbol{z}\in\mathcal{X}} \langle \boldsymbol{x} - \boldsymbol{z}, \nabla f(\boldsymbol{x})\rangle$ then satisfies, for all $\boldsymbol{x} \in \mathcal{X}$*

$$\tilde{g}(\boldsymbol{x}) \geq f(\boldsymbol{x}) - f(\boldsymbol{x}^\star) \tag{3.5}$$

Proof. By convexity of f, we have

$$f(\mathbf{x}') \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}),\, \mathbf{x}' - \mathbf{x}\rangle, \quad \forall\, \mathbf{x},\, \mathbf{x}' \in \mathcal{D}$$

In particular, $f(\mathbf{x}^\star) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}),\, \mathbf{x}^\star - \mathbf{x}\rangle$, where $\mathbf{x}^\star$ is a solution to the convex program

$$\min\ f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{D}$$

Since $\mathbf{x}^\star$ is also solution to the variational inequality problem (2.9) with $F = \nabla f$, from the first-order optimality condition, we have

$$\tilde{g}(\mathbf{x}) \geq \langle \nabla f(\mathbf{x}),\, \mathbf{x} - \mathbf{x}^\star\rangle \geq f(\mathbf{x}) - f(\mathbf{x}^\star) \tag{3.6}$$

where the first inequality is from the definition of the duality gap $\tilde{g}$, and the second inequality is from the convexity of f. Hence $\tilde{g}$ is also a sub-optimality certificate for the convex program. □

3.2 Frank-Wolfe algorithm applied to the routing game

The homogeneous selfish routing game has the general potential form

$$\min\ f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{D} \tag{3.7}$$

where f is convex and continuously differentiable potential, and the domain $\mathcal{D}$ is a compact convex subset of a Hilbert space $\mathcal{X}$. We note that in the context of the routing game, the convexity and differentiability assumptions of the potential function hold when the edge cost functions are increasing and continuous, see Section 6.2 for more details.

Algorithm 3.1 Frank-Wolfe algorithm

1. Initialize with $\mathbf{x}_0 \in \mathcal{D}$ and let $k := 0$
2. Get a search direction $\mathbf{d}_k = \mathbf{y}_k - \mathbf{x}_k$ by solving the LP $\mathbf{y}_k \in \operatorname{argmin}_{\mathbf{y}\in\mathcal{D}}\{\nabla f(\mathbf{x}_k)^T\mathbf{y}\}$
3. Choose step length $\alpha_k \in [0,1]$ such that $f(\mathbf{x}_k + \alpha_k\mathbf{d}_k) = f((1-\alpha)\mathbf{x}_k + \alpha\mathbf{y}_k) < f(\mathbf{x}_k)$
4. Update $\mathbf{x}_{k+1} = (1-\alpha_k)\mathbf{x}_k + \alpha_k\mathbf{y}_k$
5. Let $k := k+1$ and go to step 2.

The Frank-Wolfe algorithm is among the earliest documented iterative optimization algorithms for solving constrained convex programs. It has been proposed in 1956 by Frank and Wolfe [68]. The algorithm is described in Algorithm 3.2. It marks a historical departure from linear programming, going to quadratic programming and convex optimization.

We note that at every iteration of the Frank-Wolfe algorithm, the duality gap (3.1) is computed automatically at the iterate $\mathbf{x}_k$ since

$$g'(\mathbf{x}_k) = \max_{\mathbf{y}\in\mathcal{D}} \nabla f(\mathbf{x}_k)^T(\mathbf{x}_k - \mathbf{y}) = \nabla f(\mathbf{x}_k)^T\mathbf{x}_k - \min_{\mathbf{y}\in\mathcal{D}} \nabla f(\mathbf{x}_k)^T\mathbf{y}$$

A common stopping criterion is, for a fixed $\epsilon > 0$,

$$g'(\mathbf{x}_k) = \nabla f(\mathbf{x}_k)^T(\mathbf{x}_k - \mathbf{y}_k) \leq \epsilon \tag{3.8}$$

With the additional assumption that f is continuously differentiable and strongly convex with parameter c, (3.8) implies that $\|\mathbf{x}_k - \mathbf{x}^\star\|_2^2 \leq \epsilon/c$ from Proposition 3.2, where $\mathbf{x}^\star$ is an optimal solution to the optimization program (3.7).

The Frank-Wolfe algorithm is directly suitable for solving "sparse" convex problems, where the optimal solution is a linear combination of a few vertices of the feasible domain, since each iteration adds one new vertex. In the case of the traffic assignment problem, each vertex of the feasible domain Δ defined in (2.12) consists in assigning the mass d_k of a population k to a simple path $p \in \mathcal{P}_k$ from an origin $s_k \in \mathcal{V}$ to a destination $t_k \in \mathcal{V}$. Since the gradient of the potential function (2.28) with respect to path flows is given by $\nabla_{\boldsymbol{\mu}} f(\boldsymbol{\mu}) = (\ell_p(\boldsymbol{\mu}))_{p\in\mathcal{P}}$, then $\boldsymbol{\nu}^T\nabla_{\boldsymbol{\mu}} f(\boldsymbol{\mu}) = \sum_{k\in[K]}\sum_{p\in\mathcal{P}_k} \ell_p(\boldsymbol{\mu})\,\nu_p$, and the quantity $\mathbf{y}_k$ in the descent direction $\mathbf{d}_k = \mathbf{y}_k - \mathbf{x}_k$ in step 2 of the Frank-Wolfe algorithm is solution to:

$$\min_{\boldsymbol{\nu}} \sum_{k\in[K]}\sum_{p\in\mathcal{P}_k} \ell_p(\boldsymbol{\mu})\,\nu_p \quad \text{s.t.} \quad \sum_{p\in\mathcal{P}_k} \nu_p = d_k,\ \forall\, k \in [K],\ \boldsymbol{\nu} \succeq 0$$

where $\boldsymbol{\mu}$ is the current iterate. The above problem is separable by population $k \in [K]$

$$\min_{\boldsymbol{\nu}^k} \sum_{p\in\mathcal{P}_k} \ell_p(\boldsymbol{\mu})\,\nu_p \quad \text{s.t.} \quad \sum_{p\in\mathcal{P}_k} \nu_p = d_k,\ \boldsymbol{\nu}^k \succeq 0$$

The above problem consists in assigning the population mass d_k to a shortest path between s_k and t_k. Hence, computing the search direction reduces to finding the shortest path between each pair of origin and destination, which can be obtained with Dijkstra's algorithm.

We have implemented a solver for the homogeneous routing game available in `github.com/megacell/python-traffic-assignment/blob/master/frank_wolfe_2.py`. The method `solver` implements the Frank-Wolfe algorithm where the step size α_k is set to $\alpha_k := \frac{2}{k+2}$ at the k-th iteration. We show in the next Section that, under some additional smoothness assumptions, the Frank-Wolfe algorithm converges in $O(1/k)$ where k

The search direction (step 2 in the Frank-Wolfe algorithm above) is computed using the `get_shortest_paths` method from the `python-igraph` package, which implementation is available in `github.com/megacell/python-traffic-assignment/blob/master/AoN_igraph.py`.

3.3 Convergence analysis of the Frank-Wolfe algorithm

Following the approach in [87], we define a measure of "non-linearity" of our objective function f over the domain $\mathcal{D}$. The *curvature constant* C of a convex and differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ with respect to a compact domain $\mathcal{D}$ is defined as

$$C := \sup_{\mathbf{x},\mathbf{s}\in\mathcal{D},\gamma\in[0,1]} \frac{2}{\gamma^2}(f(\mathbf{y}) - f(\mathbf{x}) - \langle \mathbf{y}-\mathbf{x}, \nabla f(\mathbf{x})\rangle) \tag{3.9}$$

The definition of C implies that, for all $\mathbf{x}, \mathbf{s} \in \mathcal{D}$, and for all $\gamma \in [0, 1]$

$$f(\mathbf{x} + \gamma(\mathbf{s}-\mathbf{x})) \leq f(\mathbf{x}) + \gamma\langle \mathbf{s}-\mathbf{x}, \nabla f(\mathbf{x})\rangle + \frac{\gamma^2}{2}C$$

Hence, the curvature C bounds by how much the function f at the next iterate $\mathbf{x} + \gamma(\mathbf{s}-\mathbf{x})$ deviates from the linearization of f given by $\nabla f(\mathbf{x})$ at $\mathbf{x}$, where the bound is given by a quadratic function $\gamma \mapsto \frac{\gamma^2}{2}C$. We also note that for linear functions, the curvature constant C is zero.

Algorithm 3.2 Frank-Wolfe algorithm with approximate linear subproblems [87]

1. Initialize with $\mathbf{x}_0 \in \mathcal{D}$, let $\alpha > 1$, and let $k := 0$
2. Let $\gamma := \frac{\alpha}{k+\alpha}$
3. Get a search direction $\mathbf{d}_k = \mathbf{y}_k - \mathbf{x}_k$ such that $\mathbf{y}_k^T \nabla f(\mathbf{x}_k) \leq \min_{\mathbf{y}\in\mathcal{D}} \mathbf{y}^T \nabla f(\mathbf{x}_k) + \frac{1}{2}\delta\gamma C$
4. Update $\mathbf{x}_{k+1} = (1-\gamma)\,\mathbf{x}_k + \gamma\,\mathbf{y}_k$
5. Let $k := k+1$ and go to step 2.

Lemma 3.1. *Let $\alpha > 1$. Assume $a_{k+1} \leq (1-\gamma_k)\,a_k + \gamma_k^2\,\delta$ for $\delta > 0$. If $\gamma_k = \frac{\alpha}{k+\alpha}$ then $a_k \leq \gamma_k \frac{\delta\,\alpha}{\alpha-1}$.*

Proof. We first look for a constant $C > 0$ such that

$$a_k \leq \gamma_k\, C \tag{3.10}$$

where we assume that $\gamma_k = \frac{\alpha}{k+\alpha}$, for $k \geq 1$. We then show that it is sufficient to have $C \leq \frac{\delta\,\alpha}{\alpha-1}$.

For the base case $k = 0$, we have $a_1 \leq (1-\gamma_0)\,a_0 + \gamma_0^2\,\delta = \delta$. Hence, a sufficient condition to have (3.10) satisfied for $k = 1$, *i.e.* $a_1 \leq C\gamma_1 = \frac{\alpha\,C}{1+\alpha}$, is $\delta \leq \frac{\alpha\,C}{1+\alpha}$, or equivalently

$$C \geq \frac{(1+\alpha)\,\delta}{\alpha} \tag{3.11}$$

We now assume that $a_k \leq \gamma_k\,C = \frac{\alpha\,C}{k+\alpha}$ for $k \geq 1$, and derive a sufficient condition on C so that $a_{k+1} \leq \frac{\alpha\,C}{k+1+\alpha}$. We have

$$\begin{aligned}
a_{k+1} &\leq (1-\gamma_k)\,a_k + \gamma_k^2\,\delta \\
&\leq \left(1 - \frac{\alpha}{k+\alpha}\right)\frac{\alpha\,C}{k+\alpha} + \left(\frac{\alpha}{k+\alpha}\right)^2\delta \\
&= \left(1 - \frac{\alpha}{k+\alpha} + \frac{\delta\,\alpha}{C(k+\alpha)}\right)\frac{\alpha\,C}{k+\alpha} \\
&= \frac{\alpha\,C}{k+\alpha}\frac{k+(\delta/C)\alpha}{k+\alpha}
\end{aligned}$$

A sufficient condition for (3.10) to be satisfied for the step $k+1$ is

$$\frac{\alpha\,C}{k+\alpha}\frac{k+(\delta/C)\alpha}{k+\alpha} \leq \frac{\alpha\,C}{k+1+\alpha}$$

or equivalently

$$\frac{\delta\,\alpha}{C} \leq \frac{(k+\alpha)^2}{k+1+\alpha} - k \tag{3.12}$$

Hence we would like the above inequality to be satisfied for every $k \geq 1$. We define the function $f(x) = \frac{(x+\alpha)^2}{x+1+\alpha} - x$ for $x \geq 1$. Its derivative is

$$f'(x) = \frac{(x+\alpha)(x+\alpha+2)}{(x+1+\alpha)^2} - 1$$

From the inequality between geometric and arithmetic means, we have $\sqrt{(x+\alpha)(x+\alpha+2)} < \frac{(x+\alpha)+(x+\alpha+2)}{2} = x+1+\alpha$ for all $x \geq 1$, hence $\frac{(x+\alpha)(x+\alpha+2)}{(x+1+\alpha)^2} \leq 1$ for $x \geq 1$, and f is decreasing on $[1, +\infty)$. In addition, $f(x)$ can be re-written as

$$f(x) = \frac{(\alpha-1)x + \alpha^2}{x+1+\alpha} \underset{x\to+\infty}{\longrightarrow} \alpha - 1$$

thus $f(x) \geq \alpha - 1$ for all $x \geq 1$. Hence, a sufficient condition for having $a_{k+1} \leq \frac{\alpha\,C}{k+1+\alpha}$ for $k \geq 1$ is $\frac{\delta\,\alpha}{C} \leq \alpha - 1$, or equivalently

$$C \geq \frac{\delta\,\alpha}{\alpha-1}$$

Noting that the above sufficient condition also implies (3.11) completes the proof. □

Theorem 3.1. *For each $k \geq 1$, the iterates $\boldsymbol{x}_k$ of Algorithm 3.2 satisfy, for $\alpha > 1$*

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}^\star) \leq \frac{\alpha\, C}{k+\alpha} \tag{3.13}$$

where $\boldsymbol{x}^\star \in \mathcal{D}$ is an optimal solution to problem (3.7).

Proof. We have

$$\begin{aligned} f(\mathbf{x}_{k+1}) &= f((1-\gamma)\mathbf{x}_k + \gamma \mathbf{y}_k) \\ &\leq f(\mathbf{x}_k) + \gamma\langle \mathbf{y}_k - \mathbf{x}_k, \nabla f(\mathbf{x}_k)\rangle + \frac{\gamma^2}{2}C \\ &\leq f(\mathbf{x}_k) - \gamma \tilde{g}(\mathbf{x}_k) + \frac{\gamma^2}{2}C \\ &\leq f(\mathbf{x}_k) - \gamma(f(\mathbf{x}_k) - f(\mathbf{x}^\star)) + \frac{\gamma^2}{2}C \end{aligned}$$

where the first inequality is from the definition of the curvature constant C of our convex function f, the second inequality is from the definition (3.2) of the duality gap $\tilde{g}(\mathbf{x}) = \max_{\mathbf{z}\in\mathcal{D}} \langle \mathbf{x} - \mathbf{z}, \nabla f(\mathbf{x})\rangle$ (where the map F is subtituted with ∇f), and the third inequality is from $\tilde{g}(\mathbf{x}) \geq f(\mathbf{x}) - f(\mathbf{x}^\star)$ in Proposition 3.3. This leads to the following inequality:

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}^\star) \leq (1-\gamma)(f(\mathbf{x}_k) - f(\mathbf{x}^\star)) + \frac{\gamma^2}{2}C$$

Using Lemma 3.1 and the fact that γ is set to $\frac{\alpha}{k+\alpha}$ in the k-th iteration of Algorithm 3.2, we get the bound (3.13).

□

Combining the above results with Proposition 2.1, we can derive a bound on the iterates $\mathbf{x}_k$:

Lemma 3.2. *If f is strongly monotone with parameter c, then for each $k \geq 1$, the iterations $\boldsymbol{x}_k$ of Algorithm 3.2 satisfy, for $\alpha > 1$*

$$\|\boldsymbol{x}_k - \boldsymbol{x}^\star\|_2^2 \leq \frac{2}{c}\frac{\alpha C}{k+\alpha} \tag{3.14}$$

3.4 Frank-Wolfe algorithm applied to the heterogeneous game

In general, the heterogeneous selfish routing game cannot be written in potential form, see Section 2.6 for more details. It can be written as a variational inequality problem (2.34). We recall the general formulation of a variational inequality problem. Given $\mathcal{D}$ a compact convex

subset of $\mathbb{R}^n$, and $F : \mathcal{D} \to \mathbb{R}^n$ a continuous mapping, the problem consists in finding $\mathbf{x}^\star \in \mathcal{D}$ such that

$$\langle F(\mathbf{x}^\star),\, \mathbf{x} - \mathbf{x}^\star \rangle \geq 0, \quad \forall\, \mathbf{x} \in \mathcal{D} \tag{3.15}$$

In Section 4.3.1. of her work [76], Hammond proposes the "generalized fictitious play algorithm":

Algorithm 3.3 Generalized fictitious play algorithm

1. Initialize with $\mathbf{x}_0 \in \mathcal{D}$ and let $k := 0$
2. Get a search direction $\mathbf{d}_k = \mathbf{y}_k - \mathbf{x}_k$ by solving the LP $\mathbf{y}_k \in \operatorname{argmin}_{\mathbf{y} \in \mathcal{D}} \{F(\mathbf{x}_k)^T \mathbf{y}\}$
3. Set $\mathbf{x}_{k+1} := \frac{1}{k+1} \sum_{i=0}^{k} \mathbf{y}_i$
4. Let $k := k + 1$ and go to step 2.

The fictitious play algorithm can be seen as the Frank-Wolfe algorithm where the line search is replaced by the averaging over the previous iterates. Hammond shows that when the mapping F is continuously differentiable and monotone, the domain $\mathcal{D}$ is compact and strongly convex, and there is no point $\mathbf{x} \in \mathcal{D}$ such that $f(\mathbf{x}) = 0$, then the iterates converge to the solution of the variational inequality problem, see Theorem 4.6 in [76].

However, to our knowledge, there is no convergence results in the case of the heterogeneous routing game expressed as a variational inequality problem (2.34), since the domain $\mathcal{D}$ is a bounded polyhedron (and not strongly convex), where each vertex is a path between the source and the destination of some population. Hammond still conjectured that for a variational inequality problem where F is uniformly monotone and $\mathcal{D}$ is a bounded polyhedron, the fictitious play algorithm will solve the variational inequality problem.

Even though there is no convergence guarantee, we have implemented a solver for the heterogeneous routing game available in `github.com/megacell/python-traffic-assignment/blob/master/frank_wolfe_heterogeneous.py`. In the case of the heterogeneous traffic assignment problem, each vertex of the feasible domain $\tilde{\Delta} := \Pi_{t \in [T]} \Delta_t$, where Δ_t is defined in (2.30) consists in assigning the mass $d_{t,k}$ of a population k of type t to a simple path $p \in \mathcal{P}_k$ from an origin $s_k \in \mathcal{V}$ to a destination $t_k \in \mathcal{V}$.

Given the general path flow vector $\tilde{\boldsymbol{\mu}} = (\boldsymbol{\mu}_t)_{t \in [T]} \in \Pi_{t \in [T]} \Delta_t$, the mapping in the heterogenous routing game is given by

$$\tilde{\ell}(\tilde{\boldsymbol{\mu}}) = (\ell_t(\tilde{\boldsymbol{\mu}}))_{t \in [T]} = (\ell_{t,p}(\tilde{\boldsymbol{\mu}}))_{p \in \mathcal{P},\, t \in [T]} = (\ell_{t,p}(\textstyle\sum_{t' \in [T]} \boldsymbol{\mu}_{t'}))_{p \in \mathcal{P},\, t \in [T]}$$

getting the search direction in the generalized fictitious play algorithm consists in solving

$$\min_{\boldsymbol{\nu}} \textstyle\sum_{t \in [T]} \sum_{k \in [K]} \sum_{p \in \mathcal{P}_k} \ell_{t,p}(\tilde{\boldsymbol{\mu}})^T \nu_{t,p} \quad \text{s.t.} \quad \sum_{p \in \mathcal{P}_k} \nu_{t,p} = d_{t,k},\ \forall\, k \in [K],\, \forall\, t \in [T],\ \boldsymbol{\nu} \succeq 0$$

where $\boldsymbol{\nu}$ is the current iterate. The above problem is separable by population $k \in [K]$ and type $t \in [T]$

$$\min_{\boldsymbol{\nu}} \textstyle\sum_{p \in \mathcal{P}_k} \ell_{t,p}(\boldsymbol{\mu})^T \nu_{t,p} \quad \text{s.t.} \quad \sum_{p \in \mathcal{P}_k} \nu_{t,p} = d_{t,k},\ \nu_{t,p} \geq 0,\, \forall\, p \in \mathcal{P}_k$$

The above problem consists in assigning the population mass $d_{t,k}$ to a shortest path between s_k and t_k. Hence, computing the search direction reduces to finding the shortest path between each pair of origin and destination, which can be obtained with Dijkstra's algorithm.

Chapter 4

Application: evaluating the impact of GPS-enabled shortest path routing on mobility

In this chapter, we use the selfish routing game framework to study the impact of the increasing penetration of routing apps on road usage. Our conclusions apply both to manned vehicles in which human drivers follow app directions, and unmanned vehicles following shortest path algorithms. To address the problem caused by the increased usage of routing apps, we model two distinct classes of users, one having limited knowledge of low-capacity road links. This approach is in sharp contrast with some previous studies assuming that each user has full knowledge of the network and optimizes his/her own travel time. We show that the increased usage of GPS routing provides a lot of benefits on the road network of Los Angeles, such as decrease in average travel times and total vehicle miles traveled. However, this global increased efficiency in urban mobility has negative impacts as well, which are not addressed by the scientific community: increase in traffic in cities bordering highway from users taking local routes to avoid congestion.

The organization of the present chapter is as follows: we introduce the concept of multiplicative cognitive cost to model non-routed users' preference for high-capacity roads and show that this choice is in general rational under low traffic demand; we expand on the established heterogeneous traffic assignment problem to quantify the road usage when there is a ratio α of routed users and $1-\alpha$ of non-routed users in the urban network; we finally show that the low-capacity network sees a significant increase in traffic pressuring local governments to build additional infrastructure to reduce the nuisance related to it.

4.1 Motivation

Navigation applications such as Google Maps, Waze, INRIX, or Apple Maps, have deeply modified our approach of driving in the past. Pushed by the increasing penetration of smart

phones and the rapid expansion of Mobility-as-a-Service systems such as Uber and Lyft, a significant percentage of drivers now use these tools daily, as they provide an easy way to optimize one's route choices and decrease one's travel time, specifically during peak hours. Since public agencies cannot indefinitely extend the capacity of urban road networks, these tools represent an opportunity to reallocate traffic in a way that might be more efficient (or not). Nonetheless, the impact of these applications on road traffic and urban congestion are not well-studied and understood. Cities bordering major highways in the United States have noticed an increase of traffic demand on their networks, presumably due to application users leaving highways to avoid congestion [66]. This alleged flow transfer is a challenge for public policy, as cities infrastructure, mostly financed by and for local taxpayers, receive a higher traffic demand.

The aim of the present chapter is to propose and develop a framework to describe heterogeneous traffic in which a percentage of drivers use these applications. The main research question is the following: "how does the percentage of application users impact traffic redistribution and corresponding optimality of flows assignment?"

Historically, high-capacity roads, *e.g.* expressways and highways, have been developed to improve safety, comfort, and traveling speed. Today, a vast majority of drivers will consciously choose an expressway over a smaller road, because of all the previous benefits. We thus assume that drivers, when traveling from an origin to a destination, will aim at minimizing the time spent on low-capacity (or low-speed) roads.

While we investigate the question of the impact of navigation applications on traffic, our framework encompasses heterogeneous traffic containing both classical manned vehicles and autonomous vehicles. Specifically, an autonomous vehicle can be modeled as a vehicle following real-time routing information, the same way a user follows instructions from routing services.

4.2 Approach and terminology

In order to address the research questions summarized above, we model the behavior of users on the road network with the established traffic assignment framework [130], in which each user traveling from their origin (*e.g.* their home) to their destination (*e.g.* their office) selfishly minimizes their own cost function. However, the transportation literature generally assumes that, for each user, the cost of traveling on a given route is the travel time of this route, see, *e.g.* [44]. Hence, state of the art work implicitly assumes that each user has access to the travel time of each link in the network and rationally chooses the shortest route to its destination. Contrasting from previous approaches, we model two types of users:

***Routed users*:** they have access to navigation information and thus follow the shortest route from their origin to their destination based on the network's current travel times. These vehicles can be drivers equipped with a GPS device (*e.g.* Garmin, TomTom, embedded navigation system), or a GPS-enabled mobile phone with a navigation app (*e.g.* Google maps, Waze, Apple maps), or they can be connected autonomous vehicle following routing

directions from navigation services. Hence, for *routed users*, the cost of using a route is its travel time. In addition, users with expert knowledge of the network are also considered as *routed users* since they are able to find shortest routes without the use of navigation apps.

***Non-routed users*:** They do not have access to updated traffic information and thus have a limited knowledge of the travel times in the network. Since highways traditionally enable to travel with limited information and provide perceived benefits such as safety and higher travel speeds, *non-routed users* are assumed to choose high-capacity roads over low-capacity ones. The precise mathematical model of the behavior of these *non-routed users* will be introduced below.

The lack of information of users has been addressed previously in the field of transportation [112], [69], and in economics [45]. They collectively describe *bounded rational* users who make suboptimal choices due to the lack and/or price of information. Since local roads are arguably less known while major highways are in the information set of most of users, we choose an approach similar to studies modeling users with different objective functions than just minimizing travel times, *e.g.* seeking out less congested or scenic routes [16]. However, instead of using the nested logit model [17], we model the preference of *non-routed users* for larger roads segments and their limited knowledge of small streets. Hence, we define two types of road segments:

High-capacity road segments: highways and major arterial roads and avenues. High-capacity roads mainly serve users just passing through or nearby the city to go to their destination, hence they are maintained at a county or state level. We also assume that *non-routed users* favor this type of roads since, with limited knowledge on the local network, they represent a convenient way to move towards the destination by following signs.

Low-capacity road segments: They include small residential streets and small arterial streets. The low-capacity network is maintained by local taxpayers and is designed to provide mobility to local users, who either live or work in the area. It was originally not meant by planners to be used by through traffic, which should be confined to the high-capacity network.

Multiplicative cognitive cost to encode user choice: We add a multiplicative factor $C > 1$ to low-capacity links' cost functions to model the preference of *non-routed users* for high-capacity links. The multiplicative cognitive cost conserves the proportions between low-capacity links' travel times and models users that want to reduce the time spent on low-capacity links in favor to high-capacity ones. We also show that, in the Los Angeles network, in free flow, preference for highways is rational since it enables the users to choose routes that are close to being optimal without the use of GPS routing.

Heterogeneous game: To study the increasing penetration of GPS routing, we consider a heterogeneous routing game with two types of users: *routed users* for which the cost of using an edge is the travel time, and *non-routed users* for which the cost of using an edge is the travel time if it is high-capacity, or C times the travel time if it is low-capacity. Heterogeneous games have been studied before for the purpose of designing toll strategies [91], [114], and in a more general setting in [63]. To our knowledge, this is the first use of heterogeneous games to model the impact of routing via navigation apps, on flow allocation.

Outline and contributions

The main contribution of this chapter is twofold. In Section II, we introduce the concept of *multiplicative cognitive cost* to model *non-routed users*' preference for high-capacity roads and show that this choice is in general rational under low traffic demand. However, during peak hours, we show that this preference results in a poor allocation of the traffic with higher travel times, thus encouraging app based routing. In Section III, we expand on the established heterogeneous traffic assignment problem to quantify the road usage when there is a ratio α of *routed users* and $1-\alpha$ of *non-routed users* in the urban network. We show that the use of app-based routing is rational since it decreases each user's travel time and allocates the flow efficiently throughout the network. However, this hidden cost is high as the low-capacity network sees a significant increase in traffic pressuring local governments to build additional infrastructure to reduce the nuisance related to it.

4.3 A Multiplicative Cognitive cost model

In this section, we present and motivate the multiplicative cognitive cost model using the traffic assignment framework.

Mathematical formulation and notations

We recall the selfish routing game framework. We consider a given road network modeled as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ and directed arc set $\mathcal{E}$. We have K populations indexed by $k \in [K]$, modeling a mass $d_k \in \mathbb{R}_+$ of drivers traveling from a common source $s_k \in \mathcal{V}$ to a common sink $t_k \in \mathcal{V}$. They choose between routes $p \in \mathcal{P}_k$ such that their travel cost is minimized, where $\mathcal{P}_k$ is the set of all paths from s_k to t_k. Hence, the state of the network is described by the vector of route flows $\boldsymbol{\mu} = (\mu_p)_{p\in\mathcal{P}} \in \mathbb{R}^{\mathcal{P}}$ where $\mathcal{P} = \sqcup_{k\in[K]}\mathcal{P}_k$ is the set of all paths in the network. A flow vector $\boldsymbol{\mu} \in \mathbb{R}^P$ is then feasible if for all $k \in [K]$, $\sum_{p\in P_k} \mu_p = d_k$ and $\mu_p \geq 0$, $\forall\, p \in \mathcal{P}_k$. In matrix form, $\boldsymbol{\mu}$ said to be feasible if it belongs to the following set

$$\Delta := \{\boldsymbol{\mu} \in \mathbb{R}^{\mathcal{P}} \,:\, \boldsymbol{\mu} \geq 0,\, \mathbf{A}\boldsymbol{\mu} = \mathbf{d}\} \tag{4.1}$$

where $\mathbf{A}$ is the population to path incidence matrix. *Non-routed users* have travel costs $\ell_p^{\mathrm{nr}}(\cdot)$ along each path $p \in \mathcal{P}$ given by $\ell_p^{\mathrm{nr}}(\boldsymbol{\mu}) = \sum_{e\in p} c_e^{\mathrm{nr}}(x_e)$ where $c_e^{\mathrm{nr}}(x_e)$ is the *non-routed users*' cost of edge e. We assume that the cost of a road segment e only depends on the flow x_e of vehicles on this segment, where x_e is expressed as

$$x_e = \textstyle\sum_{p\in\mathcal{P}:e\in p} \mu_p \tag{4.2}$$

the sum of the flows of every route passing through edge e. In matrix form, $\mathbf{x} = \mathbf{M}\boldsymbol{\mu}$ where the edge-path incidence matrix is given by $\mathbf{M} = [\mathbf{1}_{e\in p}]_{e\in\mathcal{E},\, p\in\mathcal{P}}$. Hence, we write that an edge

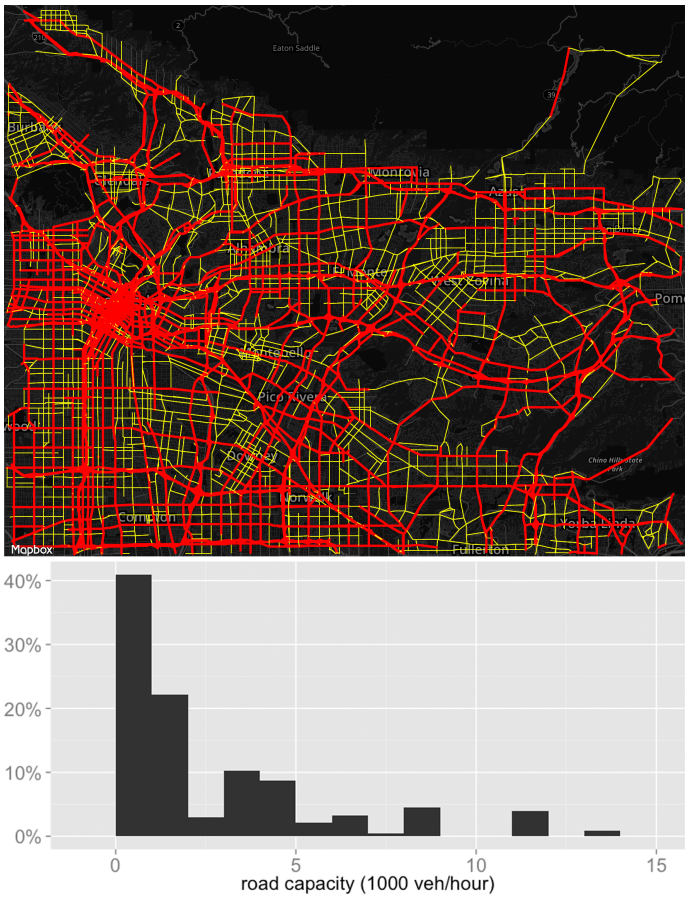

Figure 4.1: The map of Los Angeles, CA used for the present study composed of 28,376 arcs and 14,617 nodes extracted from OpenStreetMap. Information for each edge includes the free-flow travel time, length, capacity, and speed limit. Links with capacity less than 1000 vehicles per hour are considered low-capacity (in yellow) while links with 1000 vehicles per hour or more are considered high-capacity (red). The histogram of the different road capacities are shown in the bottom figure, with more than 40% of low-capacity links.

flow vector $\mathbf{x} = (x_e)_{e\in\mathcal{E}}$ is feasible if it is in the following set

$$\mathcal{D} := \mathbf{M}\Delta = \{\mathbf{x} \in \mathbb{R}_+^{\mathcal{E}} \,:\, \exists\, \boldsymbol{\mu} \in \Delta,\, \mathbf{x} = \mathbf{M}\boldsymbol{\mu}\} \tag{4.3}$$

We formalize the behavior of *non-routed users* by partitioning the edge set $\mathcal{E}$ into a set of low-capacity edges $\mathcal{E}^{\text{lo}} = \{e \in \mathcal{E} \,:\, m_e < m_{\text{lo}}\}$ and a set of high-capacity edges $\mathcal{E}^{\text{hi}} := \{e \in \mathcal{E} \,:\, m_e \geq m_{\text{lo}}\}$ where each edge has a capacity m_e and m_{lo} is an arbitrary threshold. Throughout our study, we consider road segments with capacities less than 1000 vehicles per hour as low-capacity, which amount for 40% of the road segments in the Los Angeles network, see

Figure 4.1. The *non-routed users*' costs are then

$$c_e^{\mathrm{nr}}(x_e) = \begin{cases} C \cdot t_e(x_e) & \text{if } e \in \mathcal{E}^{\mathrm{lo}} \\ t_e(x_e) & \text{if } e \in \mathcal{E}^{\mathrm{hi}} \end{cases} \tag{4.4}$$

This results in the following non-routed path costs

$$\ell_p^{\mathrm{nr}}(\boldsymbol{\mu}) = \sum_{e \in p^{\mathrm{hi}}} t_e(x_e) + C \sum_{e \in p^{\mathrm{lo}}} t_e(x_e) \tag{4.5}$$

where $t_e(x_e)$ is the travel time of road segment e under flow x_e, $C > 1$ is a constant that models how strongly non-routed users favor high-capacity roads over low-capacity roads, and p^{hi} (resp. p^{lo}) are the segments of roads in path p that are high (resp. low) capacity. Note that the multiplicative cognitive cost conserves the proportions between low-capacity links' travel times while increasing their costs.

Rationale behind preference for high-capacity links

Under low traffic demand, high-capacity roads generally enable to travel quickly between origins and destinations far apart. To validate this on the Los Angeles network, we collected the OD trip data from the American Community Survey (ACS), composed of $K = 96,077$ OD pairs and a demand vector $\mathbf{d} \in \mathbb{R}^K$. In the Los Angeles network in free flow, for each population $k \in [K]$, we extracted a path p_k^{nr} with lowest non-routed cost $\min_{p \in \mathcal{P}_k} \ell_p^{\mathrm{nr}}(0)$ for each OD pair $k \in [K]$ using python-igraph package, and found that that associated free-flow travel time $\sum_{e \in p_k^{\mathrm{nr}}} t_e(0)$ is on average only 10% longer than the shortest route, as illustrated by Figure 4.2.a). In addition, travel times of non-routed users in the free-flow regime are not sensitive to the cognitive cost when it is above 1000. Hence, for the remainder of this work, we fix the non-routed costs c_e^{nr} with a cognitive cost $C = 3000$ and focus on the sensitivity of road usage to variations in the traffic demand and in the percentage of *routed users*. Moreover, Figure 4.2.b) shows a small shift of the travel time distribution in positive direction as the cognitive cost increases from 1 to 1000. Hence, without traffic, the Los Angeles high-capacity network provides a reliable and nearly optimal route for traversing cities with no information on local roads, thus justifying the rationale behind *non-routed users*' preference.

Rationale behind routing on low-capacity links

With increasing demand, high-capacity roads such as highways become congested since *non-routed users* choose them over low-capacity routes. We model flow of vehicles on roads using the traffic assignment framework [130] in which each *non-routed user*, represented as an infinitesimal amount of flow, selfishly chooses the path with the lowest cost $\ell_p^{\mathrm{nr}}(\boldsymbol{\mu})$. This concept is known in the transportation literature as Wardrop's first principle [165].

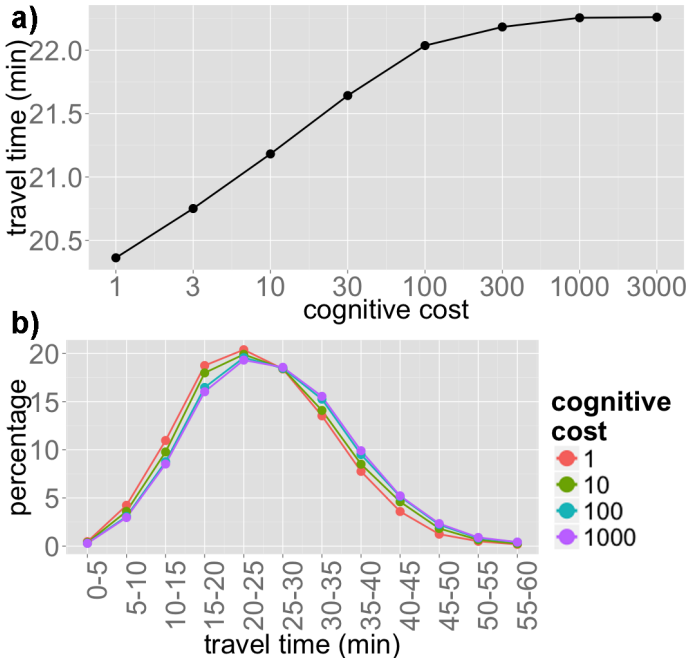

Figure 4.2: Travel times in Los Angeles when all edges are in free flow for *non-routed users* with perceived costs given by (4.4), as a function of the cognitive cost C. Figure a) shows the average travel time, Figure b) shows the distribution of travel times.

The resulting flow is an equilibrium flow $\boldsymbol{\mu} \in \mathbb{R}^{\mathcal{P}}$ for which the associated equilibrium edge flow $\mathbf{x} = (x_e)_{e\in\mathcal{E}} \in \mathbb{R}^{\mathcal{E}}$ is unique when the travel time functions t_e are continuously differentiable, positive and strictly increasing [13]. Under these assumptions on the travel functions, Beckmann et al. [13] show that the equilibrium edge flow of the routing game can be expressed as the optimal solution of the following convex program

$$\min_{\mathbf{x}} \ \phi(\mathbf{x}) = \sum_{e\in\mathcal{E}} \int_0^{x_e} c_e^{\mathrm{nr}}(u)du \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{D} \tag{4.6}$$

where ϕ is a potential function, c_e^{nr} is given by (4.4), and $\mathcal{D}$ is given by (4.3). We obtain different traffic demands by multiplying the demand vector $\mathbf{d} \in \mathbb{R}^K$ obtained from the ACS data by a scalar $\alpha \in [0.1, 1]$. We then solve (4.6) with a cognitive cost $C = 3000$ and different traffic demands to obtain various non-routed equilibrium flows $\mathbf{x}^{\mathrm{nr}}$. The network with 100% of *non-routed users* settles in a suboptimal state with imbalances in the flow allocation where high-capacity links are over-utilized and low-capacity links are under-utilized. We compare it to the *routed equilibrium* arc flow $\mathbf{x}^{\mathrm{r}}$, where every user follows the shortest path, with costs given by

$$\ell_p^{\mathrm{r}}(\boldsymbol{\mu}) = \sum_{e\in p} t_e(x_e), \quad \forall\, p \in \mathcal{P} \tag{4.7}$$

The equilibrium is obtained by solving (4.6) with arc costs $c_e^{\mathrm{nr}}(\cdot)$ equal to the travel time functions $t_e(\cdot)$. The ratio of the respective total travel times $\sum_{e\in\mathcal{E}} x_e^{\mathrm{nr}} c_e^{\mathrm{nr}}(x_e^{\mathrm{nr}})$ and $\sum_{e\in\mathcal{E}} x_e^{\mathrm{r}} t_e(x_e^{\mathrm{r}})$ are shown in turquoise in Figure 4.3. Figure 4.4.b) also shows that 20% of the users experience a 10-20% delay and 12% experience a 20-30% delay compared to the routed equilibrium.

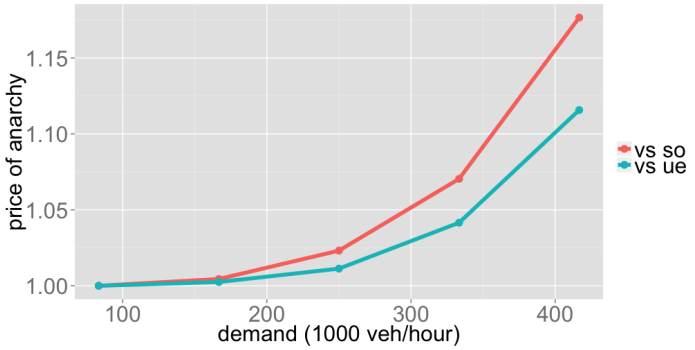

Figure 4.3: Ratio of the average travel time when the perceived non-routed costs are given by (4.4) with $C = 3000$ over the user equilibrium (blue) and the social optimum (red), as a function of the demand in the network.

We also compare the non-routed equilibrium to the *social optimum* where the total cost incurred by all users in the network is minimized

$$\min \sum_{e\in\mathcal{E}} x_e\, t_e(x_e) \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{D} \tag{4.8}$$

Figures 4.3 and 4.4 show that the preference for high-capacity links steers the equilibrium state further from the social optimum where 25% of users experience 10-20% delay and 18% of users experience a 20-30% delay. Hence, rational users are pushed to choose low-capacity roads to avoid segments of high-capacity roads that are not along the shortest route due to congestion under heavy traffic demand.

4.4 Multiclass traffic assignment problem

The sharp increase of app-based routing spurred by the increasing penetration of navigation devices progressively increases the number of *routed users* on the road. It is likely that with the full advent of automated driving, this trend will accelerate in the future. This emerging behavior is in sharp contrast with non-routed users who favor high-capacity roads regardless to the level of congestion. To quantify the impact of *routed users* on traffic conditions, we introduce our heterogeneous traffic assignment problem with both *routed users* and *non-routed users*.

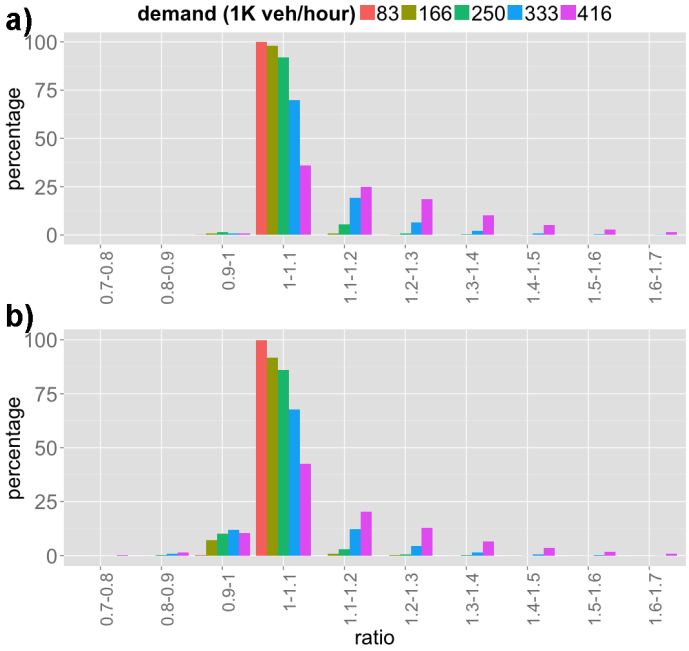

Figure 4.4: The distribution of the ratio of the travel times over the social optimum per OD pair (a), and the user equilibrium (b), when all users are non-routed, when the perceived costs are given by (4.4) with $C = 3000$.

Multiclass traffic assignment problem

We consider a demand vector $\mathbf{d}_k^{\mathrm{r}} \in \mathbb{R}_+^K$ of *routed users* and a demand vector $\mathbf{d}_k^{\mathrm{nr}} \in \mathbb{R}_+^K$ of *non-routed users* between each OD pair (or population) $k \in [K]$. The state of the network is described by the routed users' path flow vector $\boldsymbol{\mu}^{\mathrm{r}} = (\mu_p^{\mathrm{r}})_{p\in\mathcal{P}}$ and the non-routed users' path flow vector $\boldsymbol{\mu}^{\mathrm{nr}} = (\mu_p^{\mathrm{nr}})_{p\in\mathcal{P}}$. They are feasible if they are in Δ^{r}, Δ^{nr} given by

$$\Delta^{\mathrm{r}} := \{\boldsymbol{\mu}^{\mathrm{r}} \in \mathbb{R}^P \,:\, \boldsymbol{\mu}^{\mathrm{r}} \succeq 0,\, \mathbf{A}\boldsymbol{\mu}^{\mathrm{r}} = \mathbf{d}^{\mathrm{r}}\} \tag{4.9}$$

$$\Delta^{\mathrm{nr}} := \{\boldsymbol{\mu}^{\mathrm{nr}} \in \mathbb{R}^P \,:\, \boldsymbol{\mu}^{\mathrm{nr}} \succeq 0,\, \mathbf{A}\boldsymbol{\mu}^{\mathrm{nr}} = \mathbf{d}^{\mathrm{nr}}\} \tag{4.10}$$

where $\mathbf{A}$ is the OD-path incidence matrix. With $\mathbf{M}$ the arc-path incidence matrix, we denote $\mathbf{x}^{\mathrm{r}} = (\mathbf{x}_e^{\mathrm{r}})_{e\in\mathcal{E}} = \Delta\boldsymbol{\mu}^{\mathrm{r}}$ and $\mathbf{x}^{\mathrm{nr}} = (x_e^{\mathrm{nr}})_{e\in\mathcal{E}} = \Delta\boldsymbol{\mu}^{\mathrm{nr}}$ the routed and non-routed arc flow vectors respectively. Hence $\mathbf{x}^{\mathrm{r}}$, $\mathbf{x}^{\mathrm{nr}}$ are feasible if they belong to the following sets respectively

$$\mathcal{D}^{\mathrm{r}} := \{\mathbf{x}^{\mathrm{r}} \in \mathbb{R}^{\mathcal{E}} \,:\, \exists\, \boldsymbol{\mu}^{\mathrm{r}} \in \Delta^{\mathrm{r}},\, \mathbf{x}^{\mathrm{r}} = \mathbf{M}\boldsymbol{\mu}^{\mathrm{r}}\} \tag{4.11}$$

$$\mathcal{D}^{\mathrm{nr}} := \{\mathbf{x}^{\mathrm{nr}} \in \mathbb{R}^{\mathcal{E}} \,:\, \exists\, \boldsymbol{\mu}^{\mathrm{nr}} \in \Delta^{\mathrm{nr}},\, \mathbf{x}^{\mathrm{nr}} = \mathbf{M}\boldsymbol{\mu}^{\mathrm{nr}}\} \tag{4.12}$$

The total path flow is $\boldsymbol{\mu} = \boldsymbol{\mu}^{\mathrm{r}} + \boldsymbol{\mu}^{\mathrm{nr}} = (\mu_p^{\mathrm{r}} + \mu_p^{\mathrm{nr}})_{p\in\mathcal{P}}$ and the total arc flow is $\mathbf{x} = \mathbf{x}^{\mathrm{r}} + \mathbf{x}^{\mathrm{nr}} = (x_e^{\mathrm{r}} + x_e^{\mathrm{nr}})_{e\in\mathcal{E}}$. As both routed and non-routed users make selfish choices by minimizing their associated costs, the resulting flow essentially describes the Nash equilibrium on road

networks. Mathematically, the equilibrium flow are feasible flows $\boldsymbol{\mu}^{\mathrm{r}} \in \Delta^{\mathrm{r}}$, $\boldsymbol{\mu}^{\mathrm{nr}} \in \Delta^{\mathrm{nr}}$ such that $\forall\, k \in [K]$

$$\forall\, p \in \mathcal{P}_w,\, \mu_p^{\mathrm{r}} > 0 \implies \ell_p^{\mathrm{r}}(\boldsymbol{\mu}) = \min_{q \in \mathcal{P}_w} \ell_q^{\mathrm{r}}(\boldsymbol{\mu}) \tag{4.13}$$

$$\forall\, p \in \mathcal{P}_w,\, \mu_p^{\mathrm{nr}} > 0 \implies \ell_p^{\mathrm{nr}}(\boldsymbol{\mu}) = \min_{q \in \mathcal{P}_w} \ell_q^{\mathrm{nr}}(\boldsymbol{\mu}) \tag{4.14}$$

where the routed and non-routed path costs ℓ_p^{r} and ℓ_p^{nr} are given by (4.7) and (4.5) respectively. Hence, only the least-cost paths are used between each origin and destination with respect to the associated type of users. The equilibrium $\boldsymbol{\mu}$ described in (4.13) and (4.14) can be expressed as a feasible solution $(\boldsymbol{\mu}^{\mathrm{r}}, \boldsymbol{\mu}^{\mathrm{nr}}) \in \Delta^{\mathrm{r}} \times \Delta^{\mathrm{nr}}$ of the following variational inequality problem

$$\ell^{\mathrm{r}}(\boldsymbol{\mu})^T \boldsymbol{\nu}^{\mathrm{r}} + \ell^{\mathrm{nr}}(\boldsymbol{\mu})^T \boldsymbol{\nu}^{\mathrm{nr}} \geq \ell^{\mathrm{r}}(\boldsymbol{\mu})^T \boldsymbol{\mu}^{\mathrm{r}} + \ell^{\mathrm{nr}}(\boldsymbol{\mu})^T \boldsymbol{\mu}^{\mathrm{nr}}, \quad \forall\, (\boldsymbol{\nu}^{\mathrm{r}}, \boldsymbol{\nu}^{\mathrm{nr}}) \in \Delta^{\mathrm{r}} \times \Delta^{\mathrm{nr}} \tag{4.15}$$

Contrary to the homogeneous routing game, the general heterogeneous game cannot be formulated as a potential game of the form (4.6), see [140], [63]. However, by using the theory of variational inequality [61], it is possible to solve for the equilibrium described in (4.15) with the Frank-Wolfe algorithm [175].

Positive impact

We apply the multi-class traffic assignment framework to the network of Los Angeles with a variable percentage α of *routed users*, and a cognitive cost $C = 3000$ for *non-routed users*, which means that their perceived cost on low-capacity links is 3000 times the real travel-time. We assume a uniform ratio of *routed users* for each OD pair $k \in [K]$, hence $d_k^{\mathrm{r}} = \alpha\, d_k$ and $d_k^{\mathrm{nr}} = (1-\alpha)\, d_k$, where $\alpha \in [0, 1]$ and the total traffic demand $\mathbf{d}$ is given by the ACS data. As the fraction α of *routed users* increases, Figure 4.5 shows a shift of the travel time distribution to the left as a result of users allocating themselves optimally (but selfishly) between the low-capacity and high-capacity networks. At an aggregate level, GPS routing can alleviate the road network with a possible decrease in Vehicle-miles Traveled (VMT) from 7.94 million miles per hour to 7.15 million, hence a potential decrease of .79 million miles per hour, see Figure 4.6.b), thus corroborating the belief that GPS routing is able to alleviate gridlock in congested areas.

Negative externalities

Even though the increase in usage of app-based routing enables better navigation and time savings, they allegedly transfer large amounts of traffic in cities bordering highways, since navigation apps users have been reported to leave highways to avoid congestion [66]. For instance, in the Los Angeles network used for the present study and shown in Figure 4.1, we find that app-based routing can potentially increase the VMT on local roads by .34 million miles per hour, which represents a threefold increase in traffic on low-capacity links, while

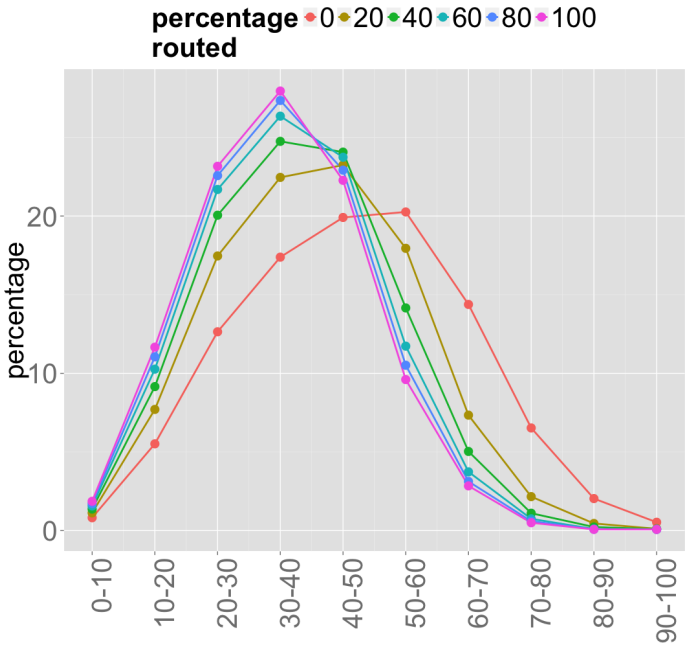

Figure 4.5: Distribution of travel times as a function of the percentage of routed users, with cognitive cost $C = 3000$ for *non-routed users*.

there is only a 10% decrease in VMT on high-capacity roads, see Figure 4.6. Moreover, Figure 4.7 shows that an increase in *routed users*' ratio α is accompanied with a sharp increase in the percentage of users spending between 10 and 20 min on low-capacity links (we reiterate that we apply the framework to the Los Angeles network presented in Figure 4.1). Figure 4.8 shows that, despite a general decrease in VMT due to more efficient routing, the relative increase on low-capacity roads is very important for each 10% increase in routed users, due to the small traffic flow on the low-capacity network. This causes residential streets to be congested, encouraging cities to spend millions in infrastructure to steer the traffic away.

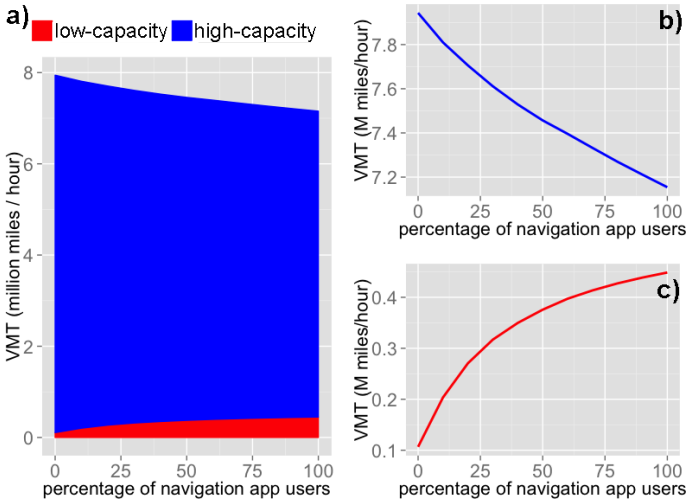

Figure 4.6: General VMT versus VMT on local roads as a function of the percentage of routed users.

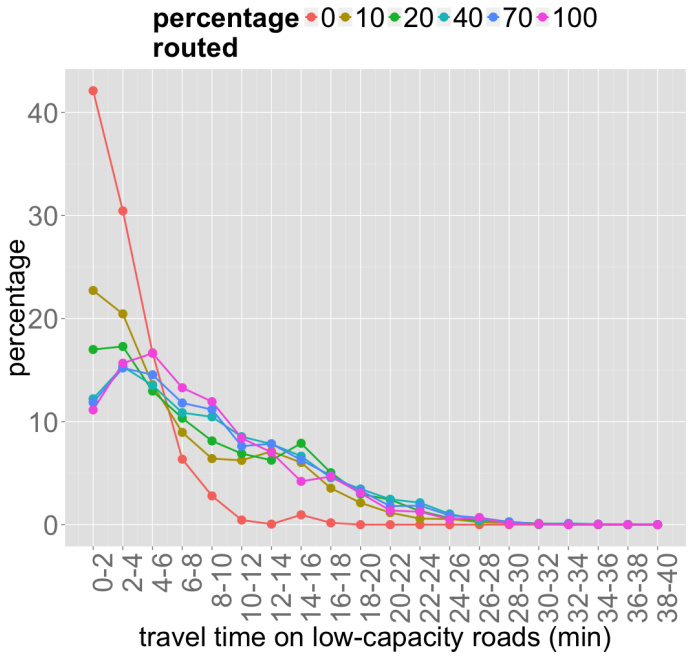

Figure 4.7: Distribution of travel times on local roads as a function of the percentage of routed users.

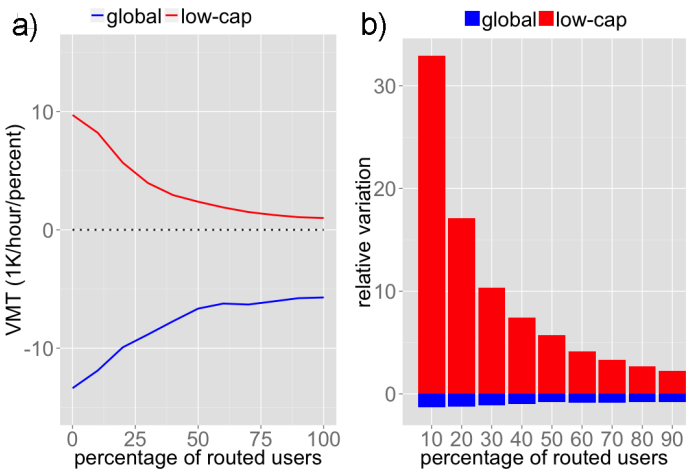

Figure 4.8: a) Variation in VMT for 1% increase in routed users. b) Relative variation in VMT for 10% increase in routed users.

Part II

Statistics of learning the edge cost functions in selfish routing games

In this part, we study the learnability of the edge cost functions in routing games from observations of the equilibrium flows, where the learnability is measured as the minimum number of samples needed to maintain a high consistency of the empirical risk, which is a statistical estimator for the quality of the learned model. To provide an upper bound on the minimum sample size, we motivate the analysis of the uniform laws of large numbers on a class of loss functions indexed over the space of parameters we want to estimate. On one hand, leveraging results on the complexity of function classes and in approximation theory, we investigate how the behavior of the empirical risk relates to a notion of complexity of the parameter space. On the other hand, using sensitivity analysis in optimization, we study how variations in the parameter space translate into variations in the class of loss functions. This allows us to show, independently of the sample distribution and of the edge costs parametrization, that the sample size required to maintain a high consistency grows quadratically with the number of edges in the network (or linearly if all the edge cost functions have the same shape), and grows linearly with the Lipschitz constant of the edge cost functions, and quadratically with the inverse of the strong monotonicity constant of the edge cost functions.

Chapter 5

Learnability of edge cost functions

We study the learnability of the edge cost functions in routing games from observations of the equilibrium flows, where the learnability is measured as the minimum number of samples needed to maintain a high consistency of the empirical risk, which is a statistical estimator for the quality of the learned model. To provide an upper bound on the minimum sample size, we motivate the analysis of the uniform laws of large numbers on a class of loss functions indexed over the space of parameters we want to estimate. On one hand, leveraging results on the complexity of function classes and in approximation theory, we investigate how the behavior of the empirical risk relates to a notion of complexity of the parameter space. On the other hand, using sensitivity analysis in optimization, we study how variations in the parameter space translate into variations in the class of loss functions. This allows us to show, independently of the sample distribution and of the edge costs parametrization, that the sample size required to maintain a high consistency grows quadratically with the number of edges in the network (or linearly if all the edge cost functions have the same shape), and grows linearly with the Lipschitz constant of the edge cost functions, and quadratically with the inverse of the strong monotonicity constant of the edge cost functions.

5.1 Introduction

Routing games have been extensively studied in transportation settings, see [131] and the references therein. Such models enable to study drivers' routing decisions in a network modeled as a directed graph, in which traveling each edge incurs a cost. It is well-known that if agents selfishly route themselves, the aggregate cost in the network is worse than the system's optimum [137]. In many instances however, the design of strategies, such as taxation schemes [65], [91], to incentivize drivers' decisions that are system optimal, relies upon knowing the shape of the edge cost functions that are being modified [30]. Estimating the edge cost functions is a challenging task since they may represent some combination of the actual travel time, the tolls, and disutility from environmental factors, which are not directly observable. In practice, it is often possible to observe, through the sensing infrastructure,

the equilibrium flows induced by the selfish routing of agents, and learn the underlying cost functions. This spurs the recent study of a class of learning problems known as *inverse optimization* [85, 94, 19, 150].

Seeking to learn the edge cost functions, empirical risk minimization is a standard decision-theoretic approach of estimating them by choosing the ones giving the lowest expected loss under the empirical measure. Thus, a critical question on the learning process is whether or not, and at which rate, the empirical risk approaches the population risk, which is the expected value of the out-of-sample loss. The population risk gives us a ultimate measure of the prediction capability of a trained model, and its estimation thus lies at the heart of techniques concerned with model selection, see *e.g.* [78, Chap. 7]. Thus, analyzing the consistency of the empirical risk is extremely important in practice since it assesses the viability of the empirical risk minimization method applied to our instance.

Outline. Section 5.2 describes the routing game in detail, and Section 5.3 sets up the framework for learning the edge cost functions. In Section 5.4, we construct a class of loss functions which will be the focus of our analysis, and show in Section 5.5 that studying whether or not the uniform law of large numbers holds for this class, and related convergence rates, helps us to assess the performance of our learning process. In Sections 6.4 and 6.5, we relate the consistency properties of the empirical risk to the complexity of the space over which we fit our parameters, and show in Section 6.3 how this can be applied to our inverse optimization problem using results on the sensitivity of optimization programs. Section 5.9 applies our general results to the routing game.

Notations. In our paper, we consider continuous mappings $F : \mathcal{X} \to \mathcal{Y}$ over a compact domain $\mathcal{X}$, and a norm over them defined by $|||F||| := \sup_{\mathbf{x}\in\mathcal{X}} \|F(\mathbf{x})\|_{\mathcal{Y}}$, where $\|\cdot\|_{\mathcal{Y}}$ is a norm on $\mathcal{Y}$. Since the domain $\mathcal{X}$ is compact, $|||\cdot|||$ always exists. And we will say that a function class $\mathcal{F} = \{F_{\boldsymbol{\theta}} \,|\, \boldsymbol{\theta} \in \Theta\}$ is L-smoothly parametrized if $|||F_{\boldsymbol{\theta}} - F_{\boldsymbol{\theta}'}||| \leq L\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\Theta$ for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$, where $\|\cdot\|_\Theta$ is a norm on Θ. Hence, for all $\mathbf{x} \in \mathcal{X}$, $\|F_{\boldsymbol{\theta}}(\mathbf{x}) - F_{\boldsymbol{\theta}'}(\mathbf{x})\|_{\mathcal{Y}} \leq \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\Theta$.

5.2 Selfish routing

Setting. We consider the routing game based on Wardrop's principles [166]. It is a non-cooperative game on a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in which the set of players is partitioned in *populations* $\{\mathcal{X}_k\}_{k\in[K]}$, where $[K]$ denotes the set $\{1, \cdots, K\}$. For each $k \in [K]$, players in $\mathcal{X}_k$ have available a set of simple paths $\mathcal{P}_k$ from a common source $s_k \in \mathcal{V}$ to a common sink $t_k \in \mathcal{V}$. For each population $\mathcal{X}_k$, we define $d_k \in \mathbb{R}_+$ its total flow, and $\boldsymbol{\mu}^k := (\mu_p^k)_{p\in\mathcal{P}_k} \in \mathbb{R}_+^{\mathcal{P}_k}$ its *path assignment*, which satisfies $\sum_{p\in\mathcal{P}_k} \mu_p^k = d_k$. We denote $\mathcal{P}$ the disjoint union $\mathcal{P} = \sqcup_{k=1}^K \mathcal{P}_k$, thus $\mathbb{R}_+^{\mathcal{P}} = \Pi_{k=1}^K \mathbb{R}_+^{\mathcal{P}_k}$. Under demand $\mathbf{d} = (d_k)_{k\in[K]} \in \mathbb{R}_+^K$, the path assignment can be summarized by $\boldsymbol{\mu} := (\boldsymbol{\mu}^1, \cdots, \boldsymbol{\mu}^K)$ lying within the feasible set

$$\Delta(\mathbf{d}) := \left\{ \boldsymbol{\mu} \in \mathbb{R}_+^{\mathcal{P}} : \textstyle\sum_{p\in\mathcal{P}_k} \mu_p^k = d_k, \, \forall\, k \in [K] \right\} \tag{5.1}$$

The path assignment determines the *edge flow* defined as $x_e = \sum_{k=1}^{K} \sum_{p \in \mathcal{P}_k : e \in p} \mu_p^k$, which can be written compactly as $x_e = (\mathbf{M}\boldsymbol{\mu})_e$ where $\mathbf{M} \in \mathbb{R}^{\mathcal{E} \times \mathcal{P}}$ is an incidence matrix with entries defined as $M_{e,p} = \mathbf{1}_{e \in p}$. For each edge $e \in \mathcal{E}$, the edge flow incurs a cost $c_e(x_e)$ where $c_e(\cdot) : \mathbb{R}_+ \to \mathbb{R}_{>0}$ is *positive, continuous, strictly increasing* cost function. The cost of choosing a path p is the sum of edge costs along the path $\sum_{e \in p} c_e(x_e)$. Since the path cost is entirely defined by $\boldsymbol{\mu}$, we can express the cost of path $p \in \mathcal{P}_k$ as the mapping $\ell_p^k : \boldsymbol{\mu} \in \Delta(\mathbf{d}) \mapsto \sum_{e \in p} c_e((\mathbf{M}\boldsymbol{\mu})_e)$. We write $\ell(\boldsymbol{\mu})$ to denote the vector of path cost functions $(\ell_p(\boldsymbol{\mu}))_{p \in \mathcal{P}}$. If we define the mapping F as the vector of congestion functions

$$F : \mathbf{x} \in \mathbb{R}_+^{\mathcal{E}} \mapsto F(\mathbf{x}) = (c_e(x_e))_{e \in \mathcal{E}} \tag{5.2}$$

the path costs can be written compactly as the vector of functions $\ell : \boldsymbol{\mu} \in \Delta(\mathbf{d}) \mapsto \ell(\boldsymbol{\mu}) = \mathbf{M}^T F(\mathbf{M}\boldsymbol{\mu})$.

Nash equilibrium: We say that $\boldsymbol{\mu}^\star \in \Delta(\mathbf{d})$ is a Nash equilibrium if for every population k, $\mu_p^k > 0$ for some path $p \in \mathcal{P}_k$ implies that $\ell_p^k(\boldsymbol{\mu}^\star) = \min_{q \in \mathcal{P}_k} \ell_q^k(\boldsymbol{\mu}^\star)$, *i.e.* a path is only used if it is of least cost in $\mathcal{P}_k$. This is equivalent to the condition, see *e.g.* §3.2.1. in [131]

$$\langle \ell(\boldsymbol{\mu}^\star), \boldsymbol{\mu} - \boldsymbol{\mu}^\star \rangle \geq 0, \quad \forall\, \boldsymbol{\mu} \in \Delta(\mathbf{d}) \tag{5.3}$$

Since $\ell(\boldsymbol{\mu}) = \mathbf{M}^T F(\mathbf{M}\boldsymbol{\mu})$, substituting in (5.3) gives the condition $\langle F(\mathbf{M}\boldsymbol{\mu}^\star), \mathbf{M}(\boldsymbol{\mu} - \boldsymbol{\mu}^\star) \rangle \geq 0$, re-written as

$$\langle F(\mathbf{x}^\star), \mathbf{x} - \mathbf{x}^\star \rangle \geq 0, \quad \forall\, \mathbf{x} \in \mathcal{K}(\mathbf{d}) \tag{5.4}$$

where $\mathbf{x}^\star = \mathbf{M}\boldsymbol{\mu}^\star$ is the Nash equilibrium for the edge flows, and the set of feasible edge flows $\mathcal{K}(\mathbf{d})$ is given by

$$\mathcal{K}(\mathbf{d}) = \mathbf{M}\Delta(\mathbf{d}) = \{\mathbf{M}\boldsymbol{\mu} : \boldsymbol{\mu} \in \Delta(\mathbf{d})\} \subset \mathbb{R}_+^{\mathcal{E}} \tag{5.5}$$

For more details on the reformulation, we refer to *e.g.* §3.2.1. [131]. Note that the equilibrium flow $\mathbf{x}^\star$ in (5.4) is a solution to a parametric *variational inequality problem (VIP)*, denoted $\mathrm{VI}(\mathcal{K}(\mathbf{d}), F)$, with mapping F and parametric polyhedral domain $\mathcal{K}(\mathbf{d})$, with parameter the population demand $\mathbf{d} \in \mathbb{R}_+^K$, see *e.g.* [49], [61]. To avoid path enumeration, note that we can equivalently use a vertex-representation equivalent to (5.4) since the cost functions $(c_e(\cdot))_{e \in \mathcal{E}}$ are positive by assumption, see *e.g.* [131, §2.2.2].

Existence and uniqueness. For every $\mathbf{d} \in \mathbb{R}_+^K$, the parametric feasible set $\mathcal{K}(\mathbf{d})$ is non-empty and it is compact convex since it is the image of $\mathbf{M}$ restricted to the compact convex set $\Delta(\mathbf{d})$. This gives us the existence of a solution to (5.4), see *e.g.* [95, Chap. 1]. Uniqueness of the solution is given by strict monotonicity of the mapping F since the cost functions $(c_e(\cdot))_{e \in \mathcal{E}}$ are strictly increasing by assumption. Hence, for every $\mathbf{d} \in \mathbb{R}_+^K$, the parametric $\mathrm{VI}(\mathcal{K}(\mathbf{d}), F)$ admits a unique solution $\mathbf{x}^\star(\mathbf{d}) \in \mathcal{K}(\mathbf{d})$. The equilibrium flow thus defines an *implicit function* $\mathbf{d} \mapsto x^\star(\mathbf{d})$.

5.3 Statistical learning framework

Training data. We now formalize the supervised learning problem. Suppose we are given the parametric domain $\mathcal{K}(\mathbf{d})$ along with a collection $\mathbf{d}_1^N := \{\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_N\}$ of N i.i.d. samples of the demand vector $\mathbf{d}$ drawn from a set $\mathcal{D} \subset \mathbb{R}_+^K$ according to a probability distribution $\mathbb{P}$. The samples $\mathbf{d}_1^N$ are the *predictors* (or inputs) of our learning problem, and they lie within the *measure space* $(\mathcal{D}, \Sigma, \mathbb{P})$, where Σ is a σ-algebra of measurable sets. We suppose the *responses* (or outputs) are given by the implicit function $\mathbf{d} \mapsto \mathbf{x}^\star(\mathbf{d})$ solution to $\mathrm{VI}(\mathcal{K}(\mathbf{d}), F)$ given in (5.4)

$$\mathbf{y}_i = h(\mathbf{x}^\star(\mathbf{d}_i)) \quad i \in \{1, \cdots, N\} \tag{5.6}$$

where $h(\cdot)$ is a L_h-Lipschitz *observation mapping* from the state space $\mathbb{R}_+^{\mathcal{E}}$ to the observed space $\mathbb{R}^m$. The mapping h models the sensing infrastructure. For example, m can index the subset of edges from $\mathcal{E}$ on which flows are measured, then $h(\cdot)$ is a projection operator onto the subspace $\mathbb{R}^m$ of $\mathbb{R}^{\mathcal{E}}$. The collection of predictor-response pairs $\{(\mathbf{d}_1, \mathbf{y}_1), \cdots, (\mathbf{d}_N, \mathbf{y}_N)\}$ is our *training data*, from which our aim is to learn F in the parametric $\mathrm{VI}(\mathcal{K}(\mathbf{d}), F)$.

Maximum demand. We suppose that the random demand vector $\mathbf{d}$ has total mass $\|\mathbf{d}\|_1$ less than $\bar{d}$ almost surely (it lies within a simplex of mass $\bar{d}$), the random polyhedron $\mathcal{K}(\mathbf{d})$ is thus contained in the hypercube $[0, \bar{d}]^{\mathcal{E}}$. Concretely, this means that the total flow on each edge $e \in \mathcal{E}$ is less than the *maximum total demand* $\bar{d}$ almost surely.

Implicit function class. We want to find an estimate $\hat{F}$ of F, where $\hat{F}$ is chosen over a function class. Concretely, we first define a *base class* of univariate functions:

$$\mathcal{M} = \big\{ f : [0,1] \to \mathbb{R}_+,\ L\text{-Lipschitz}, \tag{5.7}$$

$$c\text{-strong-monotone},\ f(0) = 0 \big\} \tag{5.8}$$

where $L, c \in \mathbb{R}_{>0}$. We are also given a *positive zero-flow cost* c_e^0 and an *positive edge capacity* m_e for each $e \in \mathcal{E}$. Without loss of generality, we suppose that $\{m_e\}_{e \in \mathcal{E}}$ were uniformly rescaled such that $\bar{d} = \min_{e \in \mathcal{E}} m_e$. Hence, the normalized edge flows satisfy $x_e/m_e \leq \bar{d}/m_e \leq 1$ almost surely. We now choose $\hat{F}$ within one of these *cost classes*

$$\mathcal{F}_1 = \{\mathbf{x} \in [0, \bar{d}]^{\mathcal{E}} \mapsto (c_e^0 + f_e(\tfrac{x_e}{m_e}))_e \,|\, \{f_e\}_e \in \mathcal{M}^{\mathcal{E}}\} \tag{5.9}$$

$$\mathcal{F}_2 = \{\mathbf{x} \in [0, \bar{d}]^{\mathcal{E}} \mapsto (c_e^0 + f(\tfrac{x_e}{m_e}))_e \,|\, f \in \mathcal{M}\} \tag{5.10}$$

The class $\mathcal{F}_2$ considers candidate cost functions $\{c_e(\cdot)\}_{e \in \mathcal{E}}$ which are uniformly equal to a single function f of the normalized flows $\big(\frac{x_e}{m_e}\big)_{e \in \mathcal{E}}$ (modulo an additive term). This is a standard assumption in traffic modeling, see *e.g.* [31], [29], and in inverse modeling [19], [150]. Since every function in $\mathcal{M}$ is L-Lipschitz and c-strong-monotone, then every mapping in $\mathcal{F}_1$ or $\mathcal{F}_2$ is $\big(\frac{L}{\min_{e \in \mathcal{E}} m_e}\big)$-Lipschitz and $\big(\frac{c}{\max_{e \in \mathcal{E}} m_e}\big)$-strong-monotone with respect to the Euclidean norm $\|\cdot\|_2$ and the Euclidean inner product $\langle \cdot, \cdot \rangle$.

5.4 Problem statement

We are interested in learning methods based on *empirical risk minimization*. To formalize this approach, we consider an indexed-family $\mathcal{F} := \{F_{\boldsymbol{\theta}} \,|\, \boldsymbol{\theta} \in \Theta\}$ of strictly monotone mappings, and suppose that there exists some fixed but unknown $\boldsymbol{\theta}^\star \in \Theta$ such that $F = F_{\boldsymbol{\theta}^\star}$, *i.e.* $\mathrm{VI}(\mathcal{K}(\mathbf{d}), F)$ is the same as $\mathrm{VI}(\mathcal{K}(\mathbf{d}), F_{\boldsymbol{\theta}^\star})$. For example, $\mathcal{F}$ can be either $\mathcal{F}_1$ and $\mathcal{F}_2$ in (5.9) and (5.10), which are indexed over the base classes $\mathcal{M}^{\mathcal{E}}$ and $\mathcal{M}$ respectively. With this setting, we now want to fit $F_{\boldsymbol{\theta}}$ to $F_{\boldsymbol{\theta}^\star}$, where $F_{\boldsymbol{\theta}}$ is chosen within $\mathcal{F} := \{F_{\boldsymbol{\theta}} \,|\, \boldsymbol{\theta} \in \Theta\}$, *i.e.* we fit the parameter $\boldsymbol{\theta} \in \Theta$ to $\boldsymbol{\theta}^\star$. This gives us an indexed-family $\{\mathbf{x}^\star_{\boldsymbol{\theta}}(\cdot) \,|\, \boldsymbol{\theta} \in \Theta\}$ of implicit functions such that for each $(\mathbf{d}, \boldsymbol{\theta}) \in \mathcal{D} \times \Theta$, the vector $\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d})$ is the unique solution to $\mathrm{VI}(\mathcal{K}(\mathbf{d}), F_{\boldsymbol{\theta}})$. The response is then given by

$$\mathbf{y}_i = h(\mathbf{x}^\star_{\boldsymbol{\theta}^\star}(\mathbf{d}_i)) \quad i \in \{1, \cdots, N\} \tag{5.11}$$

Empirical risk minimization. Given some norm on the observed space $\mathbb{R}^m$, we pose the loss function as

$$\ell_{\boldsymbol{\theta}}(\mathbf{d}) := \|h(\mathbf{x}^\star_{\boldsymbol{\theta}^\star}(\mathbf{d})) - h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d}))\| \tag{5.12}$$

Since the response is $\mathbf{y} = h(\mathbf{x}^\star_{\boldsymbol{\theta}^\star}(\mathbf{d}))$ by assumption (5.11), the loss function is the distance $\|\mathbf{y} - h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d}))\|$ between the real and predicted responses. Given samples $(\mathbf{d}_1^N, \mathbf{y}_1^N) = \{(\mathbf{d}_1, \mathbf{y}_1), \cdots, (\mathbf{d}_N, \mathbf{y}_N)\}$ of predictor-response pairs with relationship given in (5.11), a standard decision-theoretic approach of estimating the parameter $\boldsymbol{\theta}^\star$ is to minimize the expected loss under the empirical measure

$$R_N(\boldsymbol{\theta}) := \tfrac{1}{N} \textstyle\sum_{i=1}^N \ell_{\boldsymbol{\theta}}(\mathbf{d}_i) \tag{5.13}$$

This quantity is known as the *empirical risk* and can be re-written in terms of $(\mathbf{d}_1^N, \mathbf{y}_1^N)$ as $\frac{1}{N}\sum_{i=1}^N \|\mathbf{y}_i - h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d}_i))\|$. Empirical risk minimization methods thus seek to compute an element in $\arg\min_{\boldsymbol{\theta}\in\Theta} R_N(\boldsymbol{\theta})$, see [158]. Note that $R_N(\boldsymbol{\theta})$ should be contrasted with the *population risk*

$$R(\boldsymbol{\theta}) := \mathbb{E}_{\mathbf{d}}\left[\ell_{\boldsymbol{\theta}}(\mathbf{d})\right] \tag{5.14}$$

Goal. Let $\hat{\boldsymbol{\theta}}$ be an estimate of $\boldsymbol{\theta}^\star$ computed from samples $(\mathbf{d}_1^N, \mathbf{y}_1^N)$. We do not assume that $\hat{\boldsymbol{\theta}}$ is a minimizer of the empirical risk (5.13), and we refer to, *e.g.*, [94], [19], [150] for practical methods for minimizing (5.13). Instead, the main focus of the present chapter is to control the quantity $|R_N(\hat{\boldsymbol{\theta}}) - R(\hat{\boldsymbol{\theta}})|$. In general, $\hat{\boldsymbol{\theta}}$ depends on the samples, hence it is random, and controlling $|R_N(\hat{\boldsymbol{\theta}}) - R(\hat{\boldsymbol{\theta}})|$ requires a strong result, such as a uniform bound over Θ, namely $\sup_{\boldsymbol{\theta}\in\Theta} |R_N(\boldsymbol{\theta}) - R(\boldsymbol{\theta})|$. To achieve this, we turn to the *uniform laws of large numbers*. In its general form, this class of results considers a collection $X_1^N := \{X_1, \cdots, X_N\}$ of i.i.d. samples from some distribution $\mathbb{P}$ over $\mathcal{X}$ and a class $\mathcal{F}$ of real-valued integrable functions with domain $\mathcal{X}$, and studies the convergence properties of the *random variable*

$$\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}} := \sup_{f \in \mathcal{F}} \left| \frac{1}{N} \sum_{i=1}^{N} f(X_i) - \mathbb{E}[f(X)] \right| \tag{5.15}$$

where $\mathbb{P}_N$ is the *empirical distribution*, assigning mass $1/N$ to each of $X_1, \cdots, X_N$. The quantity $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}}$ measures the absolute deviation between the sample average and the population average. Let us define the *loss class*

$$\mathcal{L} := \{\mathbf{d} \in \mathcal{D} \mapsto \ell_{\boldsymbol{\theta}}(\mathbf{d}) \,|\, \boldsymbol{\theta} \in \Theta\} \tag{5.16}$$

Hence, our aim is now to find whether or not the random variable $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}} = \sup_{\boldsymbol{\theta} \in \Theta} |R_N(\boldsymbol{\theta}) - R(\boldsymbol{\theta})|$ converges to 0 as $N \to \infty$, where $\mathbb{P}$ is the distribution of $\mathbf{d}$. More precisely, we would like to know if there is almost sure convergence (in the uniform norm), and derive rates of convergence. These rates will allow us to find a sufficient condition on the number N of samples to have $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}} \leq \epsilon$, thus giving us a measure of the *learnability* of the loss class $\mathcal{L}$, and by extension a measure of the learnability of the implicit class $\{\mathbf{x}^\star_{\boldsymbol{\theta}}(\cdot) \,|\, \boldsymbol{\theta} \in \Theta\}$ and of the cost class $\{F_{\boldsymbol{\theta}}(\cdot) \,|\, \boldsymbol{\theta} \in \Theta\}$. We will also seek to understand how this sufficient condition depends on the characteristics of the network $\mathcal{G}$ of the routing game, and on the strong monotonicity and Lipschitz constants c and L in the base class $\mathcal{M}$ of edge cost functions (5.7)-(5.8).

Related work. Such convergence properties in the uniform norm is known as *Glivenko-Cantelli properties* [156], [157]. We will leverage a classic result relating the convergence properties of $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}}$ with a notion of complexity of the loss class $\mathcal{L}$, its *Rademacher complexity*. We note that, the *Rademacher complexity* has been studied extensively in the specific context of uniform laws of large numbers and empirical risk minimization, see *e.g.* [12], [11], [99]. However, in contrast to the literature, $\mathbf{d} \mapsto \mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d})$ is from an implicit class since it is defined as solutions to the family of problems $\{\mathrm{VI}(\mathcal{K}(\mathbf{d}), F_{\boldsymbol{\theta}}) \,|\, \mathbf{d} \in \mathcal{D}\}$. Relating to the problem of learning the edge cost functions, $\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d})$ is implicitly indexed by the mapping $F_{\boldsymbol{\theta}}$ over the cost classes $\mathcal{F}_1$ or $\mathcal{F}_2$ defined in (5.9) and (5.10), where $F_{\boldsymbol{\theta}}$ is itself indexed by $\boldsymbol{\theta}$ in the base class $\mathcal{M}$ in (5.7)-(5.8). Translating variations in the parameter space Θ into variations in the loss class $\mathcal{L}$ thus requires sensitivity results in optimization theory such as in [48], [174]. This will allow us to extend notions of learnability to classes of implicit functions defined as Nash equilibria in routing games, and more generally as solutions to convex programs or variational inequality problems. These types of learning problems were adressed in *e.g.* [85], [94], [19], [150], but with little to no analysis on their learnability.

5.5 Motivation

Generalization performance. In practice, having good generalization guarantees for a trained model is extremely important because it gives us a ultimate measure of the quality of the model [78, Chap. 7]. If we draw a new sample $(\mathbf{d}_{N+1}, \mathbf{y}_{N+1})$ from $\mathbb{P}$, and independently

from the training data $(\mathbf{d}_1^N, \mathbf{y}_1^N)$ (on which we trained our model), the *expected prediction error* is $\mathbb{E}[\|\mathbf{y}_{N+1} - h(\mathbf{x}^\star_{\hat{\boldsymbol{\theta}}}(\mathbf{d}_{N+1}))\|] = R(\hat{\boldsymbol{\theta}})$. Hence the population risk $R(\hat{\boldsymbol{\theta}})$ is a measure of the *prediction capability* of $\mathbf{x}^\star_{\hat{\boldsymbol{\theta}}}(\cdot)$ on independent test data. We can relate $R(\hat{\boldsymbol{\theta}})$ to the quantity $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}}$ via the bound

$$R(\hat{\boldsymbol{\theta}}) \leq R_N(\hat{\boldsymbol{\theta}}) + \|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}}$$

which follows by definition of $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}}$. Hence a small value of the empirical risk $R_N(\hat{\boldsymbol{\theta}})$, which is available as the objective in empirical risk minimization methods, does not allow us to assess the prediction capability of the trained model. If $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}}$ is large, $R(\hat{\boldsymbol{\theta}})$ can take large values despite a small $R_N(\hat{\boldsymbol{\theta}})$, and we may be overfitting our training data $(\mathbf{d}_1^N, \mathbf{d}_1^N)$ due to, *e.g.*, a function class $\{\mathbf{x}^\star_{\boldsymbol{\theta}} \,|\, \boldsymbol{\theta} \in \Theta\}$ that is too 'rich' or *complex*. In fact, connections between the complexity of a function class and the Glivenko-Cantelli property are well-known [12], [11], [99] and will be at the heart of our analysis in the remaining Sections.

Excess risk. Our model for the cost functions may not capture the real model since it is a simplified version of it. In other words, $\boldsymbol{\theta}^\star$ may not lie within the index set Θ. Assume that $\hat{\boldsymbol{\theta}} \in \arg\min_{\boldsymbol{\theta}\in\Theta} R_N(\boldsymbol{\theta})$ and $\boldsymbol{\theta}_0 \in \arg\min_{\boldsymbol{\theta}\in\Theta} R(\boldsymbol{\theta})$. We are thus interested in controlling the *excess risk* $R(\hat{\boldsymbol{\theta}}) - R(\boldsymbol{\theta}_0)$, which gives us a measure the prediction quality of the model trained on $(\mathbf{d}_1^N, \mathbf{y}_1^N)$ against the best one we can hope to fit, given the chosen parametrization. We can write

$$\begin{aligned} R(\hat{\boldsymbol{\theta}}) - R(\boldsymbol{\theta}_0) = &[R(\hat{\boldsymbol{\theta}}) - R_N(\hat{\boldsymbol{\theta}})] + [R_N(\hat{\boldsymbol{\theta}}) - R_N(\boldsymbol{\theta}_0)] \\ &+ [R_N(\boldsymbol{\theta}_0) - R(\boldsymbol{\theta}_0)] \end{aligned}$$

The second quantity in the sum is non-positive by definition of $\hat{\boldsymbol{\theta}}$, the third term converges to zero by the (pointwise) *law of large numbers* as $N \to \infty$ since $\boldsymbol{\theta}_0$ is fixed, and the first term requires a uniform bound since $\hat{\boldsymbol{\theta}}$ is random, motivating again the study of $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}} = \sup_{\boldsymbol{\theta}\in\Theta} |R_N(\boldsymbol{\theta}) - R(\boldsymbol{\theta})|$.

5.6 Rademacher complexity and learnability

Given some function class $\mathcal{F}$, a classic approach in the study of uniform laws consists in relating the quantity $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}}$ defined in (5.15) to the *Rademacher complexity* of $\mathcal{F}$. We first denote $(\sigma_1, \cdots, \sigma_N)$ the collection of Rademacher random variables, i.i.d. uniform in $\{\pm 1\}$.

Definition 5.1. *(Rademacher complexity) Given a class $\mathcal{F}$ of real-valued functions with domain $\mathcal{X}$ and a collection $X_1^N := (X_1, \cdots, X_N)$ of random samples within $\mathcal{X}$, the Rademacher complexity of $\mathcal{F}$ is given by*

$$\mathcal{R}_N(\mathcal{F}) := \mathbb{E}_{X,\sigma}\Big[\sup_{f\in\mathcal{F}} \Big| \tfrac{1}{N}\textstyle\sum_{i=1}^N \sigma_i f(X_i)\Big|\Big] \tag{5.17}$$

The Rademacher complexity is the average of the maximum correlation between the vector $(f(X_1), \cdots, f(X_N))$ and the "noise vector" $(\sigma_1, \cdots, \sigma_N)$. Intuitively, as a function class grows, it is easier to find a function that correlates well with a randomly drawn noise vector, making the Rademacher complexity grow. In particular, §3.4 of [5] gives an important result bounding the tail of the probability distribution of the random variable $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}}$ with the Rademacher complexity.

Theorem 5.1. *For any b-uniformly bounded function class $\mathcal{F}$, any positive integer N and any $\delta \in \mathbb{R}_{>0}$, we have*

$$\mathbb{P}\Big[\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}} \leq 2\mathcal{R}_N(\mathcal{F}) + \sqrt{\tfrac{2\,b^2}{N}\ln(\tfrac{1}{\delta})}\Big] \geq 1 - \delta$$

Hence, $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}} = 2\mathcal{R}_N(\mathcal{F}) + O(\frac{1}{\sqrt{N}})$ with "high probability". We now seek to find an upper bound on $\mathcal{R}_N(\mathcal{F})$, which will inform us if $\mathcal{R}_N(\mathcal{F}) \to 0$. In addition, if we can derive a rate of convergence for $\mathcal{R}_N(\mathcal{F})$, we can find a lower bound on N guaranteeing that $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}} \leq \epsilon$ with probability at least $1 - \delta$. It also follows from Theorem 5.1

Corollary 5.1. *For any uniformly bounded function class $\mathcal{F}$, if $\mathcal{R}_N(\mathcal{F}) \to 0$, then $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}} \overset{a.s.}{\to} 0$.*

Proof. For all $\delta \in \mathbb{R}_{>0}$, it follows from Theorem 5.1 that
$\mathbb{P}\Big[\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}} \geq 2\,\mathcal{R}_N(\mathcal{F}) + \delta\Big] \leq \exp\big(-\frac{N\,\delta^2}{2\,b^2}\big)$, thus $\sum_{N=1}^{\infty} \mathbb{P}\big[\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}} \geq 2\,\mathcal{R}_N(\mathcal{F}) + \delta\big] < \infty$. From Borel-Cantelli lemma, there exists, for each $\delta > 0$, a positive integer N_δ such that for all $N \geq N_\delta$, $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}} \leq 2\,\mathcal{R}_N(\mathcal{F}) + \delta$ almost surely. In particular, since $\mathcal{R}_N(\mathcal{F}) \to 0$, then we have $\|\mathbb{P}_N - \mathbb{P}\| \overset{\text{a.s.}}{\to} 0$. □

5.7 Tail bound with the metric entropy

We now turn to the field of *approximation theory*, which combines the notion of *metric entropy*, which was first introduced by Kolmogorov [96], and related notions of the "sizes" of various function classes, see [54], [132], [37]. This will be instrumental to bound the Rademacher complexity of a function class by a function of the *metric entropy*.

Definition 5.2. *(Covering number) A δ-cover of a set $\mathbb{T}$ with respect to a metric ρ is a set $\{\theta^1, \cdots, \theta^N\} \subset \mathbb{T}$ such that for each $\theta \in \mathbb{T}$, there exists some $i \in \{1, \cdots, N\}$ such that $\rho(\theta, \theta^i) \leq \delta$. The δ-covering number $N(\delta, \mathbb{T}, \rho)$ is defined as:*

$$N(\delta, \mathbb{T}, \rho) := \min\{N \mid \{\theta^1, \cdots, \theta^N\} \text{ is a } \delta\text{-cover of } \mathbb{T}\}$$

Definition 5.3. *(Metric entropy) Under the assumptions of Definition 5.2, the metric entropy of $\mathbb{T}$ is defined as the function $\delta \mapsto \log N(\delta, \mathbb{T}, \rho)$.*

Covering of hypercubes. As an illustration, consider the interval $[0, a]$ in $\mathbb{R}$, equipped with the metric $|x - x'|$. If $\delta \geq \frac{a}{2}$ then $\frac{a}{2}$ is a δ-cover of $[0, a]$ and the covering number is $N(\delta, [0, a], |\cdot|) = 1$. Now suppose $\delta < \frac{a}{2}$. We divide the interval $[0, a]$ into $K := \lfloor \frac{a}{2\delta} \rfloor + 1$ sub-intervals[1] of the form $[2\delta(i-1), 2\delta i]$ for $i = 1, \cdots, \lfloor \frac{a}{2\delta} \rfloor$, the last sub-interval being $[2\delta \lfloor \frac{a}{2\delta} \rfloor, a]$. By construction, each sub-interval is of length at most 2δ, and denote $x^1, \cdots, x^K$ the centers of each one of them. For any point $x \in [0, a]$, there is some $j \in \{1, \cdots, K\}$ such that $|x - x^j| \leq \delta$, which shows that $N(\delta, [0, a], |\cdot|) \leq \frac{a}{2\delta} + 1$. This also implies that the hypercube has covering number $N(\delta, [0, a]^d, \|\cdot\|_\infty) \leq \left(\frac{a}{2\delta} + 1\right)^d$ with respect to the infinite norm.

Bounding the Rademacher complexity. We now make precise the connection between the Rademacher complexity (5.17) and the metric entropy. Let us fix a class $\mathcal{F}$ of real-valued functions with domain $\mathcal{X}$, and a collection $x_1^N := \{x_1, \cdots, x_N\}$ of elements of $\mathcal{X}$. We recall that $(\sigma_1, \cdots, \sigma_N)$ is the collection of Rademacher random variables (i.i.d. uniform in $\{\pm 1\}$). The quantity $\sum_{i=1}^N \sigma_i f(x_i)$ that appears in the Rademacher complexity is a *sub-Gaussian process*[2] with respect to the Euclidean norm on the set $\mathcal{F}(x_1^N)$:

$$\mathcal{F}(x_1^N) := \big\{(f(x_1), \cdots, f(x_N)) \,|\, f \in \mathcal{F}\big\}$$

Indeed, if we denote by $f(x_1^N) := (f(x_1), \cdots, f(x_N))$ each element of $\mathcal{F}(x_1^N)$, we have for every, $f, f' \in \mathcal{F}$, and $\lambda \in \mathbb{R}$,

$$\begin{aligned}
\mathbb{E}[e^{\lambda \sum_i \sigma_i (f(x_i) - f'(x_i))}] &= \Pi_i \mathbb{E}[e^{\lambda \sigma_i (f(x_i) - f'(x_i))}] \\
&\leq \Pi_i e^{\frac{\lambda^2 (f(x_i) - f'(x_i))^2}{2}} \\
&= \exp\big(\tfrac{\lambda^2 \|f(x_1^N) - f'(x_1^N)\|_2^2}{2}\big)
\end{aligned}$$

where in the inequality we applied Hoeffding's lemma on each random variable $\sigma_i(f(x_i) - f'(x_i))$. Noting that the expected absolute supremum $\mathbb{E}_\sigma\big[\sup_{f \in \mathcal{F}} |\sum_{i=1}^N \sigma_i f(x_i)|\big]$ of the sub-Gaussian process $\sum_{i=1}^N \sigma_i f(X_i)$ appears in the Rademacher complexity (5.17), leads us to apply Dudley's theorem, see [57] and Chapter 11 of [102].

Theorem 5.2. *(Dudley's theorem) Let $\{X_\theta,\, \theta \in \mathbb{T}\}$ be a zero-mean sub-Gaussian process with respect to the metric ρ. Then $\mathbb{E}\big[\sup_{\theta \in \mathbb{T}} X_\theta\big] \leq 8\sqrt{2} \int_0^\infty \sqrt{\log N(u, \mathbb{T}, \rho)} du$.*

This gives us a bound on the expected suprema of sub-Gaussian processes with the entropy integral.

Proposition 5.1. *For any function class $\mathcal{F}$ such that $0 \in \mathcal{F}$,*

$$\mathcal{R}_N(\mathcal{F}) \leq \tfrac{16\sqrt{2}}{N}\, \mathbb{E}_X\Big[\int_0^\infty \sqrt{\log N(u, \mathcal{F}(X_1^N), \|\cdot\|_2)}\, du\Big]$$

[1] For a scalar $a \in \mathbb{R}$, the notation $\lfloor a \rfloor$ denotes the greatest integer less than or equal to a.

[2] A collection of zero-mean random variables $\{X_\theta,\, \theta \in \mathbb{T}\}$ is a sub-Gaussian process with respect to a metric ρ on $\mathbb{T}$ if, for all $\theta, \theta' \in \mathbb{T}$, and $\lambda \in \mathbb{R}$, we have $\mathbb{E}[e^{\lambda(X_\theta - X_{\theta'})}] \leq \exp\big(\frac{\lambda^2 \rho^2(\theta, \theta')}{2}\big)$

Proof. Denoting $-\mathcal{F} := \{-f \,|\, f \in \mathcal{F}\}$ and the Rademacher process $r_N(f(x_1^N)) := \sum_i \sigma_i f(x_i)$, we have for any collection $x_1^N = \{x_1, \cdots, x_N\}$ of points

$$\begin{aligned}\sup_{f\in\mathcal{F}} |\textstyle\sum_i \sigma_i f(x_i)| &= \sup_{f\in\mathcal{F}\cup-\mathcal{F}} \textstyle\sum_i \sigma_i f(x_i) \\ &\leq \sup_{f\in\mathcal{F}} \textstyle\sum_i \sigma_i f(x_i) + \sup_{f\in-\mathcal{F}} \textstyle\sum_i \sigma_i f(x_i)\end{aligned}$$

where we used that fact that $\sup_{f\in\mathcal{F}} \sum_i \sigma_i f(x_i)$ and $\sup_{f\in-\mathcal{F}} \sum_i \sigma_i f(x_i)$ are non-negative since $0 \in \mathcal{F}$ by assumption. Since the Rademacher random variables are i.i.d. uniform in $\{\pm 1\}$, σ_i and $-\sigma_i$ have same distribution, and $\mathbb{E}_\sigma\big[\sup_{f\in\mathcal{F}} \sum_i \sigma_i f(x_i)\big] = \mathbb{E}_\sigma\big[\sup_{f\in-\mathcal{F}} \sum_i \sigma_i f(x_i)\big]$. And by Dudley's theorem, this last quantity is less than $\frac{8\sqrt{2}}{N} \displaystyle\int_0^\infty \sqrt{\log N(u, \mathcal{F}(x_1^N), \|\cdot\|_2)}\, du$ since we know that the process $\{\sum_{i=1}^N \sigma_i f(x_i) \,|\, f \in \mathcal{F}\}$ is sub-Gaussian with respect to $\|\cdot\|_2$ on the set $\mathcal{F}(x_1^N)$. Finally,

$$\begin{aligned}\mathcal{R}_N(\mathcal{F}) &= \tfrac{1}{N}\mathbb{E}_{X,\sigma}\big[\sup_{f\in\mathcal{F}} \big|\textstyle\sum_{i=1}^N \sigma_i f(X_i)\big|\big] \\ &\leq \tfrac{2}{N}\mathbb{E}_{X,\sigma}\big[\sup_{f\in\mathcal{F}} \textstyle\sum_{i=1}^N \sigma_i f(X_i)\big] \\ &\leq \tfrac{16\sqrt{2}}{N}\,\mathbb{E}_X\Big[\displaystyle\int_0^\infty \sqrt{\log N(u, \mathcal{F}(X_1^N), \|\cdot\|_2)}\, du\Big]\end{aligned}$$

□

Theorem 5.3. *For any indexed class $\mathcal{F} = \{f_{\boldsymbol{\theta}} \,|\, \boldsymbol{\theta} \in \Theta\}$ of real-valued functions which is b-uniformly bounded and L-smoothly parametrized with respect to a norm $\|\cdot\|_\Theta$ on Θ, and such that $0 \in \mathcal{F}$,*

$$\mathcal{R}_N(\mathcal{F}) \leq \tfrac{16\sqrt{2}L}{\sqrt{N}} \displaystyle\int_0^\infty \sqrt{\log N(v, \Theta, \|\cdot\|_\Theta)}\, dv$$

Proof. For any $f_{\boldsymbol{\theta}}, f_{\boldsymbol{\theta}'} \in \mathcal{F}$, and collection x_1^N

$$\begin{aligned}\|f_{\boldsymbol{\theta}}(x_1^N) - f_{\boldsymbol{\theta}'}(x_1^N)\|_2^2 &= \textstyle\sum_{i=1}^N (f_{\boldsymbol{\theta}}(x_i) - f_{\boldsymbol{\theta}'}(x_i))^2 \\ &\leq \textstyle\sum_{i=1}^N |||f_{\boldsymbol{\theta}} - f_{\boldsymbol{\theta}'}|||^2 \\ &\leq NL^2\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\Theta^2\end{aligned}$$

Thus $N(\delta, \mathcal{F}(x_1^N), \|\cdot\|_2) \leq N\big(\frac{\delta}{L\sqrt{N}}, \Theta, \|\cdot\|_\Theta\big)$ by definition of the covering number. Hence, a Lipschitz parameterization allows us to translate a cover of the parameter space Θ into a cover of the data-dependent function space $\mathcal{F}(x_1^N)$. Applying Proposition 5.1 with the change of variable $v := \frac{u}{L\sqrt{N}}$ in the entropy integral gives the claimed result. □

Hence, a Lipschitz parametrization allows us to bound the Rademacher complexity of $\mathcal{F}$ by the entropy integral of its parameter space Θ

Smooth parametrization on hypercubes. As an illustration, assume $\mathcal{F} := \{f_{\boldsymbol{\theta}} \,|\, \boldsymbol{\theta} \in [0,a]^d\}$ is L-smoothly parametrized with respect to the infinite norm $\|\cdot\|_\infty$ on the hypercube $[0,a]^d$. Using our earlier result on the covering of hypercubes, the entropy integral of $[0,a]^d$ is

$$\begin{aligned}\int_0^\infty \sqrt{\log N(\delta, [0,a]^d, \|\cdot\|_\infty)}\, d\delta &\leq \int_0^{\frac{a}{2}} \sqrt{d \log\left(\tfrac{a}{2\delta}+1\right)}\, d\delta \\ &= \tfrac{a\sqrt{d}}{2} \int_0^1 \sqrt{\log(\tfrac{1}{u}+1)}\, du\end{aligned}$$

Theorem 5.3 gives $\mathcal{R}_N(\mathcal{F}) \leq \frac{8\,aL\sqrt{2d}}{\sqrt{N}} \int_0^1 \sqrt{\log(\frac{1}{u}+1)}\, du$.

5.8 Smooth parametrization of VIP's

The implicit function $\mathbf{p} \mapsto \mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p})$, mapping from the predictor space $\mathcal{D}$ to the state space $\mathbb{R}^n$ is defined as the solution of the variational inequality problem $\mathrm{VI}(\mathcal{K}(\mathbf{d}), F_{\boldsymbol{\theta}})$, where $F_{\boldsymbol{\theta}}$ is chosen within a function class $\{F_{\boldsymbol{\theta}} \,|\, \boldsymbol{\theta} \in \Theta\}$. Hence, requiring $\{F_{\boldsymbol{\theta}} \,|\, \boldsymbol{\theta} \in \Theta\}$ to be smoothly parametrized does not allow us to directly apply Theorem 5.3 to bound the Rademacher complexity by the entropy integral of Θ. To resolve this difficulty, we use a modified version of the results in [174] on the Lipschitz continuity of solutions to variational inequalities. Let us define $P_{\mathcal{X}}$ the Euclidean projection onto a subset $\mathcal{X}$ of $\mathbb{R}^n$.

Lemma 5.1. *For any compact convex subset $\mathcal{K}$ of $\mathbb{R}^n$, for any c-strong-monotone, L-Lipschitz mapping F, and for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n$, and denoting $k := \sqrt{1-c^2/L^2}$, we have*

$$\|P_{\mathcal{K}}(\boldsymbol{x} - \tfrac{c}{L^2}F(\boldsymbol{x})) - P_{\mathcal{K}}(\boldsymbol{x}' - \tfrac{c}{L^2}F(\boldsymbol{x}'))\|_2 \leq k\|\boldsymbol{x} - \boldsymbol{x}'\|_2$$

Proof. Let us denote $\alpha := \frac{c}{L^2}$. Since $\mathcal{K}$ is convex, $P_{\mathcal{K}}$ is 1-Lipschitz and the left-hand side of the inequality is less than $\|(\mathbf{x}-\mathbf{x}') - \alpha(F(\mathbf{x}) - F(\mathbf{x}'))\|_2$. The square of this quantity is $\|\mathbf{x}-\mathbf{x}'\|_2^2 + \alpha^2\|F(\mathbf{x}) - F(\mathbf{x}')\|_2^2 - 2\alpha\langle \mathbf{x}-\mathbf{x}', F(\mathbf{x}) - F(\mathbf{x}')\rangle$. By Lipschitz continuity, $\|F(\mathbf{x}) - F(\mathbf{x}')\|_2^2 \leq L^2\|\mathbf{x}-\mathbf{x}'\|_2^2$, and by strong monotonicity, $-\langle \mathbf{x}-\mathbf{x}', F(\mathbf{x}) - F(\mathbf{x}')\rangle \leq -c\|\mathbf{x}-\mathbf{x}'\|_2^2$. Hence the left-hand side of the inequality is less than $(1+\alpha^2L^2 - 2\alpha\, c)\|\mathbf{x}-\mathbf{x}'\|_2^2$. Noting that $1+\alpha^2L^2 - 2\alpha\, c = 1 - c^2/L^2 \in [0,1)$ since we have necessarily $L \geq c$, we take the square root and obtain our claim. □

Proposition 5.2. *(Smoothly parametrized VIP) For any parametric variational inequality problem, $\{VI(\mathcal{K}(\boldsymbol{d}), F_{\boldsymbol{\theta}}) \,|\, (\boldsymbol{\theta}, \boldsymbol{d}) \in \Theta \times \mathcal{D}\}$, such that*

(a) $\mathcal{K}(\boldsymbol{d})$ is compact convex subset of $\mathbb{R}^n$ for all $\boldsymbol{d} \in \mathcal{D}$

(b) $F_{\boldsymbol{\theta}}$ is c-strong-monotone and L-Lipschitz for all $\boldsymbol{\theta} \in \Theta$

(c) $|||F_{\boldsymbol{\theta}} - F_{\boldsymbol{\theta}'}||| \leq L_\Theta\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\Theta$ for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$

the unique solution $\boldsymbol{x}^\star_{\boldsymbol{\theta}}(\boldsymbol{d})$ to $VI(\mathcal{K}(\boldsymbol{d}), F_{\boldsymbol{\theta}})$ satisfies, for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$ and for all $\boldsymbol{d}, \boldsymbol{d}' \in \mathcal{D}$

$$\|\boldsymbol{x}^\star_{\boldsymbol{\theta}'}(\boldsymbol{d}) - \boldsymbol{x}^\star_{\boldsymbol{\theta}}(\boldsymbol{d})\|_2 \leq \frac{cL_\Theta}{L^2\left(1-\sqrt{1-\frac{c^2}{L^2}}\right)}\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\Theta$$

Proof. Since the unique solution $\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d})$ of $\mathrm{VI}(\mathcal{K}(\mathbf{d}), F_{\boldsymbol{\theta}})$ can be equivalently characterized as the unique solution of the fixed point problem $\mathbf{x} = P_{\mathcal{K}(\mathbf{d})}(\mathbf{x} - \frac{c}{L^2}F_{\boldsymbol{\theta}}(\mathbf{x}))$, see [61, §1.5.8.], we define, for all $(\boldsymbol{\theta}, \mathbf{d}, \mathbf{x}) \in \Theta \times \mathcal{D} \times \mathbb{R}^n$

$$Q_{\boldsymbol{\theta},\mathbf{d}}(\mathbf{x}) := P_{\mathcal{K}(\mathbf{d})}(\mathbf{x} - \tfrac{c}{L^2}F_{\boldsymbol{\theta}}(\mathbf{x}))$$

We have for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$, and for all $\mathbf{d} \in \mathcal{D}$

$$\begin{aligned}\|\mathbf{x}^\star_{\boldsymbol{\theta}'}(\mathbf{d}) - \mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d})\|_2 &= \|Q_{\boldsymbol{\theta}',\mathbf{d}}(\mathbf{x}^\star_{\boldsymbol{\theta}'}(\mathbf{d})) - Q_{\boldsymbol{\theta},\mathbf{d}}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d}))\|_2 \\ &\leq \|Q_{\boldsymbol{\theta}',\mathbf{d}}(\mathbf{x}^\star_{\boldsymbol{\theta}'}(\mathbf{d})) - Q_{\boldsymbol{\theta}',\mathbf{d}}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d}))\|_2 \\ &\quad + \|Q_{\boldsymbol{\theta}',\mathbf{d}}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d})) - Q_{\boldsymbol{\theta},\mathbf{d}}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d}))\|_2\end{aligned}$$

From Lemma 5.1, the first term in the sum is less than $\sqrt{1 - \frac{c^2}{L^2}}\,\|\mathbf{x}^\star_{\boldsymbol{\theta}'}(\mathbf{d}) - \mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d})\|_2$. Note that $1 - \frac{c^2}{L^2} \in [0, 1)$. And $P_{\mathcal{K}(\mathbf{d})}$ being 1-Lipschitz, the second term in the sum, denoted by T, is upper bounded by

$$\begin{aligned}T &\leq \|\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d}) - \tfrac{c}{L^2}F_{\boldsymbol{\theta}}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d})) - (\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d}) - \tfrac{c}{L^2}F_{\boldsymbol{\theta}'}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d})))\|_2 \\ &= \tfrac{c}{L^2}\|F_{\boldsymbol{\theta}}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d})) - F_{\boldsymbol{\theta}'}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d}))\|_2 \\ &\leq \tfrac{c\,L_\Theta}{L^2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\Theta\end{aligned}$$

Putting together both bounds and re-arranging the terms proves our claim. □

Combining Theorem 5.3 and Proposition 5.2, we obtain a tail bound on the distribution of the random variable $\sup_{\boldsymbol{\theta}\in\Theta} |R_N(\boldsymbol{\theta}) - R(\boldsymbol{\theta})| = \|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}}$, which we recall is a function of the collection $\mathbf{d}_1^N$ of i.i.d. samples drawn from a distribution $\mathbb{P}$ over the predictor space $\mathcal{D}$.

Theorem 5.4. *For any parametric variational inequality problem* $\{\, VI(\mathcal{K}(\boldsymbol{d}), F_{\boldsymbol{\theta}}) \,|\, (\boldsymbol{\theta}, \boldsymbol{d}) \in \Theta \times \mathcal{D}\}$ *satisfying assumptions*

(a) $\mathcal{K}(\boldsymbol{d})$ *is compact convex subset of* $\mathbb{R}^n$ *for all* $\boldsymbol{d} \in \mathcal{D}$

(b) $F_{\boldsymbol{\theta}}$ *is c-strong-monotone and L-Lipschitz for all* $\boldsymbol{\theta} \in \Theta$

(c) $|||F_{\boldsymbol{\theta}} - F_{\boldsymbol{\theta}'}||| \leq L_\Theta\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\Theta$ *for all* $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$

(d) $diam_{\|\cdot\|_\Theta}(\Theta) < \infty$

and for any L_h*-Lipschitz function* $h : \mathbb{R}^n \to \mathbb{R}^m$*, the loss class* $\mathcal{L}$ *in* (5.16) *is* $\tilde{L}$*-smoothly parametrized and b-uniformly bounded with constants*

$$\tilde{L} := \frac{c\,L_h L_\Theta}{L^2\left(1 - \sqrt{1 - \frac{c^2}{L^2}}\right)} \tag{5.18}$$

$$b := \tilde{L}\; diam_{\|\cdot\|_\Theta}(\Theta) \tag{5.19}$$

and we have $\mathcal{R}_N(\mathcal{L}) \leq \frac{16\sqrt{2}\tilde{L}}{\sqrt{N}} \displaystyle\int_0^\infty \sqrt{\log N(v, \Theta, \|\cdot\|_\Theta)}\, dv.$

Proof. Since $\ell_{\boldsymbol{\theta}^\star} = 0 \in \mathcal{L}$, we only need to show that the loss class $\mathcal{L}$ is smoothly parametrized and uniformly bounded with given in (5.18) and (5.19), since the tail bound then follows immediately from Theorem 5.3. In order to simplify notation, we pose $h^\star_{\boldsymbol{\theta}}(\mathbf{d}) = h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d}))$, then the loss is $\ell_{\boldsymbol{\theta}}(\mathbf{d}) = \|h(\mathbf{x}^\star_{\boldsymbol{\theta}^\star}(\mathbf{d})) - h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d}))\| = \|h^\star_{\boldsymbol{\theta}^\star}(\mathbf{d}) - h^\star_{\boldsymbol{\theta}}(\mathbf{d})\|$. We have, for all $\mathbf{d}$ and for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$

$$\begin{aligned}
|\ell_{\boldsymbol{\theta}}(\mathbf{d}) - \ell_{\boldsymbol{\theta}'}(\mathbf{d})| &\leq \|(h^\star_{\boldsymbol{\theta}^\star}(\mathbf{d}) - h^\star_{\boldsymbol{\theta}}(\mathbf{d})) - (h^\star_{\boldsymbol{\theta}^\star}(\mathbf{d}) - h^\star_{\boldsymbol{\theta}'}(\mathbf{d}))\| \\
&= \|h^\star_{\boldsymbol{\theta}}(\mathbf{d}) - h^\star_{\boldsymbol{\theta}'}(\mathbf{d})\| \\
&\leq L_h \|\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{d}) - \mathbf{x}^\star_{\boldsymbol{\theta}'}(\mathbf{d})\|_2 \\
&\leq \tfrac{c\, L_h L_\Theta}{L^2(1-\sqrt{1-c^2/L^2})} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\Theta
\end{aligned}$$

where the first inequality is obtained from the 1-Lipschitz continuity of any norm (from the triangle inequality), and the third inequality from Proposition 5.2. This gives us the Lipschitz constant $\tilde{L}$. And for all $(\boldsymbol{\theta}, \mathbf{d}) \in \Theta \times \mathcal{D}$, we have $\ell_{\boldsymbol{\theta}}(\mathbf{d}) = \|h^\star_{\boldsymbol{\theta}^\star}(\mathbf{d}) - h^\star_{\boldsymbol{\theta}}(\mathbf{d})\| \leq \tilde{L}\|\boldsymbol{\theta} - \boldsymbol{\theta}^\star\|_\Theta \leq \tilde{L}\, \mathrm{diam}_{\|\cdot\|_\Theta}(\Theta)$. Hence $\mathcal{L}$ is $(\tilde{L}\, \mathrm{diam}_{\|\cdot\|_\Theta}(\Theta))$-uniformly bounded. □

In the setting of Theorem 5.4, assume the entropy integral $\displaystyle\int_0^\infty \sqrt{\log N(v, \Theta, \|\cdot\|_\Theta)}\, dv$ of the index set Θ finite. Then $\mathcal{R}_N(\mathcal{L}) \to 0$, and Corollary 5.1 states that $\|\mathbb{P}_N - \mathbb{P}\|_\mathcal{L} \overset{a.s.}{\to} 0$. Using the fact that $\mathcal{R}_N(\mathcal{L}) = O(\frac{1}{\sqrt{N}})$ strengthens the result

Corollary 5.2. *Consider the setting of Theorem 5.4. If the entropy integral defined by* $\displaystyle\int_0^\infty \sqrt{\log N(v, \Theta, \|\cdot\|_\Theta)}\, dv$ *is finite, then, for all* $\alpha \in (0, \frac{1}{2})$*, the loss class* $\mathcal{L}$ *defined in* (5.16) *is such that* $\|\mathbb{P}_N - \mathbb{P}\|_\mathcal{L} = O(\frac{1}{N^\alpha})$ *almost surely.*

Proof. It follows from Theorem 5.4 that there exists a positive scalar κ such that $\mathcal{R}_N(\mathcal{L}) \leq \frac{\kappa}{\sqrt{N}}$. From Theorem 5.1, $\mathbb{P}\Big[\|\mathbb{P}_N - \mathbb{P}\|_\mathcal{L} \geq \frac{2\kappa}{\sqrt{N}} + \delta\Big] \leq \exp\big(-\frac{N\,\delta^2}{2\,b^2}\big)$ for all $\delta \in \mathbb{R}_{>0}$. Let α be a scalar in $(0, \frac{1}{2})$. Substituting $\delta := \frac{1}{N^\alpha}$, we have $\mathbb{P}\Big[\|\mathbb{P}_N - \mathbb{P}\|_\mathcal{L} \geq \frac{2\kappa}{\sqrt{N}} + \frac{1}{N^\alpha}\Big] \leq \exp\big(-\frac{N^{1-2\alpha}}{2\,b^2}\big)$. Hence, $\sum_{N=1}^\infty \mathbb{P}\Big[\|\mathbb{P}_N - \mathbb{P}\|_\mathcal{L} \geq \frac{2\kappa}{\sqrt{N}} + \frac{1}{N^\alpha}\Big] \leq \sum_{N=1}^\infty e^{-\frac{N^{1-2\alpha}}{2\,b^2}} < \infty$ since $1 - 2\alpha > 0$. From Borel-Cantelli lemma, there exists a positive integer N_0 such that for all $N \geq N_0$, we have $\|\mathbb{P}_N - \mathbb{P}\|_\mathcal{L} \leq \frac{2\kappa}{\sqrt{N}} + \frac{1}{N^\alpha}$ almost surely. We conclude our proof by noting that $\frac{2\kappa}{\sqrt{N}} = O(\frac{1}{N^\alpha})$ since $\alpha \in (0, \frac{1}{2})$. □

The results derived up to now, specifically Theorem 5.4 and Corollary 5.2, apply to any parametric variational inequality problem *and* any parametric *convex optimization problem* $\{\min\, f_{\boldsymbol{\theta}}(\mathbf{x})$ s.t. $\mathbf{x} \in \mathcal{K}(\mathbf{d}) \,|\, (\mathbf{d}, \boldsymbol{\theta}) \in \mathcal{D} \times \Theta\}$ with $f_{\boldsymbol{\theta}}$ belonging to an indexed-family of convex differentiable potentials. Indeed, the variational inequality $\mathrm{VI}(\mathcal{K}(\mathbf{d}), \nabla f_{\boldsymbol{\theta}})$, with mapping substituted with the gradient $\nabla f_{\boldsymbol{\theta}}$ of the potential $f_{\boldsymbol{\theta}}$, is known as the *first-order optimality condition* for convex programs, see *e.g.* [27, §4.2.3.], where *c-strong-monotonicity* and *L-Lipschitz continuity* of $\nabla f_{\boldsymbol{\theta}}$ are respectively equivalent to *c-strong-convexity* and *L-Lipschitz gradient* of $f_{\boldsymbol{\theta}}$.

5.9 Application to selfish routing

In the remainder of the chapter, we seek to understand how the *learnability* of the edge cost functions depends on the characteristics of the network $\mathcal{G}$ of the routing game, and on the strong monotonicity and Lipschitz constants of the functions within the base class $\mathcal{M}$. With this objective in mind, let $|||F|||_{\mathcal{F}} = \sup_{\mathbf{x}\in[0,\bar{d}]^{\mathcal{E}}} \|F(\mathbf{x})\|_2$ be a metric on $\mathcal{F}_1, \mathcal{F}_2$ defined in (5.9) and (5.10), $\|f\|_\infty = \sup_{x\in[0,1]} |f(x)|$ be a metric on $\mathcal{M}$ (the index set of $\mathcal{F}_2$), and let $|||\{f_e(\cdot)\}_e|||_{\mathcal{M}^{\mathcal{E}}} = \max_{e\in\mathcal{E}} \|f_e\|_\infty$ be a metric on $\mathcal{M}^{\mathcal{E}}$ (the index set of $\mathcal{F}_1$).

Lemma 5.2. *The function classes $\mathcal{F}_1$ and $\mathcal{F}_2$ defined in* (5.9) *and* (5.10) *are both $\sqrt{|\mathcal{E}|}$-smoothly parametrized over respective base classes $\mathcal{M}^{\mathcal{E}}$ and $\mathcal{M}$, where $\mathcal{M}$ is given in* (5.7)-(5.8).

Proof. We prove our claim for $\mathcal{F}_1$, the proof for $\mathcal{F}_2$ being similar. For all $F_{\mathbf{f}}, F_{\mathbf{g}} \in \mathcal{F}_1$, we have

$$\begin{aligned}
|||F_{\mathbf{f}} - F_{\mathbf{g}}|||_{\mathcal{F}}^2 &= \sup_{\mathbf{x}\in[0,\bar{d}]^{\mathcal{E}}} \textstyle\sum_e (f_e(x_e/m_e) - g_e(x_e/m_e))^2 \\
&\le \textstyle\sum_e (\sup_{x_e\in[0,\bar{d}]} |f_e(x_e/m_e) - g_e(x_e/m_e)|)^2 \\
&\le \textstyle\sum_e (\sup_{x\in[0,1]} |f_e(x) - g_e(x)|)^2 \\
&= \textstyle\sum_e \|f_e - g_e\|_\infty^2 \\
&\le |\mathcal{E}|\, |||\mathbf{f} - \mathbf{g}|||_{\mathcal{M}^{\mathcal{E}}}^2 \qquad \square
\end{aligned}$$

Hence the cost classes $\mathcal{F}_1$ and $\mathcal{F}_2$ satisfy assumption (c) in Theorem 5.4 with $L_\Theta := \sqrt{|\mathcal{E}|}$. Assumption (b) is satisfied by design since $\mathcal{F}_1$ and $\mathcal{F}_2$ only contain $\big(\frac{L}{\min_{e\in\mathcal{E}} m_e}\big)$-Lipschitz and $\big(\frac{c}{\max_{e\in\mathcal{E}} m_e}\big)$-strong-monotone mappings. Assumption (a) follows from the definition of the routing game, and Assumption (d) is satisfied since $\mathrm{diam}_{\|\cdot\|_\infty}(\mathcal{M}) = \mathrm{diam}_{|||\cdot|||}(\mathcal{M}^{\mathcal{E}}) = L - c$. Theorem 5.4 then gives us a bound on the complexity of the loss class $\mathcal{R}_N(\mathcal{L})$ in terms of the entropy integral of the base class $\mathcal{M}^{\mathcal{E}}$ if the cost class is $\mathcal{F}_1$, or in terms of the entropy integral of the base class $\mathcal{M}$ if the cost class is $\mathcal{F}_2$.

Lemma 5.3. *The base class $\mathcal{M}$ defined in* (5.7)-(5.8) *has metric entropy* $\log N(\delta, \mathcal{M}, \|\cdot\|_\infty) \le \big(\frac{L-c}{\delta} + 1\big)\log 2$ *for $\delta \in (0, \frac{L-c}{2})$ and metric entropy zero for $\delta \ge \frac{L-c}{2}$.*

Corollary 5.3. *The base class $\mathcal{M}$ defined in* (5.7)-(5.8) *has entropy integral defined by* $\int_0^\infty \sqrt{\log N(\delta, \mathcal{M}, \|\cdot\|_\infty)}\, d\delta$ *that is less than* $(L-c)\sqrt{\log 2}\int_0^{\frac{1}{2}} \sqrt{\frac{1}{u} + 1}\, du$.

The proofs of Lemma 5.3 and Corollary 5.3 appear in the Appendix. Corollary 5.3 gives us a concrete upper bound on on the class complexity $\mathcal{R}_N(\mathcal{L})$ if the cost class is $\mathcal{F}_2$. Let us define the constant κ and the ratio r of the minimum edge capacity over the maximum edge

capacity

$$\kappa := 16\sqrt{2\log 2}\int_0^{\frac{1}{2}} \sqrt{\tfrac{1}{u}+1}\,du \approx 29 \tag{5.20}$$

$$r := \frac{\min_{e\in\mathcal{E}} m_e}{\max_{e\in\mathcal{E}} m_e} \tag{5.21}$$

Theorem 5.5. *For the routing game with edge cost functions parametrized by the cost class $\mathcal{F}_2$ in (5.10) with base class $\mathcal{M}$ in (5.7)-(5.8), given the predictor-response relationship in (5.11) with a L_h-Lipschitz observation mapping h, and given constants κ and r in (5.20) and (5.21), the Rademacher complexity $\mathcal{R}_N(\mathcal{L})$ of the loss class $\mathcal{L}$ in (5.16) is bounded by*

$$\mathcal{R}_N(\mathcal{L}) \leq \frac{\kappa\, L_h\sqrt{|\mathcal{E}|}\min_{e\in\mathcal{E}} m_e}{\sqrt{N}}\,\mathcal{J}(c,L,r) \tag{5.22}$$

$$\mathcal{J}(c,L,r) := \frac{r\,c\,(L-c)}{L^2\left(1-\sqrt{1-\frac{c^2}{L^2}\,r^2}\right)} \tag{5.23}$$

If we allow for different shapes of the cost functions between edges, *i.e.* the cost class is $\mathcal{F}_1$, we note that its index set $\mathcal{M}^{\mathcal{E}}$ has metric entropy $\log N(\delta, \mathcal{M}^{\mathcal{E}}, \|\!|\cdot|\!\|_{\mathcal{M}^{\mathcal{E}}}) = |\mathcal{E}|\log N(\delta,\mathcal{M},\|\cdot\|_\infty)$, hence an additional factor $\sqrt{|\mathcal{E}|}$ appears in the right-hand side of (5.22). We recall that $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}}$ is equal to $\sup_{\boldsymbol{\theta}\in\Theta}|R_N(\boldsymbol{\theta}) - R(\boldsymbol{\theta})|$, the absolute deviation between the population and empirical risks. Combining with Theorem 5.1 and noting that $\mathcal{L}$ is uniformly bounded by $L_h \min_{e\in\mathcal{E}} m_e(L-c)\sqrt{|\mathcal{E}|}\mathcal{J}(c,L,r)$, we obtain

Theorem 5.6. *Given the setting of Theorem 5.5 and $\epsilon,\delta\in\mathbb{R}_{>0}$, a sufficient condition for having $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}} \leq \epsilon$ with probability at least $1-\delta$ is for the sample size N to be*

$$\sqrt{N} \geq \frac{L_h\sqrt{|\mathcal{E}|}\min_{e\in\mathcal{E}} m_e\,\mathcal{J}(c,L,r)\left(2\kappa + (L-c)\sqrt{2\log\left(\frac{1}{\delta}\right)}\right)}{\epsilon}$$

Asymptotic analysis. We now seek to understand how the above lower bound, which is a measure of the learnability of the edge cost functions, is affected by the ratio $r := \frac{\min_{e\in\mathcal{E}} m_e}{\max_{e\in\mathcal{E}} m_e}$, along with the Lipschitz constant L and the strong monotonicity constant c of the functions in the base class $\mathcal{M}$. We note that the ratio $\frac{c}{L}$, which is always in $(0,1]$, measures by how much the elements of $\mathcal{M}$ deviates from an affine function. In particular, if $c = L$, the base class $\mathcal{M}$ is reduced to the singleton $\{x\in[0,1]\mapsto c\,x\}$.

Proposition 5.3. *Denoting $f(s)\sim g(s)$ if $\frac{f(s)}{g(s)} \underset{s\to s_0}{\to} 1$ (asymptotic equivalence), $\mathcal{J}(c,L,r)$ in (5.23) is such that*

$$\tfrac{c}{L}\to 0 \quad\Longrightarrow\quad \mathcal{J}(c,L,r)\sim \tfrac{2L}{c\,r} \tag{5.24}$$

$$r\to 0 \quad\Longrightarrow\quad \mathcal{J}(c,L,r)\sim \tfrac{2(L-c)}{c\,r} \tag{5.25}$$

$$\tfrac{c}{L}\to 1 \quad\Longrightarrow\quad \mathcal{J}(c,L,r)\sim \tfrac{r(1-\frac{c}{L})}{1-\sqrt{1-r^2}}\to 0 \tag{5.26}$$

Proof. Denoting $\alpha := \frac{c}{L}$, we can re-write $\mathcal{J}(c, L, r)$ as

$$\mathcal{J}(c, L, r) = \frac{L(1-\alpha)}{\frac{c\,r}{(\alpha\,r)^2}\left(1-\sqrt{1-(\alpha\,r)^2}\right)} \tag{5.27}$$

since $\frac{1}{x}\left(1-\sqrt{1-x}\right) = \frac{1}{x}\left(\frac{1}{2}x + o(x)\right) \to \frac{1}{2}$, whenever $\alpha\,r = o(1)$, the denominator in (5.27) is asymptotically equivalent to $\frac{c\,r}{2}$, which proves (5.25). In addition, if $\alpha = o(1)$, the numerator in (5.27) converges to $\sqrt{L}$, which proves (5.24). To prove (5.26), we re-write $\mathcal{J}(c, L, r) = \frac{r\,\alpha(1-\alpha)}{1-\sqrt{1-\alpha^2 r^2}}$ □

Theorem 5.6 and Proposition 5.3 have the following interpretations: *to maintain a high consistency of the empirical risk, the sample size N needs to grow proportionally to (a) L when L grows; (b) $\frac{1}{c^2}$ when c vanishes; (c)$\frac{1}{r^2}$ when r vanishes; (d) $|\mathcal{E}|^2$ when $|\mathcal{E}|$ grows and the edges costs are allowed to be different functions of the normalized flows $\{\frac{x_e}{m_e}\}_{e\in\mathcal{E}}$; (e) $|\mathcal{E}|$ when $|\mathcal{E}|$ grows and the edges costs are the same function of the normalized flows $\{\frac{x_e}{m_e}\}_{e\in\mathcal{E}}$.*

5.10 Conclusion

We studied the learnability of the edge cost functions in the routing game from observations of the population demand and the equilibrium flow that it induces. We motivated the analysis of the uniform laws for the loss class since it plays a key role in understanding the consistency of the empirical risk as a statistical estimator for the quality of the learned model. We gave precise results on the tail bound of the uniform deviation between the population risk and empirical risk in terms of the entropy integral of their index set. Using sensitivity analysis in optimization theory, we then argued that variations in the index set can be smoothly translated into variations in the loss class, which allows us to derive lower bounds on the sample size required to have constant prediction capabilities, as a function of the characteristics of the routing game. Our results are very general since they hold independently of the sample distribution and of the edge costs parametrization. And while deriving them in the context of the routing game, we provided results that hold for general convex optimization and variational inequality problems with monotone operators.

Chapter 6

Upper bounds on the prediction error

We consider a class of supervised learning problems in which the response is implicitly defined as a solution of a convex program depending on the predictors, and for which the goal is to estimate the convex objective. Specifically, we focus on the learnability of this class of learning problems, where the learnability is measured as the number of samples needed to have a small prediction error. Bounds on the sample size depend on the complexity of a class of implicit functions mapping the predictor to the solution of the convex program. Using sensitivity analysis in optimization, we characterize the complexity of the implicit class, and then we leverage results on the complexity of function classes and in approximation theory to obtain tail bounds on the prediction error as a function of the characteristics of the convex program to be estimated. This gives sufficient conditions on the size of the training data needed to have good generalization properties, as a function of the complexity of the class of objective functions to be learned.

6.1 Introduction

In supervised learning, the goal is to predict a response variable $\mathbf{y} \in \mathcal{Y}$ from observations of the random predictor $\mathbf{p} \in \mathcal{P}$, by estimating a function from $\mathcal{P}$ to $\mathcal{Y}$ that is generally *explicit* in $\mathbf{p}$. Examples include linear and logit functions, random forests, neural networks, see *e.g.* [78] for an overview. In this chapter, for each predictor $\mathbf{p} \in \mathcal{P}$, we consider the problem of learning a parametric convex objective $f(\cdot, \mathbf{p}) : \mathbb{R}^n \to \mathbb{R}$ associated to a *convex optimization program (COP)* of the form

$$\min\ f(\mathbf{x}, \mathbf{p}) \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{D}(\mathbf{p}) \tag{6.1}$$

where $\{\mathcal{D}(\mathbf{p})\}_{\mathbf{p}\in\mathcal{P}}$ is a family of convex compact domains within $\mathbb{R}^n$. We suppose that the true objective $f_{\boldsymbol{\theta}^\star}$ belongs to an indexed family $\{f_{\boldsymbol{\theta}}(\cdot, \mathbf{p}) : \mathbb{R}^n \to \mathbb{R} \,|\, (\boldsymbol{\theta}, \mathbf{p}) \in \Theta \times \mathcal{P}\}$ of objectives $f_{\boldsymbol{\theta}}$ that are differentiable, strongly convex with parameter c, and Lipschitz gradient with constant L for each $(\boldsymbol{\theta}, \mathbf{p}) \in \Theta \times \mathcal{P}$. Hence, for each $(\boldsymbol{\theta}, \mathbf{p})$, the COP defined by $(f_{\boldsymbol{\theta}}(\cdot, \mathbf{p}), \mathcal{D}(\mathbf{p}))$, *i.e.* the program $\min_{\mathbf{x}\in\mathcal{D}(\mathbf{p})} f_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p})$, has a unique solution $\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p})$, see [27].

The well-defined map $\mathbf{x}^\star_{\boldsymbol{\theta}}(\cdot) : \mathcal{P} \to \mathbb{R}^n$ is thus an *implicit function* of the predictor $\mathbf{p} \in \mathcal{P}$, since evaluating it at each point requires to solve a COP. We note that Θ is the set of allowable parameters. It can be finite-dimensional, the learning problem is then parametric, or it can be a function class, in which case the problem is non-parametric. Formally, we endow the predictor space $\mathcal{P}$ with a structure of measure space $(\mathcal{P}, \Sigma, \mathbb{P})$ where Σ is a σ-algebra of measurable sets, and $\mathbb{P}$ is a probability measure. The learning problem consists in estimating $\boldsymbol{\theta} \in \Theta$ from a collection of i.i.d. samples $\{(\mathbf{p}_i, \mathbf{y}_i)\}_{i=1}^N$, with the $\mathbf{p}_i$'s drawn from $\mathbb{P}$, and with relationship given by

$$\mathbf{y}_i = h(\mathbf{x}^\star_{\boldsymbol{\theta}^\star}(\mathbf{p}_i)) + \boldsymbol{\epsilon}_i, \quad i \in [N] \tag{6.2}$$

where $[N]$ denotes $\{1, \cdots, N\}$, $\boldsymbol{\epsilon}_i \in \mathbb{R}^m$ is a random variable representing the noise in the i^{th} response variable $\mathbf{y}_i$, and $h(\cdot)$ is an L_h-Lipschitz observation model continuously mapping from the state space $\mathbb{R}^n$ to an observed space in $\mathbb{R}^m$. A classic decision-theoretic approach is to choose $\boldsymbol{\theta}$ giving rise to the lowest *mean-squared error (MSE)* under the empirical measure

$$R_N(\boldsymbol{\theta}) := \tfrac{1}{N} \textstyle\sum_{i=1}^N \|\mathbf{y}_i - h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}_i))\|_2^2 \tag{6.3}$$

where $\|\cdot\|_2$ is the Euclidean norm on $\mathbb{R}^m$. The expression (6.3) is known as the *empirical risk*. Its minimization lies at the heart of the *empirical risk minimization* principle. A closely related measure of the fit quality is the *population risk*

$$R(\boldsymbol{\theta}) := \mathbb{E}_{\mathbf{p},\mathbf{y}}\big[\|\mathbf{y} - h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}))\|_2^2\big] \tag{6.4}$$

Lemma 6.1. *Let $\mathcal{H}$ be a Hilbert space with inner product $\langle\cdot,\cdot\rangle$, norm $\|\cdot\|_2$, and $\mathbb{P}$ the distribution over the predictor X. For any random variables $Y \in \mathcal{H}$ and $f(X) \in \mathcal{H}$ that is $L^2(\mathbb{P})$ and $\mathbb{P}$-measurable, the MSE is $\mathbb{E}\big[\|Y - f(X)\|_2^2\big] = \mathbb{E}\big[\|Y - \mathbb{E}[Y|X]\|_2^2\big] + \mathbb{E}\big[\|\mathbb{E}[Y|X] - f(X)\|_2^2\big]$.*

Proof. Let $g(X)$ be a $\mathbb{P}$-measurable random variable. Hence,

$$\mathbb{E}[\langle \mathbb{E}[Y|X], g(X)\rangle] = \mathbb{E}[\mathbb{E}[\langle Y, g(X)\rangle|X]] = \mathbb{E}[\langle Y, g(X)\rangle]$$

The random variable $Y - \mathbb{E}[Y|X]$ is orthogonal to the space of L^2 and $\mathbb{P}$-measurable random variables, which includes $\mathbb{E}[Y|X] - f(X)$. Then, from the Pythagorean theorem, $\|Y - f(X)\|_2^2 = \|Y - \mathbb{E}[Y|X] + \mathbb{E}[Y|X] - f(X)\|_2^2 = \|Y - \mathbb{E}[Y|X]\|_2^2 + \|\mathbb{E}[Y|X] - f(X)\|_2^2$. □

We note that the conditional expectation of $\mathbf{y}$ given $\mathbf{p}$ is $\mathbb{E}[\mathbf{y}|\mathbf{p}] = h(\mathbf{x}^\star_{\boldsymbol{\theta}^\star}(\mathbf{p}))$ from assumption (6.2). If we define the following quantity

$$\rho(\boldsymbol{\theta}, \boldsymbol{\theta}^\star) := \mathbb{E}_{\mathbf{p}}\big[\|h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p})) - h(\mathbf{x}^\star_{\boldsymbol{\theta}^\star}(\mathbf{p}))\|_2^2\big]^{\frac{1}{2}} \tag{6.5}$$

then the identity in Lemma 6.1 becomes

$$R(\boldsymbol{\theta}) = R(\boldsymbol{\theta}^\star) + \rho^2(\boldsymbol{\theta}, \boldsymbol{\theta}^\star) \tag{6.6}$$

Hence, the parameter $\boldsymbol{\theta}$ minimizing criterion (6.4) is $\boldsymbol{\theta}^\star$, setting $\rho(\boldsymbol{\theta}, \boldsymbol{\theta}^\star)$ to zero, and setting the implicit function to $h(\mathbf{x}^\star_{\boldsymbol{\theta}^\star}(\cdot))$, *i.e.* it is the *Bayes' least-squares*, or the conditional expectation of $\mathbf{y}$ given $\mathbf{p}$. Since $R(\boldsymbol{\theta}^\star)$ is fixed, it is thus natural to measure the quality of an estimate $\boldsymbol{\theta}$ in terms of $\rho(\boldsymbol{\theta}, \boldsymbol{\theta}^\star)$ defined in (6.5). The quantity $\rho(\boldsymbol{\theta}, \boldsymbol{\theta}^\star)$ is also called the *expected prediction error*. Having good generalization guarantees for a trained model is extremely important in practice because it gives us a measure of the quality of the ultimately chosen model, see [78, Chap. 7]. From Lemma 6.1, the square of the prediction error $\rho(\boldsymbol{\theta}, \boldsymbol{\theta}^\star)$ is also equal to the *excess risk* $R(\boldsymbol{\theta}) - R(\boldsymbol{\theta}^\star)$.

Problem statement

Given a fixed collection of N samples $\{(\mathbf{p}_1, \mathbf{y}_1), (\mathbf{p}_2, \mathbf{y}_2), \cdots, (\mathbf{p}_N, \mathbf{y}_N)\}$, and a least-squares estimate $\hat{\boldsymbol{\theta}} \in \arg\min_{\boldsymbol{\theta} \in \Theta} R_N(\boldsymbol{\theta})$, we first study tail bounds on the empirical analogue of (6.5)

$$\rho_N(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star) := \left[\tfrac{1}{N} \sum_{i=1}^N \|h(\mathbf{x}^\star_{\hat{\boldsymbol{\theta}}}(\mathbf{p}_i)) - h(\mathbf{x}^\star_{\boldsymbol{\theta}^\star}(\mathbf{p}_i))\|_2^2\right]^{\frac{1}{2}} \tag{6.7}$$

We derive tail bounds on $\rho_N(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)$ which intuitively depends on a notion of the complexity of the *implicit function class*

$$\mathcal{H} := \{\mathbf{p} \in \mathcal{P} \mapsto h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p})) \,|\, \boldsymbol{\theta} \in \Theta\} \tag{6.8}$$

Studying the behavior of $\rho_N(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)$ gives information on the *expected prediction error* $\rho(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)$. Specifically, we want to know if $\rho_N(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)^2$ approaches $\rho(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)^2$ as the number N of observations increases. In other words, we want to know if the prediction error under the empirical measure agrees with its population average. In general, $\hat{\boldsymbol{\theta}}$ depends on the samples, hence it is random, and controlling the deviation $|\rho_N(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)^2 - \rho(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)^2|$ requires strong result, such as the uniform bound $\sup_{\boldsymbol{\theta} \in \Theta} |\rho_N(\boldsymbol{\theta}, \boldsymbol{\theta}^\star)^2 - \rho(\boldsymbol{\theta}, \boldsymbol{\theta}^\star)^2|$. We derive tail bounds on the uniform deviation, which depends on the complexity of the *loss class*

$$\mathcal{L} := \{\mathbf{p} \in \mathcal{P} \mapsto \|h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p})) - h(\mathbf{x}^\star_{\boldsymbol{\theta}^\star}(\mathbf{p}))\|_2^2\} \tag{6.9}$$

Denoting $\mathbb{P}_N$ the empirical distribution assigning mass $\frac{1}{N}$ to each of $\{\mathbf{p}_i\}_{i \in [N]}$, and the uniform bound

$$\|\mathbb{P} - \mathbb{P}_N\|_{\mathcal{L}} := \sup_{\boldsymbol{\theta} \in \Theta} |\rho(\boldsymbol{\theta}, \boldsymbol{\theta}^\star)^2 - \rho_N(\boldsymbol{\theta}, \boldsymbol{\theta}^\star)^2| \tag{6.10}$$

we can combine tail bounds on $\rho_N(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)$ and $\|\mathbb{P} - \mathbb{P}_N\|_{\mathcal{L}}$ to control the distribution of the prediction error $\rho(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)$ since we have the upper bound

$$\rho(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star) \leq \|\mathbb{P} - \mathbb{P}_N\|_{\mathcal{L}} + \rho_N(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star) \tag{6.11}$$

Motivation

The problem of estimating the convex objective f in the COP (6.1) is a very practical problem which emerges in many fields. For example, in control theory, we may seek to fit a lower complexity controller (thus easier to automate) from observing outputs of a sophisticated one available through, *e.g.* a human expert or model predictive control [164, 94]. In economics, consumer's purchases are modeled in order to maximize a utility function representing the satisfaction from one's purchases. This function is in general unknown to both the economist and the consumer, but can be learned by observing consumer purchases in response to price changes [94]. In transportation [131], routing games study drivers' routing decisions in a network in which traveling each edge incurs a cost. Estimating the edge cost functions is challenging since they may represent some combination of the travel time, the tolls, and other factors, which are not directly observable. In practice, it is often possible to observe the equilibrium flows induced by the selfish routing of agents through the sensing infrastructure, and to learn the underlying cost functions [19, 150]. In general, numerous processes involve agents that behave optimally with respect to utility functions, and [85], [19] use the COP framework to learn the utility functions in *Nash equilibrium problems*.

Modeling real-world processes as lower complexity COPs is common, as it enables to leverage powerful mathematical tools for the study of such processes. In economics, knowing the consumer utility function enables one to adjust prices to achieve some demand level [94]. In numerous cases in control, a low complexity controller requires less computation for little performance loss [94, 163]. In transportation, the selfish behavior of agents (from shorted path routing) leads to an aggregate cost in the network worse than the system's optimum, and which can be analytically quantified [137, 46]. Taxation schemes can be designed to incentivize system optimal drivers' decisions [65, 91].

However, low complexity models rely upon having an accurate approximation of the real ones. For example, system mischaracterizations in selfish routing can cause taxes designed for one problem instance to incentivize inefficient behavior on different, yet closely-related instances [30]. Hence, we want to be able to measure the quality of the learned model. In the present chapter, we present a statistical framework for the fitting of equilibrium models using the standard *empirical risk minimization* principle. For the class of implicit models (6.2), it is then critical to be able to have theoretical guarantees on the quality of the fit, as the number N of observations grows.

Related work

In order to obtain tail bounds on the empirical prediction error $\rho_N(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)$, we follow the work of [70, 98, 97, 11], and use characterization of the implicit class $\mathcal{H}$ given in (6.8) in terms of its *Gaussian complexity*, and its localized variants. We also extend the theory from real-valued functions to functions taking values in $\mathbb{R}^m$, by adapting results on additive regression models [145, 77, 119].

To obtain tail bounds on $\|\mathbb{P} - \mathbb{P}_N\|_{\mathcal{L}}$, we study convergence properties in the uniform norm, known as *Glivenko-Cantelli properties* [156], [157]. We leverage results relating the convergence properties of $\|\mathbb{P} - \mathbb{P}_N\|_{\mathcal{L}}$ with the *Rademacher complexity* of the loss class $\mathcal{L}$ given in (6.9), by following the approach of [12], [11], [99]. Then we use approximation theory [54, 132, 37] to bound the Rademacher complexity of $\mathcal{L}$ by a function of its *metric entropy* [96].

In contrast to classic prediction models, $\mathbf{x}^\star_{\boldsymbol{\theta}}(\cdot)$ is from an implicit class since it is defined as solutions to COPs. To characterize the complexity of the classes $\mathcal{H}$ and $\mathcal{L}$ given in (6.8) and (6.9), we translate variations in the parameter space Θ into variations in the classes $\mathcal{H}$ and $\mathcal{L}$. This requires sensitivity results in optimization theory [48], [174]. These types of learning problems, known as *inverse optimization*, were addressed in [85], [94], [19], [150], but with little to no analysis on their learnability.

Contributions and outline

In Section 6.2, we provide three fully-developed applications to the problem of learning a convex objective to motivate our study. We then present results on the Lipschitz properties of solutions to convex programs in Section 6.3. In Section 6.4, we provide tail bounds on the prediction error $\rho_N(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)$ and on the uniform deviation $\|\mathbb{P} - \mathbb{P}_N\|_{\mathcal{L}}$ in terms of the Gaussian and Rademacher complexities. In Section 6.5, we bound the Rademacher and Gaussian complexities by the metric entropy, which enables us to derive our final tail bounds in Section 6.6.

6.2 Applications

Routing games

Setting. Routing games go back to the 1950s [166], and are extensively studied in transportation [131]. We consider a non-cooperative game on a network represented by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ equipped with continuous, non-decreasing congestion functions $c_e(\cdot) : \mathbb{R}_+ \to \mathbb{R}_{>0}$ for each $e \in \mathcal{E}$. The set of players is partitioned in *populations* $\{\mathcal{X}_k\}_{k\in[K]}$. For each $k \in [K]$, players in $\mathcal{X}_k$ have available a set of simple paths $\mathcal{P}_k$ from a common source $s_k \in \mathcal{V}$ to a common sink $t_k \in \mathcal{V}$. For each population $\mathcal{X}_k$, we define $d_k \in \mathbb{R}_+$ its total flow, and $\boldsymbol{\mu}^k = (\mu^k_p)_{p\in\mathcal{P}_k} \in \mathbb{R}_+^{\mathcal{P}_k}$ its *path assignment*, which satisfies $\sum_{p\in\mathcal{P}_k} \mu^k_p = d_k$. We denote $\mathcal{P}$ the disjoint union $\mathcal{P} = \sqcup_{k=1}^K \mathcal{P}_k$, thus $\mathbb{R}_+^{\mathcal{P}} = \Pi_{k=1}^K \mathbb{R}_+^{\mathcal{P}_k}$. Under population demand $\mathbf{d} = (d_k)_{k\in[K]}$, the path assignment can be summarized by $\boldsymbol{\mu} = (\boldsymbol{\mu}^1, \cdots, \boldsymbol{\mu}^K)$ in the feasible set: $\Delta(\mathbf{d}) := \left\{\boldsymbol{\mu} \in \mathbb{R}_+^{\mathcal{P}} : \sum_{p\in\mathcal{P}_k} \mu^k_p = d_k, \, \forall\, k \in [K]\right\}$. The path assignment determines the *edge flow* defined as $x_e = \sum_{k=1}^K \sum_{p\in\mathcal{P}_k : e\in p} \mu^k_p$, which can be written compactly as $x_e = (\mathbf{M}\boldsymbol{\mu})_e$ where $\mathbf{M} \in \mathbb{R}^{\mathcal{E}\times\mathcal{P}}$ is an incidence matrix with entries defined as $M_{e,p} = \mathbf{1}_{e\in p}$. For each edge e,

the edge flow incurs a cost $c_e(x_e)$, and the cost of choosing a path p is the sum of edge costs along the path, *i.e.* $\sum_{e\in p} c_e(x_e)$.

Equilibrium in routing games: Let us define the function $f : \mathbb{R}_+^{\mathcal{E}} \to \mathbb{R}$ such that $f(\mathbf{x}) = \sum_{e\in\mathcal{E}} \int_0^{x_e} c_e(u)du$ with $\mathbf{x} = (x_e)_{e\in\mathcal{E}} \in \mathbb{R}^{|\mathcal{E}|}$, and the set $\mathcal{D}(\mathbf{d}) = \mathbf{M}\Delta(\mathbf{d}) = \{\mathbf{M}\boldsymbol{\mu} : \boldsymbol{\mu} \in \Delta(\mathbf{d})\} \subset \mathbb{R}_+^{\mathcal{E}}$ of feasible edge flows. We say that $\mathbf{x}^\star \in \mathcal{D}(\mathbf{d})$ is a Nash equilibrium if it is an optimal solution of the COP $\min_{\mathbf{x}\in\mathcal{D}(\mathbf{d})} f(\mathbf{x})$, see [131]. Note that f is convex since each c_e is non-decreasing by assumption, and $\mathcal{D}(\mathbf{d})$ is also compact convex since it is the image of $\mathbf{M}$ restricted to the compact convex set $\Delta(\mathbf{d})$. In this application, the demand vector $\mathbf{d}$ is the predictor, which we assume lies within a compact $\mathcal{P}$.

Learning the congestion functions: We measure the edge flows on a subset $\mathcal{A} \subseteq \mathcal{E}$ of the edges, *i.e.* the observation mapping is the projection of $\mathbb{R}_+^{\mathcal{E}}$ into $\mathbb{R}_+^{\mathcal{A}}$ defined by $h : \mathbf{x} \mapsto (x_e)_{e\in\mathcal{A}}$. It remains to define the indexed family $\{f_{\boldsymbol{\theta}} \,|\, \boldsymbol{\theta} \in \Theta\}$ of objectives to be estimated. Following standard approach in traffic modeling [31], [29], and in inverse modeling [19], [150], we assume available the capacity m_e and the base cost c_e^0 of each edge $e \in \mathcal{E}$, and define the class of univariate functions $\Theta := \{\theta : [0,1] \to \mathbb{R}_+, L\text{-Lipschitz}, c\text{-strong-monotone}\}$. Then for each $\theta \in \Theta$ and $e \in \mathcal{E}$, the cost functions are given by $c_{\theta,e}(x_e) = c_e^0 + \theta(\frac{x_e}{m_e})$, *i.e.* the functions are invariant with respect to the normalized edge flows x_e/m_e on each edge e. Then $f_\theta(\mathbf{x}) = \sum_{e\in\mathcal{E}} \int_0^{x_e} c_{\theta,e}(u)du$.

Usage. Learning the edge cost functions $c_{\theta,e}$ is used to quantify the inefficiency of equilibria in routing games [137, 46], and to design taxation schemes to incentivize system optimal decisions [65, 91], hence having an accurate estimate θ is critical.

Consumer utility

Setting. We consider n products indexed by $i \in [n]$, with prices $\mathbf{p} = (p_i)_{i\in[n]} \in [0, p_{\max}]^n$ (where $p_{\max} \in \mathbb{R}_{>0}$ is the maximum price) and demand $\mathbf{x} = (x_i)_{i\in[n]}$. Consumer purchases are assumed to solve the COP: $\min_{\mathbf{x}\in\mathbb{R}_+^n} \mathbf{p}^T\mathbf{x} - u(\mathbf{x})$, where $u : \mathbb{R}_+^n \to \mathbb{R}$ is a concave and non-decreasing utility function modeling the consumer's satisfaction from its purchases. With $\mathcal{S}_-^n$ the set of negative semi-definite matrices of $\mathbb{R}^{n\times n}$, we learn u in the class $\mathcal{F} := \{\mathbf{x} \in \mathbb{R}_+^n \mapsto \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{r}^T\mathbf{x} \,|\, (\mathbf{Q}, \mathbf{r}) \in \Theta\}$ parametrized over $\Theta := \{(\mathbf{Q}, \mathbf{r}) \in \mathcal{S}_-^n \times \mathbb{R}_+^n \,|\, \mathbf{Q}\,\mathbf{x}_{\max} + \mathbf{r} \geq 0,\, cI \preceq -\mathbf{Q} \preceq LI,\, \|\mathbf{r}\|_2 \leq r_{\max}\}$, where $\mathbf{x}_{\max} \in \mathbb{R}_+^n$ is the maximum demand vector. Hence, by construction of Θ, we assume that the utility function is concave quadratic and is increasing, *i.e.* $\nabla u(\mathbf{x}) = \mathbf{Q}\mathbf{x} + \mathbf{r} \geq 0$, for all $\mathbf{x} \in \mathbb{R}_+^n$ such that $\mathbf{x} \leq \mathbf{x}_{\max}$. Note that in this application, the objective function depends on the random predictors $\mathbf{p}$ (the prices), while the domain $\mathbb{R}_+^n$ is independent from $\mathbf{p}$. For each estimate $(\hat{\mathbf{Q}}, \hat{\mathbf{r}}) \in \Theta$, or corresponding $\hat{u} \in \mathcal{F}$, we want to say something about the quality of the fit.

Usage. The estimate $\hat{u}$ can be used to set prices $\mathbf{p}$ to achieve a target demand level $\mathbf{x}$, see [94]. Hence, having theoretical guarantees on the quality of the learned model is of great importance.

Controller fitting

Setting. We consider a dynamical system with state $\mathbf{x}_t \in \mathbb{R}^n$, input $\mathbf{u}_t \in \mathbb{R}^m$, and i.i.d. noise $\mathbf{w}_t \in \mathbb{R}^n$ at time t. The linear dynamics are $\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t$, $t \geq 0$. Given a convex stage cost function $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, the stochastic control problem consists in finding a control policy $\{\mathbf{u}_t\}_{t\geq 0}$ that minimizes $\limsup_{T\to\infty} \frac{1}{T}\mathbb{E}\sum_{t=1}^{T-1} \ell(\mathbf{x}_t, \mathbf{u}_t)$ with the constraint $\mathbf{F}\,\mathbf{u}_t \leq \mathbf{h}$, $t \geq 0$, where $\mathbf{F} \in \mathbb{R}^{p\times m}$ and $\mathbf{h} \in \mathbb{R}^p$. We refer to [18, 94] for a full technical discussion on stochastic control.

Learning an approximate control. We are given samples of state-control (or input-output) pairs $\{(\mathbf{x}_i, \mathbf{u}_i)\}_{i\in[N]}$ from a suboptimal (but complex) control policy run by a human expert or a computationally expensive controller such as model predictive control [18, 164], and we want to learn a global *approximate value function* $v : \mathbb{R}^n \to \mathbb{R}$ that gives us a lower complexity controller via the optimization program: $\min_{\mathbf{u}\,:\,\mathbf{F}\,\mathbf{u}\leq\mathbf{h}} \ell(\mathbf{x}, \mathbf{u}) + v(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u})$. This control policy is known as the *approximate dynamic programming* policy [18] and a standard approach [94] is to learn v in the class $\mathcal{F} := \{\mathbf{z} \mapsto \mathbf{z}^T\mathbf{P}\mathbf{z} \,|\, \mathbf{P} \in \Theta\}$, where the index set Θ is $\Theta := \{\mathbf{P} \in \mathcal{S}^n_+ \,:\, cI \preceq \mathbf{P} \preceq LI\}$.

Usage. We can use the program $\min_{\mathbf{u}\,:\,\mathbf{F}\,\mathbf{u}\leq\mathbf{h}} \ell(\mathbf{x}, \mathbf{u}) + v(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u})$ with a value function estimate $\hat{v}$ to approximate a policy with a computationally efficient controller, see [94].

6.3 Lipschitz properties of convex optimization programs

The distribution of $\rho_N(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)$ given in (6.7), where $\hat{\boldsymbol{\theta}}$ is the least-squares estimator minimizing $R_N(\boldsymbol{\theta})$ given in (6.3), should intuitively depend on the complexity of the implicit class $\mathcal{H}$ given in (6.8). Similarly, the distribution of the uniform deviation $\|\mathbb{P} - \mathbb{P}_N\|_{\mathcal{L}} = \sup_{\boldsymbol{\theta}\in\Theta} |\rho(\boldsymbol{\theta}, \boldsymbol{\theta}^\star)^2 - \rho_N(\boldsymbol{\theta}, \boldsymbol{\theta}^\star)^2|$, which is random because it is a function of the collection $\{\mathbf{p}_i\}_{i\in[N]}$ of i.i.d. random variables sampled from $\mathbb{P}$, should depend on the complexity of the loss class $\mathcal{L}$ given in (6.9). At the heart of $\mathcal{H}$ and $\mathcal{L}$, lies the implicit function $\mathbf{p} \mapsto \mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p})$, mapping from the predictor space $\mathcal{P}$ to the state space $\mathbb{R}^n$, and defined as the solution to the $\mathrm{COP}(f_{\boldsymbol{\theta}}(\cdot, \mathbf{p}), \mathcal{D}(\mathbf{p}))$. Since adding a constant to $f_{\boldsymbol{\theta}}$ will not affect the optimal solution, we study how the smoothness properties of $\mathbf{x}^\star_{\boldsymbol{\theta}}(\cdot)$ with respect to $\boldsymbol{\theta}$ and $\mathbf{p}$ result from the smoothness properties of the objective gradient $\nabla_{\mathbf{x}} f_{\boldsymbol{\theta}}$. This allows us to characterize the complexity of $\mathcal{H}$ and $\mathcal{L}$.

From convexity of $f_{\boldsymbol{\theta}}$ (by assumption), the first-order optimality condition for the COP $(f_{\boldsymbol{\theta}}(\cdot, \mathbf{p}), \mathcal{D}(\mathbf{p}))$ states that $\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p})$ is equivalently the solution to the following Variational Inequality Problem $\mathrm{VIP}(\nabla_{\mathbf{x}} f_{\boldsymbol{\theta}}(\cdot, \mathbf{p}), \mathcal{D}(\mathbf{p}))$, $(\boldsymbol{\theta}, \mathbf{p}) \in \Theta \times \mathcal{P}$ (see [27, §4.2.3])

$$\text{find } \mathbf{x} \in \mathcal{D}(\mathbf{p}) \quad \text{s.t.} \quad \langle \nabla_{\mathbf{x}} f_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p}),\, \mathbf{x}' - \mathbf{x}\rangle \geq 0,\ \forall\, \mathbf{x}' \in \mathcal{D}(\mathbf{p}) \tag{6.12}$$

Since the objectives in $\{f_{\boldsymbol{\theta}}(\cdot, \mathbf{p}) \,|\, (\boldsymbol{\theta}, \mathbf{p}) \in \Theta \times \mathcal{P}\}$ are c-strongly convex and L-Lipschitz gradient, which is equivalent to $\nabla_{\mathbf{x}} f_{\boldsymbol{\theta}}(\cdot, \mathbf{p})$ being L-Lipschitz and c-strongly monotone,[1]

[1] $F : \mathbb{R}^n \to \mathbb{R}^n$ is c-strongly monotone if for each $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$, $\langle \nabla f_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla f_{\boldsymbol{\theta}}(\mathbf{x}'), \mathbf{x} - \mathbf{x}'\rangle \geq c\,\|\mathbf{x} - \mathbf{x}'\|_2^2$

see [61]. In addition, we suppose there exists a convex compact subset $\mathcal{D}$ of $\mathbb{R}^n$ such that $\mathcal{D}(\mathbf{p}) \subset \mathcal{D}$ for all $\mathbf{p} \in \mathcal{P}$ (such is the case in the applications presented in Section 6.2). Let $|||F||| = \sup_{\mathbf{x}\in\mathcal{D}(\mathbf{p})} \|F(\mathbf{x})\|_2$ be a norm over the set of maps $F : \mathcal{D} \to \mathbb{R}^n$, $\|\cdot\|_\Theta$ a norm over the index set Θ, and $\|\cdot\|_\mathcal{P}$ a norm over the predictor space $\mathcal{P}$.

Definition 6.1. *Given a convex compact subset $\mathcal{D}$ of $\mathbb{R}^n$, we say that a map $F(\cdot, \boldsymbol{\theta}, \boldsymbol{p}) : \mathcal{D} \to \mathbb{R}^n$ is smoothly parametrized with respect to $\boldsymbol{\theta} \in \Theta$ and $\boldsymbol{p} \in \mathcal{P}$ if there exist constants L_Θ and $L_\mathcal{P}$ such that for each $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$ and for each $\boldsymbol{p}, \boldsymbol{p}' \in \mathcal{P}$, we have $|||F(\cdot, \boldsymbol{\theta}, \boldsymbol{p}) - F(\cdot, \boldsymbol{\theta}', \boldsymbol{p}')||| \leq L_\Theta\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\Theta + L_\mathcal{P}\|\boldsymbol{p} - \boldsymbol{p}'\|_\mathcal{P}$.*

For instance, the objective function in the routing games framework presented in Section 6.2 has gradient $\nabla_\mathbf{x} f_\theta(\mathbf{x}) = (c_{\theta,e}(x_e))_{e\in\mathcal{E}}$ and is smoothly parametrized with constants $L_\Theta = \sqrt{|\mathcal{E}|}$ and $L_\mathcal{P} = 0$, where Θ is the function class $\Theta = \{\theta : [0,1] \to \mathbb{R}_+, L\text{-Lipschitz}, c\text{-strong-monotone}\}$ equipped with infinity norm $\|\theta\|_\infty = \sup_{t\in[0,1]} |\theta(t)|$. For the application to consumer utility (see Section 6.2), the index is $\boldsymbol{\theta} = (\mathbf{Q}, \mathbf{r})$, the index set is $\Theta = \mathcal{S}^n_- \times \mathbb{R}^n_+$, the predictor space is the hypercube $[0, p_{\max}]^n$, and the objective gradient is $\nabla_\mathbf{x} f_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p}) = \mathbf{p} - \mathbf{r} - \mathbf{Q}\mathbf{x}$. With $\|\cdot\|_\mathcal{P} = \|\cdot\|_2$, $\|\mathbf{Q}\|_{\text{op}} = \sup_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{Q}\mathbf{x}\|_2$, and $\|(\mathbf{Q}, \mathbf{r})\|_\Theta = \|\mathbf{Q}\|_{\text{op}} + \|\mathbf{r}\|_2$, the Lipschitz constants are $L_\mathcal{P} = 1$ and $L_\Theta = \max(1, \|\mathbf{x}_{\max}\|_2)$.

We now provide the main result of the Section. We show that, for a "smoothly parametrized" VIP, the solution to it is also smoothly parametrized.

Theorem 6.1. *(Smooth VIP) For any parametric VIP$(F_{\boldsymbol{\theta}}(\cdot, \boldsymbol{p}), \mathcal{D}(\boldsymbol{p}))$, $(\boldsymbol{\theta}, \boldsymbol{p}) \in \Theta \times \mathcal{P}$, such that*

(a) $\mathcal{D}(\boldsymbol{p})$ is a compact convex subset of $\mathbb{R}^n$ for all $\boldsymbol{p} \in \mathcal{P}$

(b) $\|P_{\mathcal{D}(\boldsymbol{p})}(\boldsymbol{x}) - P_{\mathcal{D}(\boldsymbol{p}')}(\boldsymbol{x})\| \leq \tilde{L}_\mathcal{P}\|\boldsymbol{p} - \boldsymbol{p}'\|_\mathcal{P}$ for all $\boldsymbol{p}, \boldsymbol{p}' \in \mathcal{P}$, $\boldsymbol{x} \in \mathbb{R}^n$

(c) $F_{\boldsymbol{\theta}}$ is c-strong-monotone and L-Lipschitz for all $\boldsymbol{\theta} \in \Theta$

(d) $|||F_{\boldsymbol{\theta}}(\cdot, \boldsymbol{p}) - F_{\boldsymbol{\theta}'}(\cdot, \boldsymbol{p}')||| \leq L_\Theta\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\Theta + L_\mathcal{P}\|\boldsymbol{p} - \boldsymbol{p}'\|_\mathcal{P}$ for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$ and $\boldsymbol{p}, \boldsymbol{p}' \in \mathcal{P}$

the unique solution $\boldsymbol{x}^\star_{\boldsymbol{\theta}}(\boldsymbol{p})$ to VIP$(F_{\boldsymbol{\theta}}(\cdot, \boldsymbol{p}), \mathcal{D}(\boldsymbol{p}))$ satisfies, for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$ and for all $\boldsymbol{p}, \boldsymbol{p}' \in \mathcal{P}$

$$\|\boldsymbol{x}^\star_{\boldsymbol{\theta}'}(\boldsymbol{p}') - \boldsymbol{x}^\star_{\boldsymbol{\theta}}(\boldsymbol{p})\| \leq \frac{(cL_\mathcal{P}/L^2 + \tilde{L}_\mathcal{P})\|\boldsymbol{p} - \boldsymbol{p}'\|_\mathcal{P} + (cL_\Theta/L^2)\,\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\Theta}{1 - \sqrt{1 - c^2/L^2}}$$

Proof. Since the unique solution $\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p})$ of VIP$(F_{\boldsymbol{\theta}}, \mathcal{D}(\mathbf{p}))$ can be equivalently characterized as the unique solution of the fixed point problem $\mathbf{x} = P_{\mathcal{D}(\mathbf{p})}(\mathbf{x} - \frac{c}{L^2} F_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p}))$, we define, for all $(\boldsymbol{\theta}, \mathbf{p}, \mathbf{q}, \mathbf{x}) \in \Theta \times \mathcal{P} \times \mathcal{P} \times \mathbb{R}^n$, the function $Q_{\boldsymbol{\theta},\mathbf{p},\mathbf{q}}(\mathbf{x}) := P_{\mathcal{D}(\mathbf{q})}(\mathbf{x} - \frac{c}{L^2} F_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p}))$. Then for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$, and $\mathbf{p} \in \mathcal{P}$

$$\|\mathbf{x}^\star_{\boldsymbol{\theta}'}(\mathbf{p}') - \mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p})\| = \|Q_{\boldsymbol{\theta}',\mathbf{p}'}(\mathbf{x}^\star_{\boldsymbol{\theta}'}(\mathbf{p}')) - Q_{\boldsymbol{\theta},\mathbf{p}}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}))\| \leq T_1 + T_2 + T_3$$

where

$$\begin{aligned} T_1 &:= \|Q_{\boldsymbol{\theta}',\mathbf{p}',\mathbf{p}'}(\mathbf{x}^\star_{\boldsymbol{\theta}'}(\mathbf{p}')) - Q_{\boldsymbol{\theta}',\mathbf{p}',\mathbf{p}'}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}))\| \\ T_2 &:= \|Q_{\boldsymbol{\theta}',\mathbf{p}',\mathbf{p}'}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p})) - Q_{\boldsymbol{\theta},\mathbf{p},\mathbf{p}'}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}))\| \\ T_3 &:= \|Q_{\boldsymbol{\theta},\mathbf{p},\mathbf{p}'}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p})) - Q_{\boldsymbol{\theta},\mathbf{p},\mathbf{p}}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}))\| \end{aligned}$$

From Lemma 1, T_1 is less than $\sqrt{1-\frac{c^2}{L^2}}\,\|\mathbf{x}^\star_{\boldsymbol{\theta}'}(\mathbf{p}') - \mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p})\|_2$. Note that $1-\frac{c^2}{L^2} \in [0,1)$. And $P_{\mathcal{D}(\mathbf{p}')}$ being 1-Lipschitz, T_2 is upper bounded by

$$\begin{aligned} T_2 &\leq \|\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}) - \tfrac{c}{L^2}F_{\boldsymbol{\theta}}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}),\mathbf{p}) - (\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}) - \tfrac{c}{L^2}F_{\boldsymbol{\theta}'}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}),\mathbf{p}'))\|_2 \\ &= \tfrac{c}{L^2}\|F_{\boldsymbol{\theta}}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}),\mathbf{p}) - F_{\boldsymbol{\theta}'}(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}),\mathbf{p}')\|_2 \\ &\leq \tfrac{c\,L_\Theta}{L^2}\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|_\Theta + \tfrac{c\,L_\mathcal{P}}{L^2}\|\mathbf{p}-\mathbf{p}'\|_\mathcal{P} \end{aligned}$$

By assumption (b) we have that $T_3 \leq \tilde{L}_\mathcal{P}\|\mathbf{p}-\mathbf{p}'\|_\mathcal{D}$. Putting together the three bounds and re-arranging the terms proves our claim. □

We note that assumption (b) in Theorem 6.1 arises naturally in many applications. For example, the routing games framework presented in Section 6.2 has a parametric feasible set $\mathcal{D}(\mathbf{p}) = \mathbf{M}\Delta(\mathbf{p})$, see [174]. In general, any polyhedral feasible set with a parametric right-hand side, *i.e.* of the form $\mathcal{D}(\mathbf{p}) = \{\mathbf{x} \in \mathbb{R}^n \,|\, \mathbf{A}\mathbf{x} = \mathbf{p}\}$ has a smoothly parametrized Euclidean projection, see [174].

6.4 Tail bounds using Rademacher and Gaussian complexities

In this section, we first present general results which are instrumental in obtaining tail bounds on the prediction error $\rho_N(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)$ defined in (6.7), which we recall is

$$\rho_N(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star) := \left[\tfrac{1}{N}\textstyle\sum_{i=1}^N \|h(\mathbf{x}^\star_{\hat{\boldsymbol{\theta}}}(\mathbf{p}_i)) - h(\mathbf{x}^\star_{\boldsymbol{\theta}^\star}(\mathbf{p}_i))\|_2^2\right]^{\frac{1}{2}}$$

where $\hat{\boldsymbol{\theta}}$ is a least-squares estimator with respect to the empirical MSE $R_N(\boldsymbol{\theta})$ defined in (6.3)

$$R_N(\boldsymbol{\theta}) := \tfrac{1}{N}\textstyle\sum_{i=1}^N \|\mathbf{y}_i - h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}_i))\|_2^2$$

The general learning problem consists in estimating a function $f : \mathcal{X} \to \mathbb{R}^m$ from N predictors $\{\mathbf{x}_i\}_{i=1}^N$ from $\mathcal{X}$ and N responses $\{\mathbf{y}_i\}_{i=1}^N$ from $\mathbb{R}^m$ with relationship

$$\mathbf{y}_i = f^\star(\mathbf{x}_i) + \boldsymbol{\epsilon}_i, \quad i \in [N] \tag{6.13}$$

where $\boldsymbol{\epsilon}_i \in \mathbb{R}^m$ are i.i.d. noise vectors of independent Gaussian random variables sampled from $\mathcal{N}(0, \mathrm{diag}(\boldsymbol{\sigma}^2))$, with $\boldsymbol{\sigma}^2 := (\sigma_1^2, \cdots, \sigma_m^2)$ the vector of variances. With $\mathcal{F}$ a suitably chosen subset of $\mathbb{R}^m$-valued functions, we bound the prediction error $\|\hat{f} - f^\star\|_N := \left[\frac{1}{N}\sum_{i=1}^N \|\hat{f}(\mathbf{x}_i) - f^\star(\mathbf{x}_i)\|_2^2\right]^{\frac{1}{2}}$, where $\hat{f}$ is the least-squares estimator $\hat{f} \in \arg\min_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^N \|\mathbf{y}_i - f(\mathbf{x}_i)\|_2^2$. We have the following

Lemma 6.2. *With $\{w_{ij}\}_{i\in[N],j\in[m]}$ a collection of Nm i.i.d. samples drawn from $\mathcal{N}(0,1)$, the least-squares and Bayes' estimates $\hat{f}$ and $f^\star$ satisfy $\frac{1}{2}\|\hat{f} - f^\star\|_N^2 \leq \sum_{j=1}^m \frac{\sigma_j}{N}\sum_{i=1}^N (\hat{f}_j(\boldsymbol{x}_i) - f_j^\star(\boldsymbol{x}_i))\, w_{ij}$.*

Proof. By definition of the least squares

$$\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{y}_i - \hat{f}(\mathbf{x}_i))\|_2^2 \leq \frac{1}{N}\sum_{i=1}^{N}\|(\mathbf{y}_i - f^\star(\mathbf{x}_i))\|_2^2$$

Plugging $\mathbf{y}_i = f^\star(\mathbf{x}_i) + \boldsymbol{\epsilon}_i$, we get

$$\frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{m}(f_j^\star(\mathbf{x}_i)) + \sigma_j w_{ij} - \hat{f}_j(\mathbf{x}_i))^2 \leq \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{m}(\sigma_j w_{ij})^2$$

Developing the squares,

$$\frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{m}(\hat{f}_j(\mathbf{x}_i) - f_j^\star(\mathbf{x}_i))^2 + \sum_{j=1}^{m}\frac{\sigma_j}{N}\sum_{i=1}^{N} w_{ij}(f_j^\star(\mathbf{x}_i) - \hat{f}_j(\mathbf{x}_i) \leq 0$$

which proves the inequality. □

The above inequality is known as the *basic inequality for least squares*, which we extend to $\mathbb{R}^m$-valued functions. Focusing on the 1-dimensional case ($m = 1$), the basic inequality leads us to study *local Gaussian complexities*, which measures the complexity of a class $\mathcal{F}$ of real-valued functions in a neighborhood of the a regression function $f^\star \in \mathcal{F}$. Denoting the set $\mathcal{F}^\star := \{f - f^\star,\, f \in \mathcal{F}\}$,

Definition 6.2. *We call local Gaussian complexity of a real-valued function class $\mathcal{F}$ around $f^\star \in \mathcal{F}$ at scale δ the quantity $\mathcal{G}_N(\delta, \mathcal{F}^\star) := \mathbb{E}_w[\sup_{f\in\mathcal{F}^\star,\, \|f\|_N \leq \delta} |\frac{1}{N}\sum_{i=1}^{N} w_i f(\boldsymbol{x}_i)|]$ with $w_i \sim \mathcal{N}(0,1)$ i.i.d.*

The Gaussian complexity is the average of the maximum correlation between the vector $(f(\mathbf{x}_1), \cdots, f(\mathbf{x}_N))$ and the noise vector $(w_1, \cdots, w_N)$. Intuitively, as a function class grows, it is easier to find a function that correlates well with a randomly drawn noise vector, making the complexity grow. From the *basic inequality* in Lemma 6.2 applied to the case $m = 1$, the error $\delta := \mathbb{E}[\|\hat{f} - f^\star\|_N^2]^{\frac{1}{2}}$ should intuitively satisfy an inequality of the form $\frac{\delta^2}{2} \leq \sigma\, \mathcal{G}_N(\delta, \mathcal{F}^\star)$. If we can identify a minimal radius $\delta^\star$ such that $\frac{\delta^2}{2} \geq \sigma\, \mathcal{G}_N(\delta, \mathcal{F}^\star)$ for all $\delta > \delta^\star$, then we must have $\mathbb{E}[\|\hat{f} - f^\star\|_N^2]^{\frac{1}{2}} \leq \delta^\star$. Existence of $\delta^\star$ is guaranteed if $\mathcal{F}$ is star-shaped[2]

Lemma 6.3. *Let $\sigma \in \mathbb{R}_{>0}$. For a star-shaped class of real-valued functions $\mathcal{F}$, the function $\delta \mapsto \frac{\mathcal{G}_N(\delta,\mathcal{F})}{\delta}$ is non increasing on $\mathbb{R}_{>0}$. Thus $\sigma\, \mathcal{G}_N(\delta, \mathcal{F}) \leq \frac{\delta^2}{2}$ admits a minimal solution on $\mathbb{R}_{>0}$.*

Proof. Let $t \geq \delta$ fixed. From the star shaped condition, if $g \in \mathcal{F}$ then $\frac{\delta}{t}g \in \mathcal{F}$. Hence, $\frac{\delta}{t}\mathcal{G}_N(t, \mathcal{F}) = \mathbb{E}_w[\sup_{\|g\|_N \leq t} |\sum_{i=1}^{N} w_i \frac{\delta}{t} g(\mathbf{x}_i)|] \leq \mathbb{E}_w[\sup_{\|g\|_N \leq \delta} |\sum_{i=1}^{N} w_i g(\mathbf{x}_i)|] = \mathcal{G}_N(\delta, \mathcal{F})$, *i.e* $\frac{\mathcal{G}_n(t,\mathcal{F}^\star)}{t} \leq \frac{\mathcal{G}_n(\delta,\mathcal{F}^\star)}{\delta}$. Hence $\delta \mapsto \frac{\mathcal{G}_N(\delta,\mathcal{F})}{\delta}$ is non increasing and $\delta \mapsto \frac{\mathcal{G}_N(\delta,\mathcal{F})}{\delta} - \frac{\delta}{2\,\sigma}$ is decreasing, concluding our proof. □

[2]A function class $\mathcal{F}$ is star-shaped if for all $(h, \alpha) \in \mathcal{F} \times [0, 1]$, $\alpha h \in \mathcal{F}$

Using the star-shaped property for classes of real-valued functions enables us to obtain a tail bound on the prediction error. In order to prove Theorem 6.2, we first derive a tail bound on the local Gaussian complexity. Let us assume that the noise in (2) are Gaussian vectors $(\boldsymbol{\epsilon}_i = (\sigma_j w_{ij})_{j=1,\cdots,d})_{i=1,\cdots,N}$.

Lemma 6.4. *Let $\mathcal{F}$ be a star-shaped class of real-valued functions, let $\tilde{\delta} \in \mathbb{R}_{>0}$ be a positive solution to the inequality $\sigma\,\mathcal{G}_N(\delta, \mathcal{F}) \leq \frac{\delta^2}{2}$, and $w_i \sim \mathcal{N}(0,1)$ i.i.d. Then, for each $t \geq \max(\tilde{\delta}, \|f\|_N)$, with probability at least $1 - \exp\big(-\frac{2N\tilde{\delta}t}{\sigma^2}\big)$, we have $\frac{\sigma}{N}|\sum_{i=1}^N w_i f(\boldsymbol{x}_i)| \leq \frac{\tilde{\delta}t}{2} + 2\sqrt{\tilde{\delta}t}\|f\|_N$.*

Proof. Since the function $\mathbf{w} \mapsto \frac{\sigma}{N}|\sum_{i=1}^N w_i f(\mathbf{x}_i)|$ is $\frac{\sigma\|f\|_N}{\sqrt{N}}$-Lipschitz, we have from the concentration property of Lipschitz functions of i.i.d. Gaussian variables, for each $u \in \mathbb{R}_{>0}$: $\mathbb{P}\big[\frac{\sigma}{N}|\sum_{i=1}^N w_i f(\mathbf{x}_i)| \geq \frac{\sigma}{N}\mathbb{E}|\sum_{i=1}^N f(\mathbf{x}_i)w_i| + 2u\|f\|_N\big] \leq \exp\big(\frac{-4Nu^2\|f\|_N^2}{2\sigma^2\|f\|_N^2}\big) = \exp(\frac{-2Nu^2}{\sigma^2})$.
In addition, we have $\frac{\sigma}{N}\mathbb{E}|\sum_{i=1}^N f(\mathbf{x}_i)w_i| \leq \sigma\mathcal{G}_N(t,\mathcal{F}) = \frac{\sigma t\,\mathcal{G}_N(t,\mathcal{F})}{t} \leq \frac{\sigma t\,\mathcal{G}_N(\tilde{\delta},\mathcal{F})}{\tilde{\delta}} \leq \frac{t\tilde{\delta}}{2}$, where the first inequality is from $\|f\|_N \leq t$ and by definition of $\mathcal{G}_N(t,\mathcal{F})$, the second inequality is from Lemma 3 since $\mathcal{F}$ is star-shaped, and the third inequality stems from $\sigma\,\mathcal{G}_N(\tilde{\delta},\mathcal{F}) \leq \frac{\tilde{\delta}^2}{2}$. Thus, we have $\mathbb{P}\big[\frac{\sigma}{N}|\sum_{i=1}^N w_i f(\mathbf{x}_i)| \geq \frac{t\tilde{\delta}}{2} + 2u\|f\|_N\big] \leq \mathbb{P}\big[\frac{\sigma}{N}|\sum_{i=1}^N w_i f(\mathbf{x}_i)| \geq \frac{\sigma}{N}\mathbb{E}|\sum_{i=1}^N f(\mathbf{x}_i)w_i| + 2u\|f\|_N\big]$, and taking $u = \sqrt{t\tilde{\delta}}$ finishes the proof. □

Theorem 6.2. *Let $\mathcal{F}^\star$ be a $f^\star$-shifted function class of $\mathbb{R}^m$-valued functions. Assume that for each $\Delta := f - f^\star \in \mathcal{F}^\star$, each one of its components $\Delta_j := f_j - f_j^\star$, $j \in [m]$, belongs to a $f_j^\star$-shifted class of real-valued functions $\mathcal{F}_j^\star$ that is star-shaped. Let $\hat{f}$ be the least-squares estimate $\hat{f} \in \arg\min_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^N \|\boldsymbol{y}_i - f(\boldsymbol{x}_i)\|_2^2$, where $\boldsymbol{y}_i$ is given by (6.13). For each $j \in [m]$, let δ_j be the smallest positive solution to the inequality $\sigma_j\,\mathcal{G}_N(\delta, \mathcal{F}_j^\star) \leq \frac{\delta^2}{2}$. Let us define $\delta_{\max} := \max_{j\in[m]}\delta_j$, and $\|\hat{\Delta}\|_{\max} := \max_{j\in[m]}\|\hat{\Delta}_j\|_N$, where $\hat{\Delta} = \hat{f} - f^\star$. Then, for each $t \geq \max(\delta_{\max}, \|\hat{\Delta}\|_{\max})$, with probability at least $1 - \sum_{j=1}^m \exp\big(-\frac{2N\delta_j t}{\sigma_j^2}\big)$, we have $\|\hat{f} - f^\star\|_N^2 \leq m\,t\,\delta_{\max}(2+\sqrt{5})^2$.*

Proof. The probability $\mathbb{P}\big[\sum_{j=1}^m \frac{\sigma_j}{N}|\sum_{i=1}^N \hat{\Delta}_j(\mathbf{x}_i)\,w_{ij}| \geq \frac{m\,\delta_{\max}\,t}{2} + 2\sqrt{\delta_{\max}\,t}\sum_{j=1}^m\|\hat{\Delta}_j\|_N\big]$ is upped bounded by $\mathbb{P}\big[\sum_{j=1}^m \frac{\sigma_j}{N}|\sum_{i=1}^N \hat{\Delta}_j(\mathbf{x}_i)\,w_{ij}| \geq \sum_{j=1}^m\{\frac{\delta_j\,t}{2} + 2\sqrt{\delta_j\,t}\|\hat{\Delta}_j\|_N\}\big]$. Applying the union bound, this is less than $\sum_{j=1}^m \mathbb{P}\big[\frac{\sigma_j}{N}|\sum_{i=1}^N \hat{\Delta}_j(\mathbf{x}_i)\,w_{ij}| \geq \frac{\delta_j\,t}{2} + 2\sqrt{\delta_j\,t}\|\hat{\Delta}_j\|_N\big]$. Applying Lemma 6.4, this is less than $\sum_{j=1}^m \exp\big(-\frac{2N\delta_j t}{\sigma_j^2}\big)$. Combining with Lemma 6.2, we have with probability at least $1 - \sum_{j=1}^m \exp\big(-\frac{2N\delta_j t}{\sigma_j^2}\big)$

$$\tfrac{1}{2}\|\hat{\Delta}\|_N^2 \leq \sum_{j=1}^m \tfrac{\sigma_j}{N}|\sum_{i=1}^N \hat{\Delta}_j(\mathbf{x}_i)\,w_{ij}| \leq \tfrac{m\,\delta_{\max}\,t}{2} + 2\sqrt{\delta_{\max}\,t}\sum_{j=1}^m\|\hat{\Delta}_j\|_N$$

Using Cauchy-Schwarz's inequality we obtain, $\sum_{j=1}^m\|\hat{\Delta}_j\|_N \leq \sqrt{m}\sqrt{\sum_{j=1}^m\|\hat{\Delta}_j\|_N^2} = \sqrt{m}\|\hat{\Delta}\|_N$. With high probability, $\|\hat{\Delta}\|_N$ is thus between the two roots of the quadratic

function g defined by $g : x \mapsto \frac{1}{2}x^2 - 2\sqrt{\alpha}\,x - \frac{\alpha}{2}$, where $\alpha := \delta_{\max}\, t\, m > 0$. By simple algebra, we get that $\|\hat{\Delta}\|_N \leq (2+\sqrt{5})\sqrt{\alpha}$ with high probability. □

We also want to control the uniform deviation $\|\mathbb{P} - \mathbb{P}_N\|_{\mathcal{L}} = \sup_{\boldsymbol{\theta}\in\Theta} |\rho(\boldsymbol{\theta}, \boldsymbol{\theta}^\star)^2 - \rho_N(\boldsymbol{\theta}, \boldsymbol{\theta}^\star)^2|$ by applying results on the *uniform laws of large numbers* on the loss class given in (6.9). In more generality, we consider a collection $X_1^N := \{X_1, \cdots, X_N\}$ of i.i.d. samples from some distribution $\mathbb{P}$ over $\mathcal{X}$ and a class $\mathcal{F}$ of real-valued integrable functions with domain $\mathcal{X}$, and studies the convergence properties of the *random variable* $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}} := \sup_{f\in\mathcal{F}} \left| \frac{1}{N}\sum_{i=1}^N f(X_i) - \mathbb{E}[f(X)] \right|$, where $\mathbb{P}_N$ is the *empirical distribution*, assigning mass $1/N$ to each of $X_1, \cdots, X_N$. The quantity $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}}$ measures the absolute deviation between the sample average and the population average. A classic approach consists in relating $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}}$ to the *Rademacher complexity* of $\mathcal{F}$. We first denote $(\sigma_1, \cdots, \sigma_N)$ the collection of Rademacher random variables, i.i.d. uniform in $\{\pm 1\}$.

Definition 6.3. *Given a class $\mathcal{F}$ of real-valued functions with domain $\mathcal{X}$ and a collection X_1^N of samples in $\mathcal{X}$, the Rademacher complexity of $\mathcal{F}$ is* $\mathcal{R}_N(\mathcal{F}) := \mathbb{E}_{X,\sigma}\left[\sup_{f\in\mathcal{F}} \left| \frac{1}{N}\sum_{i=1}^N \sigma_i f(X_i) \right| \right]$.

In particular, §3.4 of [5] gives an important result bounding the tail of the probability distribution of the random variable $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}}$ with the Rademacher complexity.

Theorem 6.3. *For any b-uniformly bounded function class $\mathcal{F}$, any positive integer N and any $\delta \in \mathbb{R}_{>0}$,* $\mathbb{P}\left[\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}} \leq 2\mathcal{R}_N(\mathcal{F}) + \sqrt{\frac{2b^2}{N}\ln(\frac{1}{\delta})} \right] \geq 1 - \delta$

We now seek to find an upper bound on $\mathcal{R}_N(\mathcal{F})$, which will inform us if $\mathcal{R}_N(\mathcal{F}) \to 0$. In addition, if we can derive a rate of convergence for $\mathcal{R}_N(\mathcal{F})$, we can find a lower bound on N guaranteeing that $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}} \leq \epsilon$ with probability at least $1 - \delta$. It also follows from Theorem 6.3.

Corollary 6.1. *For any uniformly bounded function class $\mathcal{F}$, if $\mathcal{R}_N(\mathcal{F}) \to 0$, then* $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}} \overset{a.s.}{\to} 0$.

Proof. For all $\delta \in \mathbb{R}_{>0}$, Theorem 3 implies $\mathbb{P}\left[\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}} \geq 2\,\mathcal{R}_N(\mathcal{F}) + \delta \right] \leq \exp\big(-\frac{N\delta^2}{2b^2}\big)$, thus $\sum_{N=1}^{\infty} \mathbb{P}\big[\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}} \geq 2\,\mathcal{R}_N(\mathcal{F}) + \delta \big] < \infty$. From Borel-Cantelli lemma, there exists, for each $\delta > 0$, a positive integer N_δ such that for all $N \geq N_\delta$, $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}} \leq 2\,\mathcal{R}_N(\mathcal{F}) + \delta$ almost surely. In particular, since $\mathcal{R}_N(\mathcal{F}) \to 0$, then we have $\|\mathbb{P}_N - \mathbb{P}\| \overset{\text{a.s.}}{\to} 0$. □

6.5 Upper bounds with the entropy integral

To derive explicit bounds from the results of Theorems 6.2 and 6.3, it remains to upper bound the Gaussian and Rademacher complexities . Hence we turn to the field of *approximation theory* [54], [132], [37] to bound complexities of a function class by a function of the integral of the *metric entropy*.

Definition 6.4. *(Covering number) A δ-cover of a set $\mathbb{T}$ with respect to a metric ρ is a set $\{\theta^1, \cdots, \theta^N\} \subset \mathbb{T}$ such that for each $\theta \in \mathbb{T}$, there exists some $i \in \{1, \cdots, N\}$ such that $\rho(\theta, \theta^i) \leq \delta$. The δ-covering number $N(\delta, \mathbb{T}, \rho)$ is defined as $N(\delta, \mathbb{T}, \rho) := \min\{N \mid \{\theta^1, \cdots, \theta^N\}$ is a δ-cover of $\mathbb{T}\}$.*

Definition 6.5. *(Metric entropy) Under the assumptions of Definition 6.4, the metric entropy of $\mathbb{T}$ is defined as the function $\delta \mapsto \log N(\delta, \mathbb{T}, \rho)$.*

We illustrate the notion of metric entropy on a class of strongly monotone and Lipschitz functions.

Lemma 6.5. *Let $\mathcal{M} = \{f : [0,1] \to \mathbb{R}_+$, L-Lipschitz, c-strongly monotone, $f(0) = 0\}$. Then the metric entropy $\log N(\delta, \mathcal{M}, \|\cdot\|_\infty)$ is upper bounded by $\left(\frac{L-c}{\delta} + 1\right) \log 2$ for $\delta \in (0, \frac{L-c}{2})$, and is equal to zero for $\delta \geq \frac{L-c}{2}$. This implies $\int_0^\infty \sqrt{\log N(\delta, \mathcal{M}, \|\cdot\|_\infty)}\, d\delta \leq (L-c)\sqrt{\log 2} \int_0^{\frac{1}{2}} \sqrt{\frac{1}{u} + 1}\, du$.*

Proof. Let $\epsilon \in \mathbb{R}_{>0}$ and assume $L > c$. We define $M := \lfloor 1/\epsilon \rfloor$ and a partition of $[0,1]$ into intervals $[(i-1)\epsilon,\, i\epsilon)$ for $i = 1, \cdots, M$, the last interval being $[M\epsilon,\, 1]$. We claim the collection $\hat{\mathcal{M}}$ of continuous functions such that $f(0) = 0$ and which have a constant slope c or L on each interval of the partition forms a $((L-c)\,\epsilon)$-cover of $\mathcal{M}$.

Let $f \in \mathcal{M}$. We prove by induction over i that we can construct a function $g \in \hat{\mathcal{M}}$ such that $|f(t) - g(t)| \leq (L-c)\,\epsilon$ for all $t \in [(i-1)\epsilon,\, i\epsilon)$, which will allow us to conclude that $\|f - g\|_\infty \leq (L-c)\,\epsilon$. The base case $i = 0$ follows from $f(0) = g(0)$. Now let $i \in \{1, \cdots, M\}$. We have by induction hypothesis $|f(i\epsilon) - g(i\epsilon)| \leq (L-c)\,\epsilon$. If $0 \leq g(i\epsilon) - f(i\epsilon) \leq (L-c)\,\epsilon$, we choose $g(t) = g(i\epsilon) + c(t - i\epsilon)$ over $[i\,\epsilon,\, (i+1)\,\epsilon)$. Then, for all $t \in [i\,\epsilon,\, (i+1)\,\epsilon)$,

$$
\begin{aligned}
f(t) - g(t) &\leq f(i\epsilon) - g(i\epsilon) + (L-c)\,(t - i\epsilon) \leq (L-c)\,(t - i\epsilon) \leq (L-c)\,\epsilon \\
g(t) - f(t) &\leq g(i\epsilon) - f(i\epsilon) + (c-c)\,(t - i\epsilon) \leq (L-c)\,\epsilon
\end{aligned}
$$

where we used the fact that f is L-Lipschitz in the first inequality, and $f(i\epsilon) - g(i\epsilon) \leq 0$ in the second one. Otherwise, $0 \leq f(i\epsilon) - g(i\epsilon) \leq (L-c)\,\epsilon$, and we choose $g(t) = g(i\epsilon) + L(t - i\epsilon)$ for which a similar analysis enables us to finally conclude that $\|f - g\|_\infty \leq (L-c)\,\epsilon$.

Note that the cover $\hat{\mathcal{M}}$ has cardinality $2^{\lfloor 1/\epsilon \rfloor + 1} = 2^{\lceil 1/\epsilon \rceil}$. Substituting $\epsilon = \delta/(L-c)$, the metric entropy $\log N(\delta, \mathcal{M}, \|\cdot\|_\infty)$ of $\mathcal{M}$ is bounded by $\lceil \frac{L-c}{\delta} \rceil \log 2$. Note that the expression holds for $L = c$. Indeed, $\mathcal{M}$ then has only one function $x \mapsto c\,x$ and the metric entropy is zero. And noting that $\mathrm{diam}_{\|\cdot\|_\infty}(\mathcal{M}) = L - c$, the mapping $x \mapsto \frac{L+c}{2} x$ is a δ-cover of $\mathcal{M}$ for $\delta > \frac{L-c}{2}$ and thus $\log N(\delta, \mathcal{M}, \|\cdot\|_\infty) = \log 1 = 0$ if $\delta > \frac{L-c}{2}$. Finally, noting that $\lceil \frac{L-c}{\delta} \rceil \leq \frac{L-c}{\delta} + 1$ completes our proof. □

We now present the connection between the Rademacher and Gaussian complexities and the metric entropy. Let us fix a collection x_1^N of elements of $\mathcal{X}$. We recall that $(\sigma_1, \cdots, \sigma_N)$ is the collection of Rademacher random variables (i.i.d. uniform in $\{\pm 1\}$). The quantity

$\sum_{i=1}^N \sigma_i f(x_i)$ that appears in the Rademacher complexity is a *sub-Gaussian process*[3] with respect to the Euclidean norm on the set $\mathcal{F}(x_1^N) := \big\{(f(x_1), \cdots, f(x_N)) \,|\, f \in \mathcal{F}\big\}$. Noting that the expected supremum $\mathbb{E}_\sigma\big[\sup_{f\in\mathcal{F}} |\sum_{i=1}^N \sigma_i f(x_i)|\big]$ of the sub-Gaussian process $\sum_{i=1}^N \sigma_i f(X_i)$ appears in the Rademacher complexity leads us to apply Dudley's theorem [57], [102, Chap. 11].

Theorem 6.4. *(Dudley's theorem) Let $\{X_\theta,\, \theta \in \mathbb{T}\}$ be a zero-mean sub-Gaussian process with respect to the metric ρ. Then $\mathbb{E}\big[\sup_{\theta\in\mathbb{T}} X_\theta\big] \leq 8\sqrt{2}\int_0^\infty \sqrt{\log N(u, \mathbb{T}, \rho)}du$.*

This gives us a bound on the expected suprema of sub-Gaussian processes with the entropy integral.

Proposition 6.1. *For any class $\mathcal{F}$ of real-valued functions such that $0 \in \mathcal{F}$, we have $\mathcal{R}_N(\mathcal{F}) \leq \frac{16\sqrt{2}}{N}\,\mathbb{E}_X\Big[\int_0^\infty \sqrt{\log N(u, \mathcal{F}(X_1^N), \|\cdot\|_2)}\, du\Big]$.*

Proof. Denoting $-\mathcal{F} := \{-f \,|\, f \in \mathcal{F}\}$ and the Rademacher process $r_N(f(x_1^N)) := \sum_i \sigma_i f(x_i)$, we have for any collection $x_1^N = \{x_1, \cdots, x_N\}$ of points

$$\sup_{f\in\mathcal{F}} |\textstyle\sum_i \sigma_i f(x_i)| = \sup_{f\in\mathcal{F}\cup-\mathcal{F}} \sum_i \sigma_i f(x_i) \leq \sup_{f\in\mathcal{F}} \sum_i \sigma_i f(x_i) + \sup_{f\in-\mathcal{F}} \sum_i \sigma_i f(x_i)$$

where we used that fact that $\sup_{f\in\mathcal{F}} \sum_i \sigma_i f(x_i)$ and $\sup_{f\in-\mathcal{F}} \sum_i \sigma_i f(x_i)$ are non-negative since $0 \in \mathcal{F}$ by assumption. Since the σ_i's are i.i.d. uniform in $\{\pm 1\}$, σ_i and $-\sigma_i$ have same distribution, and $\mathbb{E}_\sigma\big[\sup_{f\in\mathcal{F}} \sum_i \sigma_i f(x_i)\big] = \mathbb{E}_\sigma\big[\sup_{f\in-\mathcal{F}} \sum_i \sigma_i f(x_i)\big]$. And by Dudley's theorem, this last quantity is less than $\frac{8\sqrt{2}}{N}\displaystyle\int_0^\infty \sqrt{\log N(u, \mathcal{F}(x_1^N), \|\cdot\|_2)}\, du$ since the process $\{\sum_{i=1}^N \sigma_i f(x_i) \,|\, f \in \mathcal{F}\}$ is sub-Gaussian with respect to $\|\cdot\|_2$ on the set $\mathcal{F}(x_1^N)$. Indeed, if we denote by $f(x_1^N) := (f(x_1), \cdots, f(x_N))$ each element of $\mathcal{F}(x_1^N)$, we have for every, $f, f' \in \mathcal{F}$, and $\lambda \in \mathbb{R}$,

$$\mathbb{E}[e^{\lambda \sum_i \sigma_i (f(x_i) - f'(x_i))}] = \Pi_i \mathbb{E}[e^{\lambda\sigma_i(f(x_i)-f'(x_i))}] \leq \Pi_i e^{\frac{\lambda^2 (f(x_i)-f'(x_i))^2}{2}} = e^{\frac{\lambda^2 \|f(x_1^N) - f'(x_1^N)\|_2^2}{2}}$$

where we applied Hoeffding's lemma on each random variable $\sigma_i(f(x_i) - f'(x_i))$. We conclude with $\mathcal{R}_N(\mathcal{F}) = \frac{1}{N}\mathbb{E}_{X,\sigma}\big[\sup_{f\in\mathcal{F}} \big|\sum_{i=1}^N \sigma_i f(X_i)\big|\big] \leq \frac{2}{N}\mathbb{E}_{X,\sigma}\big[\sup_{f\in\mathcal{F}} \sum_{i=1}^N \sigma_i f(X_i)\big]$ which is bounded by $\frac{16\sqrt{2}}{N}\,\mathbb{E}_X\Big[\int_0^\infty \sqrt{\log N(u, \mathcal{F}(X_1^N), \|\cdot\|_2)}\, du\Big]$. □

Theorem 6.5. *For any indexed class $\mathcal{F} = \{f_{\boldsymbol{\theta}} \,|\, \boldsymbol{\theta} \in \Theta\}$ of real-valued functions which is $\tilde{L}$-smoothly parametrized with respect to a norm $\|\cdot\|_\Theta$ on Θ, and such that $0 \in \mathcal{F}$, $\mathcal{R}_N(\mathcal{F}) \leq \frac{16\sqrt{2}\tilde{L}}{\sqrt{N}}\int_0^\infty \sqrt{\log N(v, \Theta, \|\cdot\|_\Theta)}\, dv$*

[3] A collection of zero-mean random variables $\{X_\theta,\, \theta \in \mathbb{T}\}$ is a sub-Gaussian process with respect to a metric ρ on $\mathbb{T}$ if, for all $\theta, \theta' \in \mathbb{T}$, and $\lambda \in \mathbb{R}$, we have $\mathbb{E}[e^{\lambda(X_\theta - X_{\theta'})}] \leq \exp\big(\frac{\lambda^2\rho^2(\theta,\theta')}{2}\big)$

Proof. For any $f_{\boldsymbol{\theta}}, f_{\boldsymbol{\theta}'} \in \mathcal{F}$, and collection $\mathbf{x}_1^N$

$$\|f_{\boldsymbol{\theta}}(\mathbf{x}_1^N) - f_{\boldsymbol{\theta}'}(\mathbf{x}_1^N)\|_2^2 = \sum_{i=1}^N (f_{\boldsymbol{\theta}}(\mathbf{x}_i) - f_{\boldsymbol{\theta}'}(\mathbf{x}_i))^2 \leq \sum_{i=1}^N \|\!| f_{\boldsymbol{\theta}} - f_{\boldsymbol{\theta}'} |\!\|^2 \leq NL^2 \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\Theta^2$$

Thus $N(\delta, \mathcal{F}(x_1^N), \|\cdot\|_2) \leq N\big(\frac{\delta}{L\sqrt{N}}, \Theta, \|\cdot\|_\Theta\big)$ by definition of the covering number. Hence, a Lipschitz parameterization allows us to translate a cover of the parameter space Θ into a cover of the data-dependent function space $\mathcal{F}(x_1^N)$. Applying Proposition 1 with the change of variable $v := \frac{u}{L\sqrt{N}}$ in the entropy integral gives the claimed result. □

Hence, a Lipschitz parametrization allows us to bound the Rademacher complexity of $\mathcal{F}$ by the entropy integral of its parameter space Θ. When the metric entropy of the parameter space Θ can be derived explicitly, and its entropy integral is finite, such as when Θ is a class of strong monotone and real-valued functions on a compact domain (see Lemma 6.5), we can combine Theorem 6.3 and Theorem 6.5 to obtain explicit tail bounds on the uniform deviation $\|\mathbb{P} - \mathbb{P}_N\|_{\mathcal{F}}$.

Given a fixed collection of covariates $(\mathbf{x}_1, \cdots, \mathbf{x}_N)$, Dudley's theorem can also be used in conjunction with Theorem 6.2 to bound the prediction error $\|\hat{f} - f^\star\|_N^2 = \frac{1}{N}\sum_{i=1}^N \|\hat{f}(\mathbf{x}_i) - f^\star(\mathbf{x}_i)\|_2^2$. For a class $\mathcal{F}$ of real-valued functions, the norm $\|\cdot\|_N$ becoming $\|f\|_N^2 = \frac{1}{N}\sum_{i=1}^N f(\mathbf{x}_i)^2$, we define $\mathbb{B}_N(\delta, \mathcal{F}) := \{f \in \text{star}(\mathcal{F}) \,|\, \|f\|_N \leq \delta\}$. We have the following corollary, which enables us to upper bound the minimal $\delta^\star$ such that $\sigma\mathcal{G}_N(\delta, \mathcal{F}^\star) \leq \frac{\delta^2}{2}$ for all $\delta \geq \delta^\star$, and thus can be used for Theorem 6.2.

Corollary 6.2. *Suppose that the $f^\star$-shifted class $\mathcal{F}^\star$ of real-valued functions is star-shaped. For any $\sigma \in \mathbb{R}_{>0}$ and $\delta \in (0, \sigma]$ such that $\frac{16}{\sqrt{N}} \int_{\frac{\delta^2}{4\sigma}}^{\delta} \sqrt{\log N(u, \mathbb{B}_N(\delta, \mathcal{F}^\star), \|\cdot\|_N)} du \leq \frac{\delta^2}{4\sigma}$ satisfies the critical inequality $\sigma\mathcal{G}_N(\delta, \mathcal{F}^\star) \leq \frac{\delta^2}{2}$.*

Proof. Note that for any $\delta \in (0, \delta]$, we have $\frac{\delta^2}{4\sigma} < \delta$. It is possible to consider a $\frac{\delta^2}{4\sigma}$-covering $(f^1, \cdots, f^M)$ of $\mathbb{B}_N(\delta, \mathcal{F}^\star)$ in the norm $\|\cdot\|_N$. By definition of the covering set, for all $f \in \mathbb{B}_N(\delta, \mathcal{F}^\star)$, there exists $j \in [M]$ such that $\|f - f^j\|_N \leq \frac{\delta^2}{4\sigma}$. Hence

$$\begin{aligned}
\tfrac{1}{N}|\textstyle\sum_{i=1}^N w_i f(\mathbf{x}_i)| &\leq \tfrac{1}{N}|\textstyle\sum_{i=1}^N w_i f^j(\mathbf{x}_i)| + \tfrac{1}{N}|\textstyle\sum_{i=1}^N w_i (f(\mathbf{x}_i) - f^j(\mathbf{x}_i))| \\
&\leq \max_{j=1,..,M} \tfrac{1}{N}|\textstyle\sum_{i=1}^N w_i f^j(\mathbf{x}_i)| + \sqrt{\tfrac{\sum_{i=1}^N w_i^2}{N}} \sqrt{\tfrac{\sum_{i=1}^N (f(\mathbf{x}_i) - f^j(\mathbf{x}_i))^2}{N}} \\
&\leq \max_{j=1,..,M} \tfrac{1}{N}|\textstyle\sum_{i=1}^N w_i f^j(\mathbf{x}_i)| + \sqrt{\tfrac{\sum_{i=1}^N w_i^2}{N}} \tfrac{\delta^2}{4\sigma}
\end{aligned}$$

By taking the mean of the supremum over $\mathcal{F}^\star$, and using $\mathbb{E}_{\mathbf{w}} \sqrt{\frac{\sum_{i=1}^N w_i^2}{N}} \leq 1$, we have that:

$$\mathcal{G}_N(\delta) \leq \mathbb{E}_w[\max_{j=1,..,M} \tfrac{1}{N}|\textstyle\sum_{i=1}^N w_i f^j(\mathbf{x}_i)|] + \tfrac{\delta^2}{4\sigma}$$

We will now upper bound the first term of the right hand of the inequality. Let us fix $(\mathbf{x}_1, \cdots, \mathbf{x}_N)$. The random variables $\frac{1}{\sqrt{N}}\sum_{i=1}^N w_i f^j(\mathbf{x}_i)$ are zero-mean Gaussian processes

and are associated to $\|\cdot\|_N$ as a metric. Since $g \in \mathbb{B}_N(\delta, \mathcal{F}^\star)$ we can set the coarsest resolution of the chaining to δ and the tightest to $\frac{\delta^2}{4\sigma}$ as we can reconstruct the finite set with the minimal $\frac{\delta^2}{4\sigma}$-cover considered above. We can then bound the first term in the sum by: $\mathbb{E}_w[\max_{j=1,..,M} \frac{1}{N}|\sum_{i=1}^N w_i f^j(\mathbf{x}_i)|] \leq \frac{16}{\sqrt{N}} \int_{\frac{\delta^2}{4\sigma}}^{\delta} \sqrt{\log N(u, \mathbb{B}_N(\delta, \mathcal{F}^\star), \|\cdot\|_N)}du$. Thus using the assumption on the integral's upper bound, we obtain the desired result. □

Corollary 6.3. *Let $\mathcal{F}$ be the set of L-Lipschitz functions from $[0,1]$ to $\mathbb{R}$. Then the metric entropy satisfies $\frac{16}{\sqrt{N}} \int_{\frac{\delta^2}{4\sigma}}^{\delta} \sqrt{\log N(u, \mathbb{B}_N(\delta, \mathcal{F}^\star), \|\cdot\|_N)}du \precsim \sqrt{\frac{L\delta}{N}}$, and the critical inequality $\sigma\mathcal{G}_N(\delta, \mathcal{F}^\star) \leq \frac{\delta^2}{2}$ is satisfied for $\delta \simeq (\sigma^2 \frac{L}{N})^{\frac{1}{3}}$.*

Proof. Let us denote $\mathcal{F}_{Lip}(L) := \{f : [0,1] \to \mathbb{R} |\ f \text{ is } L\text{-Lipschitz}\}$. And given $f^\star \in \mathcal{F}_{Lip}(L)$ we have the inclusions

$$\mathcal{F}^\star = \mathcal{F}^\star_{Lip}(L) = \mathcal{F}_{Lip}(L) - f^\star \subseteq \mathcal{F}_{Lip}(L) - \mathcal{F}_{Lip}(L) \subseteq \mathcal{F}_{Lip}(2L)$$

And for any $f, f' \in \mathcal{F}_{Lip}(L)$, we observe that

$$\|f - f'\|_N^2 = \frac{1}{N}\sum_{i=1}^N (f(\mathbf{x}_i) - f'(\mathbf{x}_i))^2 \leq \frac{1}{N}\sum_{i=1}^N \|f - f'\|_\infty^2 = \|f - f'\|_\infty^2$$

Combining the above arguments, we obtain

$$N(u, \mathbb{B}_N(\delta, \mathcal{F}^\star), \|\cdot\|_N) \leq N(u, \mathbb{B}_N(\delta, \mathcal{F}_{\text{Lip}}(2L)), \|\cdot\|_\infty)$$

Using arguments similar to Lemma 4, we have $\log N(\delta, \mathcal{F}_{Lip}(2L), \|\cdot\|_\infty) \precsim \frac{L}{\delta}$, thus $\frac{16}{\sqrt{N}} \int_{\frac{\delta^2}{4\sigma}}^{\delta} \sqrt{\log N(u, \mathbb{B}_N(\delta, \mathcal{F}^\star), \|\cdot\|_N)}du \precsim \sqrt{\frac{L\delta}{N}}$. To satisfy the assumptions of Corollary 2, we have to choose δ such that $\sqrt{\frac{L\delta}{N}} \precsim \frac{\delta^2}{\sigma}$ which means $\delta \simeq (\frac{L\sigma^2}{N})^{\frac{1}{3}}$. □

6.6 Final results

Proposition 6.2. *Under the conditions of Theorem 6.1 with a Gaussian noise in (6.2) and $\epsilon \in (0,1]$ a sufficient number of samples to realize $\mathbb{P}\big[\rho_N(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)^2 \leq (2+\sqrt{5})^2 m\big(\sigma^2 \frac{cL_{\mathcal{P}}/L^2 + \tilde{L}_{\mathcal{P}}}{N(1-\sqrt{1-c^2/L^2})}\big)^{\frac{2}{3}}\big] \geq 1-\epsilon$ is in $O(\frac{\sigma(1-\sqrt{1-c^2/L^2})}{L_h L_{\mathcal{P}}})^2(\frac{\log(\frac{m}{\epsilon})}{2})^3$.*

Proof. Using the looser expression of the minimum probability of Theorem 2 given in the proof of Proposition 6.1, we can use Corollary 3 and replace δ by the expression of $\delta_{\max}$ given by Proposition 6.1 in $m \exp\big(-\frac{2n\delta_{\max}^2}{\sigma^2}\big) = \epsilon$. That gives us:

$$-\frac{2\delta_{\max}^2}{\sigma^2} = \log(\frac{\epsilon}{m})$$
$$n = \frac{\sigma^2}{2\delta_{\max}^2}\log(\frac{m}{\epsilon})$$

We can then replace δ_{max} by its expression:

$$n = \log\big(\frac{m}{\epsilon}\big)\frac{\sigma^2}{2}\big(\frac{n(1-\sqrt{1-c^2/L^2})}{L_h L_{\mathcal{P}}\sigma^2}\big)^{\frac{2}{3}}$$
$$n^{\frac{1}{3}} = \log\big(\frac{m}{\epsilon}\big)\frac{\sigma^2}{2}\big(\frac{(1-\sqrt{1-c^2/L^2})}{L_h L_{\mathcal{P}}\sigma^2}\big)^{\frac{2}{3}}$$
$$n = \big(\frac{\sigma(1-\sqrt{1-c^2/L^2})}{L_h L_{\mathcal{P}}}\big)^2\big(\frac{\log\big(\frac{m}{\epsilon}\big)}{2}\big)^3$$

□

Combining Theorem 6.1, Theorem 6.3, and Theorem 6.5, we derive smooth properties of the loss class (6.9) and convergence rates for the uniform deviation $\|\mathbb{P}-\mathbb{P}_N\|_{\mathcal{L}} = \sup_{\boldsymbol{\theta}\in\Theta}|\rho_N(\boldsymbol{\theta},\boldsymbol{\theta}^\star)^2 - \rho(\boldsymbol{\theta},\boldsymbol{\theta}^\star)^2|$.

Corollary 6.4. *Under the conditions of Theorem 6.1, we suppose $diam_{\|\cdot\|_\Theta}(\Theta) < \infty$. Then, for any L_h-Lipschitz function $h : \mathbb{R}^n \to \mathbb{R}^m$, the loss class $\mathcal{L}$ in (6.9) is $\tilde{L}$-smoothly parametrized and b-uniformly bounded with constants $\tilde{L} := 2\big(\frac{c\,L_h L_\Theta}{L^2(1-\sqrt{1-c^2/L^2})}\big)^2 diam_{\|\cdot\|_\Theta}(\Theta)$ and $b := \frac{\tilde{L}}{2}\, diam_{\|\cdot\|_\Theta}(\Theta)$.*

Proof. In order to simplify notation, we pose $h^\star_{\boldsymbol{\theta}}(\mathbf{p}) = h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}))$, and we define the loss function $\ell_{\boldsymbol{\theta}}(\mathbf{p}) := \|h^\star_{\boldsymbol{\theta}^\star}(\mathbf{p}) - h^\star_{\boldsymbol{\theta}}(\mathbf{p})\|_2$. We have, for all $\mathbf{p}\in\mathcal{P}$ and for all $\boldsymbol{\theta},\boldsymbol{\theta}'\in\Theta$

$$\begin{aligned}|\ell_{\boldsymbol{\theta}}(\mathbf{p}) - \ell_{\boldsymbol{\theta}'}(\mathbf{p})| &\le \|(h^\star_{\boldsymbol{\theta}^\star}(\mathbf{p}) - h^\star_{\boldsymbol{\theta}}(\mathbf{p})) - (h^\star_{\boldsymbol{\theta}^\star}(\mathbf{p}) - h^\star_{\boldsymbol{\theta}'}(\mathbf{p}))\|_2\\ &= \|h^\star_{\boldsymbol{\theta}}(\mathbf{p}) - h^\star_{\boldsymbol{\theta}'}(\mathbf{p})\|_2\\ &\le L_h\|\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}) - \mathbf{x}^\star_{\boldsymbol{\theta}'}(\mathbf{p})\|_2\\ &\le \tfrac{c\,L_h L_\Theta}{L^2(1-\sqrt{1-c^2/L^2})}\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|_\Theta\end{aligned}$$

where the first inequality is obtained from the triangle inequality, and the third inequality from Theorem 1. Let us denote $K := \frac{c\,L_h L_\Theta}{L^2(1-\sqrt{1-c^2/L^2})}$. For all $(\boldsymbol{\theta},\mathbf{p})\in\Theta\times\mathcal{P}$, we have

$$\ell_{\boldsymbol{\theta}}(\mathbf{p}) = \|h^\star_{\boldsymbol{\theta}^\star}(\mathbf{p}) - h^\star_{\boldsymbol{\theta}}(\mathbf{p})\|_2 \le K\|\boldsymbol{\theta}-\boldsymbol{\theta}^\star\|_\Theta \le K\,\mathrm{diam}_{\|\cdot\|_\Theta}(\Theta)$$

Hence the loss functions $\ell_{\boldsymbol{\theta}}(\cdot)$ are K-Lipschitz in $\boldsymbol{\theta}$ in the infinite norm, and $(K\,\mathrm{diam}_{\|\cdot\|_\Theta}(\Theta))$-uniformly bounded. Hence $\ell^2_{\boldsymbol{\theta}}$ is $(K\,\mathrm{diam}_{\|\cdot\|_\Theta}(\Theta))^2$-uniformly bounded and smoothly parametrized with constant $2K^2\,\mathrm{diam}_{\|\cdot\|_\Theta}(\Theta)$. □

Proposition 6.3. *Consider the setting of Corollary 6.4. If the entropy integral defined by $\int_0^\infty\sqrt{\log N(v,\Theta,\|\cdot\|_\Theta)}\,dv$ is finite, then $\mathcal{R}_N(\mathcal{L}) \le \frac{16\sqrt{2}\tilde{L}}{\sqrt{N}}\int_0^\infty\sqrt{\log N(v,\Theta,\|\cdot\|_\Theta)}\,dv$, and for all $\alpha\in(0,\frac{1}{2})$, the loss class $\mathcal{L}$ defined in (6.9) is such that $\|\mathbb{P}_N-\mathbb{P}\|_{\mathcal{L}} = O(\frac{1}{N^\alpha})$ almost surely. In addition, for any $\epsilon,\delta\in(0,1)$, a sufficient condition for having $\|\mathbb{P}-\mathbb{P}_N\|_{\mathcal{L}}\le\epsilon$ with probability at least $1-\delta$ is having a number of samples N such that $\sqrt{N} \ge \frac{1}{\epsilon}\big(32\sqrt{2}\tilde{L}\int_0^\infty\sqrt{\log N(v,\Theta,\|\cdot\|_\Theta)}\,dv + \sqrt{2b^2\log(1/\delta)}\big)$.*

Proof. The inequality on $\mathcal{R}_N(\mathcal{L})$ follows directly from Corollary 3 and Theorem 5, since $\mathcal{L}$ contains the zero function. Denoting $\kappa := 16\sqrt{2}\tilde{L}\int_0^\infty \sqrt{\log N(v,\Theta,\|\cdot\|_\Theta)}dv$, we get from Theorem 3, $\mathbb{P}\Big[\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}} \geq \frac{2\kappa}{\sqrt{N}} + \delta\Big] \leq \exp\big(-\frac{N\delta^2}{2b^2}\big)$ for all $\delta \in \mathbb{R}_{>0}$. Let α be a scalar in $(0, \frac{1}{2})$. Substituting $\delta := \frac{1}{N^\alpha}$, we have $\mathbb{P}\Big[\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}} \geq \frac{2\kappa}{\sqrt{N}} + \frac{1}{N^\alpha}\Big] \leq \exp\big(-\frac{N^{1-2\alpha}}{2b^2}\big)$. Hence, $\sum_{N=1}^\infty \mathbb{P}\Big[\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}} \geq \frac{2\kappa}{\sqrt{N}} + \frac{1}{N^\alpha}\Big] \leq \sum_{N=1}^\infty e^{-\frac{N^{1-2\alpha}}{2b^2}} < \infty$ since $1-2\alpha > 0$. From Borel-Cantelli lemma, there exists a positive integer N_0 such that for all $N \geq N_0$, we have $\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}} \leq \frac{2\kappa}{\sqrt{N}} + \frac{1}{N^\alpha}$ almost surely. We conclude our proof by noting that $\frac{2\kappa}{\sqrt{N}} = O(\frac{1}{N^\alpha})$ since $\alpha \in (0, \frac{1}{2})$.

Finally, from Theorem 3, we also have $\mathbb{P}\Big[\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{F}} \leq \frac{2\kappa}{\sqrt{N}} + \sqrt{\frac{2b^2}{N}\ln(\frac{1}{\delta})}\Big] \geq 1-\delta$. Hence a sufficient condition for having $\|\mathbb{P} - \mathbb{P}_N\|_{\mathcal{L}} \leq \epsilon$ with probability at least $1-\delta$ is for N to be such that $\frac{2\kappa}{\sqrt{N}} + \sqrt{\frac{2b^2}{N}\ln(\frac{1}{\delta})} \leq \epsilon$, which proves our claim. □

The results of Propositions 6.2 and 6.3 can be applied to the examples presented in Section 6.2. Illustrating Proposition 6.3 with the routing game, parametrized by edge cost functions $c_e(\cdot)$ that are c-strong-monotone and L-Lipschitz in the normalized flow $\frac{x_e}{m_e}$, and denoting $r := \frac{\min_{e\in\mathcal{E}} m_e}{\max_{e\in\mathcal{E}} m_e}$ the ratio of the smallest over the largest capacity, we have $\tilde{L} = 2\Big(\frac{cL_h r\sqrt{|\mathcal{E}|}\min_{e\in\mathcal{E}} m_e}{L^2(1-\sqrt{1-\frac{c^2r^2}{L^2}})}\Big)^2(L-c)$ and $b = \frac{\tilde{L}}{2}(L-c)$. For the consumer utility application, see Section 6.2, $\tilde{L} = 2\big(\frac{c\,L_h\,\max(1,\|\mathbf{x}_{\max}\|_2)}{L^2(1-\sqrt{1-c^2/L^2})}\big)^2(L-c+2r_{\max})$.

6.7 Conclusion

We have studied a class of supervised learning problems in which the objective function $f_{\boldsymbol{\theta}}$ is learned such that it gives solutions $\mathbf{x}^\star_{\boldsymbol{\theta}}$ that agree with our observations. We studied smooth properties of the solutions to convex programs, thus showing that the implicit functions $\mathbf{x}^\star_{\boldsymbol{\theta}}$ belong to a larger class of Lipschitz functions. This observation enables us to derive bounds on the empirical prediction error $\rho_N(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)$ defined in (6.7) and on the uniform deviation $\sup_{\boldsymbol{\theta}\in\Theta} |\rho(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star) - \rho_N(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)|$, giving insights on the training data needed to maintain small empirical prediction errors and high consistency of the empirical risk as a function of the complexity of the learned objective function.

Chapter 7

Imputing a Variational Inequality Function or a Convex Objective Function: a Robust Approach

7.1 Introduction

Motivation

Many decision processes are modeled as a Variational Inequality (VI) or Convex Optimization (CO) problem [61, 26]. However, the function that describes these processes are often difficult to estimate while their outputs (the decisions they describe) are often directly observable. For example, the traffic assignment problem considers a road network in which each road segment is associated to a delay that is a function of the volume of traffic on the arc [130]. The Wardrop's equilibrium principles [165] describe an equilibrium flow that is easily locally measurable by induction loop detectors or video cameras. While the delay functions are in general not observable, having accurate estimates of these functions is still crucial for urban planning. However, due to their cost of maintenance, traffic sensors are sparse, we thus present an approach robust to missing values and measurement errors. In consumer utility estimation, for example, the consumer is assumed to purchase various products from different companies in order to maximize a utility function minus the price paid, where the utility function measures the satisfaction the consumer receives from his purchases. In practice, the consumer's utility function is difficult to estimate but the consumer purchases, which is a function of the products' prices, are easily observable. We refer to [93, 20] for more examples, *e.g.*, value function estimation control.

Contributions and outline

Estimating the parameters of a process based on observations is related to various lines of work, *e.g.*, inverse reinforcement learning in robotics [123, 1], the inverse shortest path

problem [32], recovering the parameters of the Lyapunov function given a linear control policy [28, §10.6]. The field of *structural estimation* in economics estimates the parameters of observed equilibrium models, *e.g.* imputing production and demand functions [139, 3, 7]. In general, *inverse problems* have been studied quite extensively and we refer to [93, 20] for more references on the subject. In [93] (resp. [20]), a program is proposed to impute a convex objective (resp. a VI function) based on complete observations of nearly optimal decisions. The program is solved via CO.

After reviewing preliminary results in VI and CO in Section 7.2 and formally stating the problem in Section 7.3, our contributions in the remainder of the present chapter is as follows. In Section 7.4, we demonstrate that the methods presented in [93, 20] are in general not robust to noise and outliers in the data. In Section 7.5, we formulate our inverse problem as a weighted sum of a distance r_{obs} from the observations and residual functions r_{eq} in the form of duality gaps or Karush-Kuhn-Tucker (KKT) residuals, and show that our method is robust to noise and outliers while it avoids the disjunctive nature of the complementary condition. In Section 7.6, we show that the proposed weighted sum defines a set of Pareto efficient points whose closure contains a solution to the programs proposed in [93, 20]. Our method thus encompasses previous ones but performs better against noise and missing data. It also provides a conceptual way to recognize the implicit assumption of full noiseless observations made by previous inverse programming approaches. In Section 7.7, we compare the KKT residual and the duality gap and derive new sub-optimality results defined by the KKT residuals. In Section 9.4, an implementation framework is proposed. Finally, we apply our method to delay inference in the road network of Los Angeles, and consumer utility estimation and pricing in oligopolies in Sections 7.9 and 7.10.

7.2 Preliminaries

Variational Inequality (VI) and Convex Optimization (CO)

VI is used to model a broad class of problems from economics, convex optimization, and game theory, see, *e.g.* [61], for a comprehensive treatment of the subject. Mathematically, a VI problem is defined as follows:

Definition 7.1. *Given a closed, convex set $\mathcal{K} \subseteq \mathbb{R}^n$ and a map $F : \mathcal{K} \to \mathbb{R}^n$, the VI problem, denoted VI($\mathcal{K}, F$), consists in finding a vector $\boldsymbol{x} \in \mathcal{K}$ such that*

$$F(\boldsymbol{x})^T(\boldsymbol{u} - \boldsymbol{x}) \geq 0, \forall\, \boldsymbol{u} \in \mathcal{K} \tag{7.1}$$

For the remainder of the chapter, we suppose that $\mathcal{K}$ is a polyhedron, written in standard form:

$$\mathcal{K} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} = \mathbf{b},\, \mathbf{x} \geq 0\} \tag{7.2}$$

This allows different characterizations of solutions to VI($\mathcal{K}, F$). We define the *primal-dual system* associated to the Linear Program (LP) $\min_{u \in \mathcal{K}} F(\mathbf{x})^T\mathbf{u}$:

Definition 7.2. *(See [4, Th. 1].) Given VI($\mathcal{K}, F$), and $(\boldsymbol{x},\ by\ ) \in \mathbb{R}^n \times \mathbb{R}^n$, we define the associated primal-dual system as follows:*

$$\begin{array}{l} F(\boldsymbol{x})^T \boldsymbol{x} = \boldsymbol{b}^T\ by \\ \boldsymbol{A}^T\ by\ \leq F(\boldsymbol{x}) \\ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geq 0 \end{array} \tag{7.3}$$

In the above system, we say that $\mathbf{x}$ is *primal feasible* if $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq 0$, and $(\mathbf{x},\ \text{by}\ )$ is *dual feasible* if $\mathbf{A}^T$ by $\leq F(\mathbf{x})$. From LP strong duality, we have:

Theorem 7.1. *(See [4, Th. 1].) Let $\mathcal{K}$ be a polyhedron given by (7.2). Then $\boldsymbol{x} \in \mathbb{R}^n$ solves VI($\mathcal{K}, F$) if and only if there exists $by\ \in \mathbb{R}^n$ such that the pair $(\boldsymbol{x},\ by\ )$ satisfies the primal-dual system (7.3).*

We also define the *Karush-Kuhn-Tucker* (KKT) system of the VI($\mathcal{K}, F$):

Definition 7.3. *Let $\mathcal{K}$ be a polyhedron given by (7.2). Given a map F and $(\boldsymbol{x},\ by\ , \boldsymbol{\pi}) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n$, we define the associated KKT system as follows:*

$$\begin{array}{l} F(\boldsymbol{x}) = \boldsymbol{A}^T\ by\ + \boldsymbol{\pi} \\ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ \boldsymbol{x} \geq 0,\ \boldsymbol{\pi} \geq 0,\ \boldsymbol{x}^T\boldsymbol{\pi} = 0 \end{array} \tag{7.4}$$

Theorem 7.2. *(See [26, §5.5.3].) Let $\mathcal{K}$ be a polyhedron given by (7.2). Then a vector $\boldsymbol{x} \in \mathbb{R}^n$ solves VI($\mathcal{K}, F$) if and only if there exists $by\ , \boldsymbol{\pi} \in \mathbb{R}^n$ such that the tuple $(\boldsymbol{x},\ by\ , \boldsymbol{\pi})$ satisfies the KKT system (7.4).*

Convex Optimization (CO) is closely related to VI, see [26] for a comprehensive treatment on the subject. A CO problem is defined as follows:

Definition 7.4. *Given a closed, convex set $\mathcal{K} \in \mathbb{R}^n$ and a convex potential $f : \mathcal{K} \to \mathbb{R}$, the CO problem, denoted CO($\mathcal{K}, f$), is a program of the form:*

$$\min\ f(\boldsymbol{x}) \quad s.t. \quad \boldsymbol{x} \in \mathcal{K} \tag{7.5}$$

We have the following optimality condition to the CO($\mathcal{K}, f$):

Theorem 7.3. *(See [26, §4.2.3].) Given CO($\mathcal{K}, f$), suppose f differentiable. Then a vector $\boldsymbol{x} \in \mathcal{K}$ is an optimal solution to CO($\mathcal{K}, f$) if and only if:*

$$\nabla f(\boldsymbol{x})^T(\boldsymbol{u} - \boldsymbol{x}) \geq 0,\ \forall\, \boldsymbol{u} \in \mathcal{K} \tag{7.6}$$

Hence VI($\mathcal{K}, F$) can be seen as a generalization of CO($\mathcal{K}, f$) where the gradient ∇f is substituted by a general map F. Hence, when f is differentiable, the primal-dual and KKT systems are both optimality conditions for the CO($\mathcal{K}, f$).

Approximate solutions

We now focus on the VI($\mathcal{K}, F$) since it encompasses CO($\mathcal{F}, f$). The residual functions associated to the primal-dual and KKT systems are defined as

Definition 7.5. *(See [20].) Given VI($\mathcal{K}, F$), a residual function r_{PD} of the primal-dual system* (7.3) *is a non-negative function which satisfies for all $(\boldsymbol{x},\ by\ ) \in \mathbb{R}^n \times \mathbb{R}^n$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{x} \geq 0$, $\boldsymbol{A}^T\ by\ \leq F(\boldsymbol{x})$:*

$$r_{PD}(\boldsymbol{x},\ by\ ) = 0 \quad \Longleftrightarrow \quad \textit{(7.3) holds at } (\boldsymbol{x},\ by\ ) \tag{7.7}$$

Definition 7.6. *(See [93].) Given VI($\mathcal{K}, F$), a residual function r_{KKT} of the primal-dual system* (7.4) *is a non-negative function which satisfies for all $(\boldsymbol{x},\ by\ , \boldsymbol{\pi}) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{x} \geq 0$, $\boldsymbol{\pi} \geq 0$, $\boldsymbol{A}^T\ by\ \leq F(\boldsymbol{x})$*

$$r_{KKT}(\boldsymbol{x},\ by\ , \boldsymbol{\pi}) = 0 \quad \Longleftrightarrow \quad \textit{(7.4) holds at } (\boldsymbol{x},\ by\ , \boldsymbol{\pi}) \tag{7.8}$$

Residual functions are used as sub-optimality certificates in iterative methods for solving VI($\mathcal{K}, F$) and CO($\mathcal{K}, f$). As in [20] and [93], we specify r_{PD} and r_{KKT} as follows:

$$r_{\text{PD}}(\mathbf{x}) = F(\mathbf{x})^T\mathbf{x} - \mathbf{b}^T\ \text{by} \tag{7.9}$$

$$r_{\text{KKT}}(\mathbf{x},\ \text{by}\ , \boldsymbol{\pi}) = \left\| \begin{bmatrix} \alpha(F(\mathbf{x}) - \mathbf{A}^T\ \text{by}\ - \boldsymbol{\pi}) \\ \mathbf{x} \circ \boldsymbol{\pi} \end{bmatrix} \right\|_1 \tag{7.10}$$

where $\mathbf{x} \circ \boldsymbol{\pi} = [x_i\pi_i]_{i=1}^n$, $\|x\|_1 = \sum_{i=1}^n |x_i|$, and $\alpha > 0$ a weighting factor. The choice of r_{PD} in (7.9) is natural since primal feasibility ($\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq 0$) and dual feasibility ($\mathbf{A}^T$ by $\leq F(\mathbf{x})$) imply that r_{PD} is non-negative from weak LP duality [4, Cor. 1], and it is tied to the following optimality gaps for VI($\mathcal{K}, F$) and CO($\mathcal{K}, f$), taken from [61, §3.1.5] and [26, §9.3.1] respectively, for all $\mathbf{x} \in \mathcal{K}$:

$$r_{\text{VI}}(\mathbf{x}) = \max_{\mathbf{u} \in \mathcal{K}} F(\mathbf{x})^T(\mathbf{x} - \mathbf{u}) \tag{7.11}$$

$$r_{\text{CO}}(\mathbf{x}) = f(\mathbf{x}) - \min_{\mathbf{u} \in \mathcal{K}} f(\mathbf{u}) \tag{7.12}$$

Theorem 7.4. *(See [20, Th. 2].) Let $\mathcal{K}$ be a polyhedron given by* (7.2)*. Then the following holds for any $\epsilon \geq 0$ and $\boldsymbol{x} \in \mathcal{K}$:*

$$r_{VI}(\boldsymbol{x}) \leq \epsilon \Longleftrightarrow \exists\ by\ \in \mathbb{R}^n\ :\ \boldsymbol{A}^T\ by\ \leq F(\boldsymbol{x}),\ r_{PD}(\boldsymbol{x},\ by\ ) \leq \epsilon \tag{7.13}$$

In addition, if F is the gradient of a convex potential f, then, for all $\boldsymbol{x} \in \mathcal{K}$:

$$r_{VI}(\boldsymbol{x}) \leq \epsilon \implies r_{CO}(\boldsymbol{x}) \leq \epsilon \tag{7.14}$$

When primal and dual feasibilities hold, $r_{\mathrm{PD}} \leq \epsilon$ is equivalent to ϵ-suboptimality for $\mathrm{VI}(\mathcal{K}, F)$ with respect to r_{VI}. When $f = \nabla F$, $r_{\mathrm{PD}} \leq \epsilon$ is sufficient for ϵ-suboptimality for $\mathrm{CO}(\mathcal{K}, f)$ with respect to r_{CO}, but not necessary. To see this, consider a quadratic function $f : \mathbb{R} \to \mathbb{R}$ with minimum attained at $a > 0$:

$$\mathcal{K} = \mathbb{R}_+, \quad f(x) = (x-a)^2, \quad F(x) = \nabla f(x) = 2(x-a) \tag{7.15}$$

so $r_{\mathrm{CO}}(a+\epsilon) = f(a+\epsilon) = \epsilon^2$ while $r_{\mathrm{VI}}(a+\epsilon) = r_{\mathrm{PD}}(a+\epsilon) = (a+\epsilon)F(a+\epsilon) = 2(a+\epsilon)\epsilon$ is arbitrarily large as a goes to $+\infty$.

Distance from solutions

Assume $\mathrm{VI}(\mathcal{K}, F)$ (resp. $\mathrm{CO}(\mathcal{K}, f)$) has a unique solution $\mathbf{x}^\star$. An alternative sub-optimality condition is that $\|\mathbf{x} - \mathbf{x}^\star\| < \epsilon$ for $\mathbf{x} \in \mathcal{K}$. Main results rely on of strict and strong monotonicity of F (resp. convexity of f):

Definition 7.7. *Given a convex set $\mathcal{K} \subseteq \mathbb{R}^n$ and a function $f : \mathcal{K} \to \mathbb{R}$, f is said to be strictly convex on $\mathcal{K}$ if; $\forall\, \boldsymbol{x},\, \boldsymbol{x}' \in \mathcal{K}$ and $\alpha \in (0,1)$ such that $\boldsymbol{x} \neq \boldsymbol{x}'$*

$$f(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{x}') < \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{x}') \tag{7.16}$$

is said to be strongly convex on $\mathcal{K}$ if; $\exists\, c > 0$ such that $\forall\, \alpha \in (0,1)$, $\forall\, \boldsymbol{x},\, \boldsymbol{x}' \in \mathcal{K}$:

$$f(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{x}') \leq \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{x}') - \frac{c}{2}\alpha(1-\alpha)\|\boldsymbol{x} - \boldsymbol{x}'\|^2 \tag{7.17}$$

Definition 7.8. *Given a convex set $\mathcal{K} \subseteq \mathbb{R}^n$ and a map $F : \mathcal{K} \to \mathbb{R}^n$, F is said to be strictly monotone on $\mathcal{K}$ if*

$$(F(\boldsymbol{x}) - F(\boldsymbol{x}'))^T (\boldsymbol{x} - \boldsymbol{x}') \geq 0, \quad \forall\, \boldsymbol{x},\, \boldsymbol{x}' \in \mathcal{K} \tag{7.18}$$

strongly monotone on $\mathcal{K}$ if $\exists\, c > 0$ such that

$$(F(\boldsymbol{x}) - F(\boldsymbol{x}'))^T (\boldsymbol{x} - \boldsymbol{x}') \geq c\|\boldsymbol{x} - \boldsymbol{x}'\|^2, \quad \forall\, \boldsymbol{x},\, \boldsymbol{x}' \in \mathcal{K} \tag{7.19}$$

When f is differentiable, f is strictly (resp. strongly) convex is equivalent to ∇f is strictly (resp. strongly) monotone. Strong monotonicity allows us to bound $\|\mathbf{x} - \mathbf{x}^\star\|$ by the residual $r_{\mathrm{VI}}(\mathbf{x})$ in (7.11):

Theorem 7.5. *(See [126, Th. 4.1].) If VI($\mathcal{K}, F$) is such that $\mathcal{K} \subseteq \mathbb{R}^n$ is closed convex and F strongly monotone, VI($\mathcal{K}, F$) admits a unique solution $\boldsymbol{x}^\star$ and:*

$$\|\boldsymbol{x} - \boldsymbol{x}^\star\|_2 \leq \sqrt{r_{VI}(\boldsymbol{x})/c}, \quad \forall\, \boldsymbol{x} \in \mathcal{K} \tag{7.20}$$

in addition, if $\exists\, f : F = \nabla f$, then $\boldsymbol{x}^\star$ is the unique solution to CO($\mathcal{K}, f$) and:

$$\|\boldsymbol{x} - \boldsymbol{x}^\star\|_2 \leq \sqrt{2\, r_{CO}(\boldsymbol{x})/c}, \quad \forall\, \boldsymbol{x} \in \mathcal{K} \tag{7.21}$$

If F is only strictly monotone, then $\mathrm{VI}(\mathcal{K}, F)$ admits at most one solution [142]. If the solution $\mathbf{x}^\star$ exists, then strict monotonicity is not strong enough for a bound similar to (7.20).

7.3 Problem statement

We present our problem statement in the most general case. We refer to Sections 7.9 and 7.10 for illustration of the problem in traffic assignment and consumer utility respectively. Let us consider a process in which decisions $\mathbf{x}$ are made by solving a parametric variational inequality $\mathrm{VI}(\mathcal{K}(\mathbf{p}), F(\cdot, \mathbf{p}))$, for a set of parameter values $\mathbf{p} \in \mathcal{P}$:

$$F(\mathbf{x}, \mathbf{p})^T(\mathbf{u} - \mathbf{x}) \geq 0, \quad \forall\, \mathbf{u} \in \mathcal{K}(\mathbf{p}) \tag{7.22}$$

$$\mathcal{K}(\mathbf{p}) := \{\mathbf{x} \in \mathbb{R}^n \,:\, \mathbf{A}(\mathbf{p})\mathbf{x} = \mathbf{b}(\mathbf{p}),\, \mathbf{x} \geq 0\} \tag{7.23}$$

where both the map $F(\cdot, \mathbf{p})$ and polyhedron $\mathcal{K}(\mathbf{p})$ depend on $\mathbf{p}$. The definitions and theorems in Section 7.2 apply for each $\mathbf{p} \in \mathcal{P}$, and the dependence of the residuals on $\mathbf{p}$ are made explicit with $r_{\mathrm{PD}}(\mathbf{x},\ \text{by}\ , \mathbf{p})$, $r_{\mathrm{KKT}}(\mathbf{x},\ \text{by}\ , \boldsymbol{\pi}, \mathbf{p})$, etc.

Inputs: We are given $\mathbf{A}(\mathbf{p})$, $\mathbf{b}(\mathbf{p})$ for all $\mathbf{p}$, along with a parametric observation process $g(\cdot, \mathbf{p}) : \mathbb{R}^n \to \mathbb{R}^q$ and N noisy observations

$$\mathbf{z}^{(j)} := g(\mathbf{x}^{(j)}, \mathbf{p}^{(j)}) + \mathbf{w}^{(j)}, \quad j = 1, \cdots, N \tag{7.24}$$

of (approximate) solutions $\mathbf{x}^{(j)}$ to $\mathrm{VI}(\mathcal{K}(\mathbf{p}^{(j)}), F(\cdot, \mathbf{p}^{(j)}))$ with random noise $\mathbf{w}^{(j)} \in \mathbb{R}^q$ and associated parameters $\mathbf{p}^{(j)}$. Unless $g(\cdot, \mathbf{p})$ is an injection from $\mathcal{K}(\mathbf{p})$ to $\mathbb{R}^q$ for all $\mathbf{p}$, the observation $\mathbf{z}^{(j)}$ contains in general less information than $\mathbf{x}^{(j)}$, thus (7.24) is our missing data model.

Objective: We want to impute the parametric map $F(\cdot, \mathbf{p})$ and the decision vectors $\mathbf{x}^{(j)}$ such that, for all j:

(a) $\mathbf{x}^{(j)}$ is an approximate solution to $\mathrm{VI}(\mathcal{K}(\mathbf{p}^{(j)}), F(\mathbf{p}^{(j)}))$.

(b) $\mathbf{x}^{(j)}$ agrees with the observations $\mathbf{z}^{(j)}$.

Formalization: Using Theorem 2.4, objective (a) consists in imputing a parametric map $F(\cdot, \mathbf{p})$ and a collection of decision vectors $\mathbf{x}^{(j)} \in \mathcal{K}(\mathbf{p}^{(j)})$, along with dual variables $\text{by}^{(j)}$ with $\mathbf{A}(\mathbf{p}^{(j)})^T\ \text{by}^{(j)} \leq F(\mathbf{x}^{(j)}, \mathbf{p}^{(j)})$, such that the following sum of residuals is minimized:

$$r_{\mathrm{eq}} := \sum_{j=1}^{N} r_{\mathrm{PD}}(\mathbf{x}^{(j)},\ \text{by}^{(j)}, \mathbf{p}^{(j)}) \tag{7.25}$$

Objective (b) consists in minimizing, with ϕ a non-negative convex function in $(\mathbf{x},\ \text{by}\ )$ such that $\phi(\mathbf{x},\ \text{by}\ ) = 0 \Leftrightarrow \mathbf{x} =\ \text{by}$:

$$r_{\mathrm{obs}} := \sum_{j=1}^{N} \phi\left(g(\mathbf{x}^{(j)}, \mathbf{p}^{(j)}),\, \mathbf{z}^{(j)}\right) \tag{7.26}$$

As discussed in [93, 20], the parametric map $F(\cdot, \mathbf{p})$ must be searched in a restricted space $\mathcal{F}$. Since the construction of $\mathcal{F}$ is not the focus of the present chapter, further details will be presented in Section 7.8.

7.4 Previous methods

Inverse Variational Inequality

Formulation: Bertsimas et al. [20] impute $F(\cdot, \mathbf{p})$ given (perfect) observations $\mathbf{z}^{(j)} = \mathbf{x}^{(j)}$ (*i.e.* $g(\cdot, \mathbf{p}) =$Id) of approximate solution to VI($\mathcal{K}(\mathbf{p}^{(j)}), F(\cdot, \mathbf{p}^{(j)})$) by setting objective r_{obs} in (7.26) to zero and solving:

$$\begin{array}{ll} \min\limits_{F,\ \text{by}} & r_{\text{eq}} = \sum_{j=1}^{N} r_{\text{PD}}(\mathbf{z}^{(j)},\ \text{by}\ {}^{(j)}, \mathbf{p}^{(j)}) \\ \text{s.t.} & \mathbf{A}(\mathbf{p}^{(j)})^T\ \text{by}\ {}^{(j)} \leq F(\mathbf{z}^{(j)}, \mathbf{p}^{(j)}), \quad \forall\, j \end{array} \tag{7.27}$$

If $r_{\text{PD}}(\mathbf{x},\ \text{by}\ , \mathbf{p}) = F(\mathbf{x}, \mathbf{p})^T\mathbf{x} - \mathbf{b}(\mathbf{p})^T\ \text{by}\ $ and $F(\cdot, \mathbf{p})$ is restricted to a finite dimensional affine parametrization $\sum_{i=1}^{K} a_i F_i(\cdot, \mathbf{p})$ with parameters $\mathbf{a} \in \mathbb{R}^K$ restricted to a convex set, then (7.27) is a convex program.

Limitations: The above formulation, which we will refer to as *Inverse VI*, assumes that we have complete observations, which is not possible in many applications, such as in traffic assignment, see Section 7.9. In addition, (7.27) overlooks the measurement errors by tightly fitting an equilibrium model to the (complete) observations, thus attempting to explain random (irreducible) errors by a deterministic process. For example, consider the following process:

$$\min_{\mathbf{x} \geq 0} (x - a)^2 \tag{7.28}$$

where $a > 0$ needs to be imputed. The associated primal-dual system is:

$$x(x - a) = 0,\ x \geq a,\ x \geq 0 \tag{7.29}$$

Given N observations $z^{(j)} \geq 0$, solving (7.27) applied to our particular case:

$$\min_{\hat{a} \geq 0} \sum_{j=1}^{N} z^{(j)}(z^{(j)} - \hat{a}) \quad \text{s.t.} \quad \hat{a} \leq \min_j z^{(j)} \tag{7.30}$$

gives an imputed parameter $\hat{a} = \min_j z^{(j)}$. Independently of the data size, a single measurement error of δ in a set of perfect observations can induce a large mean residual error. In the above example, if $z^{(1)} = a - \delta$ and $z^{(j)} = a$ for $j = 2, \cdots, N$, then the imputed value is $\hat{a} = a - \delta$, with mean residual error:

$$\frac{1}{N}\sum_{j=1}^{N} z^{(j)}(z^{(j)} - \hat{a}) = \frac{(N-1)\,a\,\delta}{N} \longrightarrow a\delta \quad \text{as} \quad N \longrightarrow +\infty \tag{7.31}$$

Inverse programming as a bilevel program

Formulation: An intuitive approach is via bilevel optimization in which the metric $r_{\text{obs}} = \sum_j \phi(\mathbf{x}^{(j)}, \mathbf{z}^{(j)})$ in (7.26) is minimized with $\mathbf{x}^{(j)}$ the decision vector predicted by the imputed process. We refer to, *e.g.*, [41] for the problem of OD matrix estimation given link cost functions and observed flows. Applying bilevel optimization to our function estimation problem:

$$\begin{array}{ll} \min\limits_{F,\,\mathbf{x},\ \text{by}} & r_{\text{obs}} = \sum_{j=1}^{N} \phi\left(g(\mathbf{x}^{(j)}, p^{(j)}),\, \mathbf{z}^{(j)}\right) \\ & \mathbf{x}^{(j)} \text{ is solution to VI}(\mathcal{K}(\mathbf{p}^{(j)}), F(\mathbf{p}^{(j)})), \quad \forall\, j \end{array} \tag{7.32}$$

With a good choice of ϕ, (7.32) can be robust to noise. For example, consider N observations $z^{(j)}$ of the minimization process (7.28). Then, (7.32) becomes:

$$\min_{\hat{a}\geq 0,\, x} \sum_{j=1}^{N} \phi(x^{(j)},\, z^{(j)}) \quad \text{s.t.} \quad x^{(j)} \in \operatorname*{argmin}_{u\geq 0} (u - \hat{a})^2, \quad \forall\, j \tag{7.33}$$

We note that $\hat{a}$ is the sample mean when $\phi(x) = x^2$, while $\hat{a}$ is the sample median when $\phi(x) = |x|$. Hence, formulation (7.32) allows different choices of penalty functions ϕ on the observation residuals, thus a fitting more robust to noise. We will refer to (7.32) as the *Bilevel Program* (BP) in the context of inverse programming.

Limitations: In general, the solution set of VI$(\mathcal{K}(\mathbf{p}^{(j)}), F(\mathbf{p}^{(j)}))$ does not have a closed-form expression, thus one approach replaces the constraint in (7.32) by the primal-dual system (7.3) or KKT system (7.4) to reduce (7.32) to a single-level program. However, the complementary condition $r_{\text{PD}}(\mathbf{z}^{(j)},\ \text{by}\ ^{(j)}, \mathbf{p}^{(j)}) = 0$ in the constraints causes the standard Mangasarian-Fromovitz Constraint Qualification (MFCQ) to be violated at any feasible point [172], hence generating severe numerical difficulties, see [89, 110].

7.5 Our method

A Weighted Sum Program

We minimize simultaneously objectives (7.25) and (7.26) subject to primal and dual feasibilities by considering the linear combination $w_{\text{eq}} r_{\text{eq}} + w_{\text{obs}} r_{\text{obs}}$:

$$\begin{array}{ll} \min\limits_{F,\mathbf{x},\ \text{by}} & w_{\text{eq}}\, \Sigma_{j=1}^{N} r_{\text{PD}}(\mathbf{x}^{(j)},\ \text{by}\ ^{(j)}, \mathbf{p}^{(j)}) + w_{\text{obs}}\, \Sigma_{j=1}^{N} \phi\left(g(\mathbf{x}^{(j)}, p^{(j)}),\, \mathbf{z}^{(j)}\right) \\ \text{s.t.} & \mathbf{x}^{(j)} \in \mathcal{K}(\mathbf{p}^{(j)}), \quad \forall\, j \\ & \mathbf{A}(\mathbf{p}^{(j)})^T \text{ by } ^{(j)} \leq F(\mathbf{x}^{(j)}, \mathbf{p}^{(j)}), \quad \forall\, j \end{array} \tag{7.34}$$

where w_{eq} and w_{obs} are positive scalars that articulate the preferences between the two objectives. This approach is known as the *Weighted Sum method* in Pareto Optimization (PO) theory, and is sufficient for Pareto optimality, *i.e.*, it is not possible to strictly decrease

one objective among r_{eq} and r_{obs} without strictly increasing the other one, see, *e.g.*, [116, 58] for further details on PO.

One approach to explore the Pareto curve is shown in Algorithm 7.1. In step 1, we note that it is often desirable to scale the objective functions to have a consistent comparison between them. Varying the weights provides information about available trade-offs between the objectives. Specifically, for each of the different weights in step 2, if the solution is such that r_{eq} and r_{obs} are large, it means that either our model is not a good model to explain the observations, or that our observations are very noisy.

Algorithm 7.1 `Weighted-sum(·)` Weighted sum method

1: Normalize objectives (7.25) and (7.26) for consistent comparisons.
2: Solve (7.34) with $w_{\text{obs}} + w_{\text{eq}} = 1$ and $w_{\text{obs}} \in \{0.001, 0.01, 0.1, 0.9, 0.99, 0.999\}$
3: Check the values of (7.25) and (7.26).

The proposed weighted Sum Program (WSP) is robust since it accommodates different penalty functions ϕ depending on the type of measurement errors, *e.g.* $\phi(\mathbf{x}, \mathbf{z}) = \|\mathbf{x} - \mathbf{z}\|_1$ for robustness to outliers, and $\phi(\mathbf{x}, \mathbf{z}) = \|\mathbf{x} - \mathbf{z}\|_2$ for robustness to Gaussian noise; see, *e.g.*, [26, §6.1]. In addition, our WSP can be seen as a penalty method for constrained optimization that mitigates numerical difficulties by minimizing $r_{\text{PD}}(\mathbf{z},\ \text{by}\ , \mathbf{p})$ instead of setting $r_{\text{PD}}(\mathbf{z},\ \text{by}\ , \mathbf{p})$ to 0.

Example

Given N observations $z^{(j)}$ of $\min_{x \geq 0} (x - a)^2$, the WSP (7.34) is:

$$\begin{array}{ll} \min\limits_{\hat{a},\, x} & w_{\text{eq}} \sum_{j=1}^{N} x^{(j)}(x^{(j)} - \hat{a}) + w_{\text{obs}} \sum_{j=1}^{N} \phi(x^{(j)},\, z^{(j)}) \\ \text{s.t.} & x^{(j)} \geq 0, \quad \forall\, j \\ & 0 \leq \hat{a} \leq \min\limits_{j} x^{(j)} \end{array} \tag{7.35}$$

We now set $w_{\text{obs}} = \alpha$, $w_{\text{eq}} = 1 - \alpha$ for $\alpha \in (0, 1)$ and $\phi(x, y) = |x - y|$. Following the case study in Section 7.4, assume the observations are $z^{(1)} = a - \delta$ and $z^{(j)} = a$ for $j = 2, \cdots, N$, then the set of Pareto optimal points are

$$\hat{a} = x^{(1)} \in [a - \delta,\, a], \quad x^{(j)} = a, \quad \text{for } j = 2, \cdots, N \tag{7.36}$$

Then, given estimate $\hat{a} \in [a - \delta,\, a]$, objectives r_{eq} in (7.25) and r_{obs} in (7.26) are:

$$r_{\text{eq}} = (N - 1)\, a\, (a - \hat{a}) \tag{7.37}$$

$$r_{\text{obs}} = |a - \delta - \hat{a}| = \hat{a} + \delta - a \tag{7.38}$$

Solving $\min_{\hat{a}\in[a-\delta,\,a]} w_{\text{eq}} r_{\text{eq}} + w_{\text{obs}} r_{\text{obs}} = (1-\alpha)(N-1)a(a-\hat{a}) + \alpha(\hat{a}+\delta-a)$:

$$\begin{aligned} w_{\text{obs}} = \alpha < \tfrac{a(N-1)}{1+a(N-1)} &\implies \hat{a} = a, && r_{\text{eq}} = 0, && r_{\text{obs}} = \delta \\ w_{\text{obs}} = \alpha > \tfrac{a(N-1)}{1+a(N-1)} &\implies \hat{a} = a-\delta, && r_{\text{eq}} = (N-1)a\delta, && r_{\text{obs}} = 0 \end{aligned} \tag{7.39}$$

In this case, if w_{obs} is close enough to 1, r_{eq} is large and equal to the one in the Inverse VI (see (7.31)), while with w_{obs} smaller, we have a small observation residual r_{obs} and $r_{\text{eq}} = 0$. Thus, the estimation is good for w_{obs} close enough to 0 despite a fit to the data that is not perfect due to measurement errors.

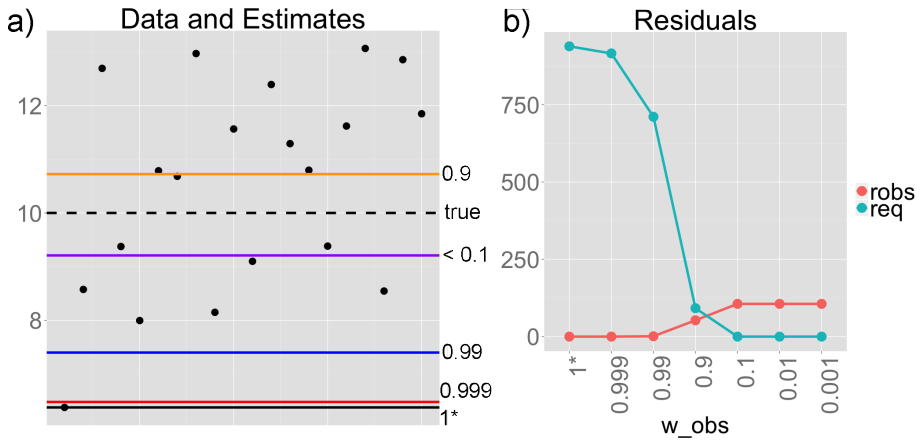

Figure 7.1: Imputation of the parametric program from $N = 20$ noisy observations with mean 10 shown in Figure a). The estimates are shown by horizontal lines labelled by the value of .

In a second example, we randomly generate $N = 20$ independent and identically distributed (i.i.d.) samples $z^{(j)}$ from a Gaussian distribution with mean $a = 10$ and variance $\sigma = 5$. We apply our WSP (7.35) with $\phi(x, y) = (x - y)^2$. The estimates $\hat{a}$ are shown in Figure 7.1.a), and the values of the residuals $r_{\text{obs}} = \sum_{j=1}^{N} x^{(j)}(x^{(j)} - \hat{a})$ and $r_{\text{eq}} = \sum_{j=1}^{N}(x^{(j)} - z^{(j)})^2$ in Figure 7.1.b), for $w_{\text{obs}} \in \{0.001, 0.01, 0.1, 0.9, 0.99, 0.999\}$ and $w_{\text{eq}} = 1 - w_{\text{obs}}$. In addition, we compare our method to the Inverse VI (7.30), which is tagged with label $w_{\text{obs}} = 1$ in Figure 7.1. For $w_{\text{obs}} < 0.1$, $r_{\text{eq}} = 0$, $\hat{a} = 9.2$ is close to 10, and r_{obs} is small, while for $w_{\text{obs}} > 0.99$ and for the Inverse VI, r_{eq} is large and $\hat{a}$ is largely under-estimating $a = 10$. In the presence of Gaussian noise, our WSP also performs well. Finally, we note that for large values of w_{obs}, our method behaves similarly to the Inverse VI method.

7.6 Relation to previous methods

Preliminary results

Intuitively, as $(w_{\text{eq}}, w_{\text{obs}})$ approaches $(0, 1)$, the WSP (7.34) mimics the Inverse VI (7.27), and as $(w_{\text{eq}}, w_{\text{obs}})$ approaches $(1, 0)$, it mimics the BP (7.32). Formally, given f_1, f_2 two

non-negative continuous functions, a compact set $\mathcal{C} \subseteq \mathbb{R}^n$, and $w_1, w_2 > 0$, consider the general weighted sum program along with its solution set $\mathcal{S}(w_1, w_2)$ and the set $\mathcal{S}$ of all Pareto efficient points associated to it:

$$\min\ w_1 f_1(\mathbf{u}) + w_2 f_2(\mathbf{u}) \quad \text{s.t.} \quad \mathbf{u} \in \mathcal{C} \tag{7.40}$$

$$\mathcal{S}(w_1, w_2) := \arg\min_{\mathbf{u}\in\mathcal{C}} w_1 f_1(\mathbf{u}) + w_2 f_2(\mathbf{u}) \tag{7.41}$$

$$\mathcal{S} := \Big\{(w_1, w_2, \mathbf{u}^\star)\ :\ w_1 \in (0,1),\ w_2 = 1 - w_1,\ \mathbf{u}^\star \in \mathcal{S}(w_1, w_2)\Big\} \tag{7.42}$$

Since $\mathcal{C}$ is compact, $\mathcal{S}(w_1, w_2) \neq \emptyset$ for any w_1, w_2, hence $\mathcal{S}$ is well-defined. We also assume there exists $\mathbf{u} \in \mathcal{C}$ such that $f_1(\mathbf{u}) = 0$, and define the following constrained program and its approximate objective value $f_2^\star(\epsilon)$:

$$\min\ f_2(\mathbf{u}) \quad \text{s.t.} \quad f_1(\mathbf{u}) = 0,\ \mathbf{u} \in \mathcal{C} \tag{7.43}$$

$$f_2^\star(\epsilon) := \min_{\mathbf{u}\in\mathcal{C}\,:\, f_1(\mathbf{u})\leq\epsilon} f_2(\mathbf{u}), \quad \forall\, \epsilon \geq 0 \tag{7.44}$$

Lemma 7.1. *Let $\mathcal{S}$ be a set described by (7.42). Then for any $(w_1, w_2, \boldsymbol{u}^\star) \in \mathcal{S}$:*

$$f_1(\boldsymbol{u}^\star) \leq (w_1^{-1} - 1) f_2^\star(0) \tag{7.45}$$

$$f_2(\boldsymbol{u}^\star) \leq f_2^\star(0) \tag{7.46}$$

Proof. Let $\mathbf{u} \in \mathcal{C}$ such that $f_1(\mathbf{u}) = 0$. For any $(w_1, w_2, \mathbf{u}^\star) \in \mathcal{S}$, we have $w_1 f_1(\mathbf{u}^\star) + w_2 f_2(\mathbf{u}^\star) \leq w_2\, f_2(\mathbf{u})$, hence, from non-negativity of f_1, f_2 and positivity of w_1, w_2:

$$f_2(\mathbf{u}^\star) \leq f_2(\mathbf{u}) \tag{7.47}$$

$$f_1(\mathbf{u}^\star) \leq (w_2/w_1) f_2(\mathbf{u}) = ((1 - w_1)/w_1) f_2(\mathbf{u}) = (w_1^{-1} - 1) f_2(\mathbf{u}) \tag{7.48}$$

Since this is true for all $\mathbf{u} \in \mathcal{C}$ such that $f_1(\mathbf{u}) = 0$, minimizing f_2 for such $\mathbf{u}$ completes the proof. □

Lemma 7.2. *Let $\mathcal{S}$ be a set described by (7.42). Then $\{f_2(\boldsymbol{u}^\star)\}_{\boldsymbol{u}^\star\in\mathcal{S}(w_1,w_2)}$ converges uniformly to $f_2^\star(0)$ as $w_1 \longrightarrow 1$. There also exists a solution $\bar{\boldsymbol{u}}$ to (7.43) and a sequence $(w_1^{(n)}, w_2^{(n)}, \boldsymbol{u}_n)_{n\in\mathbb{N}} \in \mathcal{S}^{\mathbb{N}}$ such that:*

$$(w_1^{(n)}, w_2^{(n)}, \boldsymbol{u}_n) \longrightarrow (1, 0, \bar{\boldsymbol{u}}) \quad \text{as } n \longrightarrow +\infty \tag{7.49}$$

In addition, if (7.43) admits a unique solution $\bar{\boldsymbol{u}}$, any sequence $(w_1^{(n)}, w_2^{(n)}, \boldsymbol{u}_n)_{n\in\mathbb{N}} \in \mathcal{S}^{\mathbb{N}}$ such that $w_1^{(n)} \longrightarrow 1$ satisfies $\boldsymbol{u}^{(n)} \longrightarrow \bar{u}$.

Proof. First, we want to prove that $f_2^\star(\cdot)$ is continuous at 0. We note that $f_2^\star(\cdot)$ is non-increasing on $\mathbb{R}_+$, hence it has a limit from the right at 0, which we denote $f_2^\star(0_+)$. Given any sequence $(\epsilon_n)_{n\in\mathbb{N}} \in \mathbb{R}_+^{\mathbb{N}}$ such that $\epsilon_n \to 0$, there exists a sequence $(\mathbf{u}_n)_{n\in\mathbb{N}}$ such that

$\mathbf{u}_n \in \underset{\mathbf{u}\in\mathcal{C}: f_1(\mathbf{u})\leq\epsilon_n}{\arg\min} f_2(\mathbf{u})$ for all $n \in \mathbb{N}$, since $\mathcal{C}$ is compact. Hence, $f_2^\star(\epsilon_n) = f_2(\mathbf{u}_n)$ for all $n \in \mathbb{N}$. From compactness, there exists a convergent subsequence $(\tilde{\epsilon}_n, \tilde{\mathbf{u}}_n)_{n\in\mathbb{N}}$ of $(\epsilon_n, \mathbf{u}_n)_{n\in\mathbb{N}}$, and its limit $(0, \bar{\mathbf{u}})$ is such that $\mathbf{u} \in \mathcal{C}$, $f_1(\bar{\mathbf{u}}) = 0$ and $f_2^\star(0_+) = f_2(\bar{\mathbf{u}}) \leq f_2^\star(0)$ from continuity of f_1 and f_2. By definition of $f_2^\star(0)$, we must have $f_2^\star(0_+) = f_2^\star(0)$. Hence $f_2^\star(\cdot)$ is continuous at 0.

To prove the first part of the lemma, we denote $g(w_1) := (w_1^{-1} - 1)f_2^\star(0)$. For any $(w_1, w_2, \mathbf{u}^\star) \in \mathcal{S}$, we have $f_1(\mathbf{u}^\star) \leq g(w_1)$ from lemma 6.1, hence $f_2^\star(g(w_1)) \leq f_2(\mathbf{u}^\star) \leq f_2^\star(0)$ by definition of $f_2^\star(\epsilon)$. Thus, by continuity of $f_2^\star(\cdot)$ at 0: $\forall\, \mathbf{u}^\star \in \mathcal{S}(w_1, w_2), \quad |f_2(\mathbf{u}^\star) - f_2^\star(0)| \leq |f_2^\star(g(w_1)) - f_2^\star(0)| \underset{w_1\to 1}{\longrightarrow} 0$.

We prove the second part of the lemma. Given a sequence $(w_1^{(n)}, w_2^{(n)}, \mathbf{u}_n) \in \mathcal{S}^{\mathbb{N}}$ such that $w_1^{(n)} \longrightarrow 1$, consider a convergent subsequence of it from compactness of $\mathcal{C}$. Its limit $(1, 0, \bar{\mathbf{u}})$ is such that $\bar{\mathbf{u}} \in \mathcal{C}$, $f_1(\bar{\mathbf{u}}) = 0$, and $f_2(\bar{\mathbf{u}}) = f_2^\star(0)$ from continuity of f_1 and f_2. Hence $\bar{u}$ is a solution to (7.43), which gives the second result of the lemma.

For the third part of the lemma, we start from the proof of the second part and note that any convergent subsequence $(\tilde{w}_1^{(n)}, \tilde{w}_2^{(n)}, \tilde{\mathbf{u}}_n)$ of $(w_1^{(n)}, w_2^{(n)}, \mathbf{u}_n)$ is such that $\tilde{\mathbf{u}}_n$ converges to the unique solution $\bar{\mathbf{u}}$ to (7.43). Hence any convergent subsequence has the same limit $(1, 0, \bar{u})$, and $(w_1^{(n)}, w_2^{(n)}, \mathbf{u}_n)$ thus converges to $(1, 0, \bar{u})$. Since this is true for any sequence $(w_1^{(n)}, w_2^{(n)}, \mathbf{u}_n) \in \mathcal{S}^{\mathbb{N}}$ such that $w_1^{(n)} \longrightarrow 1$, we have the third result of the lemma. □

Main results

To apply the results in Section 7.6 to our WSP, we substitute $\mathbf{u}$ with the tuple $(F(\cdot, \mathbf{p}), \{\mathbf{x}^{(j)}\}_j, \{\text{ by }^{(j)}\}_j)$ and the objectives (f_1, f_2) with $(r_{\text{obs}}, r_{\text{eq}})$. Since the feasible set in (7.34) is closed, compactness is guaranteed with this assumption:

Assumption 6.1. *The variables $(F(\cdot, \boldsymbol{p}), \{\boldsymbol{x}^{(j)}\}_j, \{\text{ by }^{(j)}\}_j)$ of the WSP (7.34) are in a finite-dimensional bounded set.*

The finite dimension assumption is reasonable since restricting the map $F(\cdot, \mathbf{p})$ to a finite dimensional affine parametrization $\sum_{i=1}^K a_i F_i(\cdot, \mathbf{p})$ is intuitive (as in [93, 20]). The boundedness is essential since the primal variables $\mathbf{x}^{(j)}$, dual variables by $^{(j)}$, and the parameters $\mathbf{a}$ have physical interpretations in terms of resource allocation, resource valuation, and variations of the map $F(\cdot, \mathbf{p})$ respectively, and thus are restricted to physically (or economically) reasonable ranges. Hence Assumption 6.1 is reasonable and guarantees compactness of the set of feasible variables of the WSP, and enables to apply the results in Section 7.6. From compactness, the minimal objective value $r_{\text{eq}}^\star$ of the Inverse VI (7.27), and the minimal objective value $r_{\text{obs}}^\star$ of the BP (7.32) are also attained.

Theorem 7.6. *Under Assumption 6.1, given N approximate solutions $\boldsymbol{z}^{(j)} \in \mathcal{K}(\boldsymbol{p}^{(j)})$ to the problems VI$(\mathcal{K}(\boldsymbol{p}^{(j)}), F(\cdot, \boldsymbol{p}^{(j)}))$ for $j = 1, \cdots, N$, any optimal solution to the WSP (7.34) is such that $r_{obs} \leq r_{eq}^\star(w_{obs}^{-1} - 1)$ and $r_{eq} \leq r_{eq}^\star$. In addition, r_{eq} converges uniformly to $r_{eq}^\star$ as $w_{obs} \longrightarrow 1$ and there exists a sequence of solutions to the WSP converging to a solution to the inverse VI (7.27).*

Theorem 7.7. *Under Assumption 6.1, given N observations $\boldsymbol{z}^{(j)}$ in (7.24), any optimal solution to the WSP (7.34) is such that $r_{eq} \leq r^{\star}_{obs}(w_{eq}^{-1} - 1)$, $r_{obs} \leq r^{\star}_{obs}$, and In addition, r_{obs} converges uniformly to $r^{\star}_{obs}$ as $w_{eq} \longrightarrow 1$ and there exsits a sequence of solutions to the WSP converging to a solution to the BP (7.32).*

Finally, the objective r_{obs} in our WSP (7.34) can be generalized, thus our WSP can be seen as a smoothing method for general bilevel programs where the complementary condition $r_{\text{PD}} = 0$ is included in the objective in the form of a penalty function. Previous works have proposed smoothing methods via, *e.g.*, the perturbed Fischer-Burmeister function [53, §6.5] or a similar one [60], but our smoothing via residuals has a sub-optimality interpretation.

7.7 Comparison of the duality gap and the KKT residual

Given N observations $\mathbf{z}^{(j)}$, for $j = 1, \cdots, N$, let $(F(\cdot, \mathbf{p}), \{\mathbf{x}^{(j)}\}_j, \{\text{ by }^{(j)}\}_j)$ be an optimal solution to the WSP (7.34). Then r_{obs} in (7.26) measures how well $\mathbf{x}^{(j)}$ agree with the observations $\mathbf{z}^{(j)}$, while r_{eq} in (7.25) measures how well the imputed process $\text{VI}(\mathcal{K}(\mathbf{p}), F(\cdot, \mathbf{p}))$ explains the imputed decision vectors $\mathbf{x}^{(j)}$. If the imputed map $F(\cdot, \mathbf{p})$ admits a unique solution $\hat{\mathbf{x}}(\mathbf{p})$ for all $\mathbf{p}$ (*e.g.*, from strict monotonicity), then an alternative metric to r_{eq} is $\sum_{j=1}^{N} \|\mathbf{x}^{(j)} - \hat{\mathbf{x}}(\mathbf{p}^{(j)})\|$. If $F(\cdot, \mathbf{p})$ is strongly monotone with parameter c for all $\mathbf{p}$, from (7.13) and (7.20):

$$\|\mathbf{x}^{(j)} - \hat{\mathbf{x}}(\mathbf{p}^{(j)})\|_2 \leq \sqrt{r_{\text{PD}}(\mathbf{x}^{(j)},\ \text{by }^{(j)}, \mathbf{p}^{(j)})/c} \quad \forall\, j \tag{7.50}$$

where $r_{\text{PD}}(\mathbf{x}^{(j)},\ \text{by }^{(j)}, \mathbf{p}^{(j)})$, $j = 1, \cdots, N$ are directly available from the WSP. Note that with only strict convexity of $F(\cdot, \mathbf{p})$, we can have $\|\mathbf{x}^{(j)} - \hat{\mathbf{x}}(\mathbf{p}^{(j)})\|_2 = \delta$ while $\sqrt{r_{\text{PD}}(\mathbf{x}^{(j)},\ \text{by }^{(j)}, \mathbf{p}^{(j)})}$ is infinitely small, as shown at the end of Section 7.2.

However, there is no result of the form $\|\mathbf{x} - \mathbf{x}^{\star}\| = \mathcal{O}(\sqrt{r_{\text{KKT}}(\mathbf{x},\ \text{by }, \boldsymbol{\pi})})$ to the best of our knowledge. We define the slack variables associated to the dual feasibility condition $\mathbf{A}^T \text{ by } \leq F(\mathbf{x})$:

$$\boldsymbol{\nu} := F(\mathbf{x}) - \mathbf{A}^T \text{ by} \tag{7.51}$$

which implies that dual feasibility is equivalent to $\boldsymbol{\nu} \geq 0$. We now derive a bound for the following generalized residuals:

$$r^{\ell_p}_{\text{PD}}(\mathbf{x}) = \|\boldsymbol{\nu} \circ \mathbf{x}\|_p = \left(\sum_{i=1}^{n} |\nu_i x_i|^p \right)^{1/p} \tag{7.52}$$

$$r^{\ell_p}_{\text{KKT}}(\mathbf{x},\ \text{by }, \boldsymbol{\pi}) = \left\| \begin{bmatrix} \alpha(\boldsymbol{\nu} - \boldsymbol{\pi}) \\ \mathbf{x} \circ \boldsymbol{\pi} \end{bmatrix} \right\|_p = \left(\sum_{i=1}^{n} \alpha^p |\nu_i - \pi_i|^p + |x_i \pi_i|^p \right)^{1/p} \tag{7.53}$$

where $\|x\|_p$ is the p-norm for $p \geq 1$, and $\mathbf{u} \circ \mathbf{v} = [u_i v_i]_{i=1}^n$ for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$. Since $\|\mathbf{x}\|_p \leq \|\mathbf{x}\|_1 \leq n^{1-1/p}\|\mathbf{x}\|_p$ for all $\mathbf{x} \in \mathbb{R}^n$, we have

$$r_{\mathrm{PD}}^{\ell_p}(\mathbf{x},\ \text{by}\ ) \leq r_{\mathrm{PD}}^{\ell_1}(\mathbf{x},\ \text{by}\ ) \leq n^{1-1/p} \cdot r_{\mathrm{PD}}^{\ell_p}(\mathbf{x},\ \text{by}\ ) \tag{7.54}$$

$$r_{\mathrm{KKT}}^{\ell_p}(\mathbf{x},\ \text{by}\ , \boldsymbol{\pi}) \leq r_{\mathrm{KKT}}^{\ell_1}(\mathbf{x},\ \text{by}\ , \boldsymbol{\pi}) \leq n^{1-1/p} \cdot r_{\mathrm{KKT}}^{\ell_p}(\mathbf{x},\ \text{by}\ , \boldsymbol{\pi}) \tag{7.55}$$

When primal and dual feasibilities hold, *i.e.* $\boldsymbol{\nu} \geq 0$, $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq 0$, we note that $r_{\mathrm{PD}}^{\ell_1}$, $r_{\mathrm{KKT}}^{\ell_1}$ defined above correspond to r_{PD}, r_{KKT} in (7.9), (7.8) since, for $r_{\mathrm{PD}}^{\ell_1}$:

$$r_{\mathrm{PD}}^{\ell_1}(\mathbf{x},\ \text{by}\ ) = \sum_{i=1}^n \nu_i x_i = \boldsymbol{\nu}^T\mathbf{x} = (F(\mathbf{x}) - \mathbf{A}^T\ \text{by}\ )^T\mathbf{x} = F(\mathbf{x})^T\mathbf{x} - \mathbf{b}^T\ \text{by} \tag{7.56}$$

The results in Section 7.2 thus hold for $r_{\mathrm{PD}}^{\ell_p}$ and $r_{\mathrm{KKT}}^{\ell_p}$ with an additional $n^{1-1/p}$ factor, validating them as residuals for the primal-dual and KKT systems respectively. Before stating our main result of the section, we present a lemma:

Lemma 7.3. *Let $\mathcal{K}$ be a polyhedron given by (7.2). Then the following holds for any $\alpha > 0$, $p > 1$, $\boldsymbol{x} \in \mathcal{K}$, $\text{by}\ \in \mathbb{R}^n$ such that $\boldsymbol{A}^T\ \text{by}\ \leq F(\boldsymbol{x})$:*

$$\min_{\boldsymbol{\pi} \geq 0} r_{KKT}^{\ell_p}(\boldsymbol{x},\ \text{by}\ , \boldsymbol{\pi}) = \left(\sum_{i=1}^n \frac{(\nu_i x_i)^p}{\left(1 + (x_i/\alpha)^{\frac{p}{p-1}}\right)^{p-1}} \right)^{1/p} \tag{7.57}$$

If $p = 1$, then for any $\alpha > 0$, $\boldsymbol{x} \in \mathcal{K}$, $\text{by}\ \in \mathbb{R}^n$ such that $\boldsymbol{A}^T\ \text{by}\ \leq F(\boldsymbol{x})$, we have:

$$\min_{\boldsymbol{\pi} \geq 0} r_{KKT}^{\ell_1}(\boldsymbol{x},\ \text{by}\ , \boldsymbol{\pi}) = \sum_{i\,:\,x_i < \alpha} x_i \nu_i + \sum_{i\,:\,x_i > \alpha} \alpha \nu_i \tag{7.58}$$

Proof. For any $p \geq 1$, $\mathbf{x} \in \mathbb{R}^n$, and $\text{by}\ \in \mathbb{R}^n$:

$$\min_{\boldsymbol{\pi} \geq 0} \left(r_{\mathrm{KKT}}^{\ell_p}(\mathbf{x},\ \text{by}\ , \boldsymbol{\pi}) \right)^p = \min_{\boldsymbol{\pi} \geq 0} \sum_{i=1}^n \alpha^p |\nu_i - \pi_i|^p + |x_i \pi_i|^p \tag{7.59}$$

$$= \sum_{i=1}^n \min_{\pi_i \geq 0} \left\{ \alpha^p |\nu_i - \pi_i|^p + |x_i \pi_i|^p \right\} \tag{7.60}$$

When primal and dual feasibilities hold, $\mathbf{x} \geq 0$, $\boldsymbol{\nu} \geq 0$, which causes the map $\pi_i \geq 0 \mapsto \alpha^p |\nu_i - \pi_i|^p + |x_i \pi_i|^p$ to increase on $[\nu_i, +\infty)$ and thus to attain its minimum on $[0, \nu_i]$, on which it is also differentiable, for all $p \geq 1$, with gradient:

$$\pi_i \mapsto -p\alpha^p (\nu_i - \pi_i)^{p-1} + p x_i^p \pi_i^{p-1}, \quad i = 1, \cdots, n \tag{7.61}$$

When $p > 1$, the gradient vanishes at a unique point $\pi_i^\star$ in $[0, \nu_i]$:

$$\pi_i^\star = \frac{\nu_i}{1 + (x_i/\alpha)^{p/(p-1)}}, \quad i = 1, \cdots, n \tag{7.62}$$

Substituting in (7.60):

$$\min_{\pi_i \geq 0} \{\alpha^p |\nu_i - \pi_i|^p + |x_i \pi_i|^p\} = \frac{(\nu_i x_i)^p}{(1 + (x_i/\alpha)^{p/(p-1)})^{p-1}} \tag{7.63}$$

which gives the desired result for $p > 1$.

When $p = 1$, the map $\pi_i \geq 0 \mapsto \alpha|\nu_i - \pi_i| + |x_i \pi_i|$ is just affine on $[0, \nu_i]$, in which the minimum is, and thus attains it minimum at 0 if $x_i - \alpha \geq 0$, and ν_i if $x_i - \alpha < 0$. Hence:

$$\sum_i \min_{\pi_i \geq 0} \{\alpha|\nu_i - \pi_i| + |x_i \pi_i|\} = \sum_{i\,:\,x_i < \alpha} x_i \nu_i + \sum_{i\,:\,x_i > \alpha} \alpha \nu_i \tag{7.64}$$

which completes the proof. □

We are now present the main result of the section, where $\|\mathbf{x}\|_\infty = \max_i |x_i|$:

Theorem 7.8. *Let $\mathcal{K}$ be a polyhedron given by (7.2). Then the following holds for any $\alpha > 0$, $p \geq 1$, $\epsilon > 0$, $\boldsymbol{x} \in \mathcal{K}$, $\boldsymbol{by} \in \mathbb{R}^n$ such that $\boldsymbol{A}^T \boldsymbol{by} \leq F(\boldsymbol{x})$:*

$$r_{PD}^{\ell_p}(\boldsymbol{x}, \boldsymbol{by}) \leq \epsilon \quad \Longrightarrow \quad \exists\, \boldsymbol{\pi} \in \mathbb{R}^n : r_{KKT}^{\ell_p}(\boldsymbol{x}, \boldsymbol{by}, \boldsymbol{\pi}) \leq \epsilon \tag{7.65}$$

Reciprocally, for $p > 1$, we have; for all $\epsilon > 0$, $\boldsymbol{x} \in \mathcal{K}$, $\boldsymbol{by} \in \mathbb{R}^n : \boldsymbol{A}^T \boldsymbol{by} \leq F(\boldsymbol{x})$:

$$\exists\, \boldsymbol{\pi} \in \mathbb{R}^n_+,\, r_{KKT}^{\ell_p}(\boldsymbol{x}, \boldsymbol{by}, \boldsymbol{\pi}) \leq \epsilon \implies r_{PD}^{\ell_p}(\boldsymbol{x}, \boldsymbol{by}) \leq \epsilon \left(1 + (\|\boldsymbol{x}\|_\infty/\alpha)^{\frac{p}{p-1}}\right)^{\frac{p-1}{p}} \tag{7.66}$$

When $p = 1$, we have; for all $\epsilon > 0$, $\boldsymbol{x} \in \mathcal{K}$, $\boldsymbol{by} \in \mathbb{R}^n$, and $\boldsymbol{A}^T \boldsymbol{by} \leq F(\boldsymbol{x})$:

$$\exists\, \boldsymbol{\pi} \in \mathbb{R}^n_+,\, r_{KKT}^{\ell_1}(\boldsymbol{x}, \boldsymbol{by}, \boldsymbol{\pi}) \leq \epsilon \implies r_{PD}^{\ell_1}(\boldsymbol{x}) \leq \epsilon \max\left(\|\boldsymbol{x}\|_\infty/\alpha,\, 1\right) \tag{7.67}$$

Proof. To prove (7.65) for $p > 1$, note that for all $\mathbf{x} \in \mathcal{K}$, by $\in \mathbb{R}^n$ such that $\mathbf{A}^T$ by $\leq F(\mathbf{x})$, each term $(\nu_i x_i)^p / \left(1 + (x_i/\alpha)^{\frac{p}{p-1}}\right)^{p-1}$ in $\min_{\boldsymbol{\pi} \geq 0} r_{\mathrm{KKT}}^{\ell_p}(\mathbf{x}, \text{ by }, \boldsymbol{\pi})$ given by (7.57) is less or equal than $|\nu_i x_i|^p$. Hence:

$$\min_{\boldsymbol{\pi} \geq 0} r_{\mathrm{KKT}}^{\ell_p}(\mathbf{x}, \text{ by }, \boldsymbol{\pi}) \leq r_{\mathrm{PD}}^{\ell_p}(\mathbf{x}, \text{ by }) \tag{7.68}$$

which proves (7.65) for $p > 1$. For $p = 1$, (7.65) is true since, from $\mathbf{x} \geq 0$ and $\boldsymbol{\nu} \geq 0$:

$$\min_{\boldsymbol{\pi} \geq 0} r_{\mathrm{KKT}}^{\ell_1}(\mathbf{x}, \text{ by }, \boldsymbol{\pi}) = \sum_{i\,:\,x_i < \alpha} x_i \nu_i + \sum_{i\,:\,x_i > \alpha} \alpha \nu_i \leq \sum_i x_i \nu_i = r_{\mathrm{PD}}^{\ell_1}(\mathbf{x}, \text{ by }) \tag{7.69}$$

To prove (7.66), we note that for all $\mathbf{x} \in \mathcal{K}$, by $\in \mathbb{R}^n$ such that $\mathbf{A}^T$ by $\leq F(\mathbf{x})$, each term $(\nu_i x_i)^p / \left(1 + (x_i/\alpha)^{\frac{p}{p-1}}\right)^{p-1}$ in $\min_{\boldsymbol{\pi} \geq 0} r^{\ell_p}_{\mathrm{KKT}}(\mathbf{x},$ by $, \boldsymbol{\pi})$ is greater or equal than $\frac{\nu_i x_i)^p}{\left(1+(\|\mathbf{x}\|_\infty/\alpha)^{\frac{p}{p-1}}\right)^{p-1}}$, hence, for all $\boldsymbol{\pi} \in \mathbb{R}^n$ such that $\boldsymbol{\pi} \geq 0$:

$$r^{\ell_p}_{\mathrm{KKT}}(\mathbf{x}, \text{ by }, \boldsymbol{\pi}) \geq \min_{\boldsymbol{\pi} \geq 0} r^{\ell_p}_{\mathrm{KKT}}(\mathbf{x}, \text{ by }, \boldsymbol{\pi}) \tag{7.70}$$

$$\geq \left(\sum_i (\nu_i x_i)^p / \left(1 + (\|\mathbf{x}\|_\infty/\alpha)^{\frac{p}{p-1}}\right)^{p-1} \right)^{1/p} \tag{7.71}$$

which proves (7.66) for $p > 1$. For $p = 1$, we have, with $(x)_+ = \max(x, 0)$:

$$r^{\ell_1}_{\mathrm{PD}}(\mathbf{x}, \text{ by }) - \min_{\boldsymbol{\pi} \geq 0} r^{\ell_1}_{\mathrm{KKT}}(\mathbf{x}, \text{ by }, \boldsymbol{\pi}) = \sum_{i\,:\,x_i > \alpha} (x_i - \alpha)\nu_i \tag{7.72}$$

$$\leq (\|\mathbf{x}\|_\infty - \alpha)_+ \sum_{i\,:\,x_i > \alpha} \nu_i \tag{7.73}$$

$$\leq \frac{(\|\mathbf{x}\|_\infty - \alpha)_+}{\alpha} \min_{\boldsymbol{\pi} \geq 0} r^{\ell_1}_{\mathrm{KKT}}(\mathbf{x}, \text{ by }, \boldsymbol{\pi}) \tag{7.74}$$

hence $r^{\ell_1}_{\mathrm{PD}}(\mathbf{x}, \text{ by }) \leq (1 + (\|\mathbf{x}\|_\infty/\alpha - 1)_+) \min_{\boldsymbol{\pi} \geq 0} r^{\ell_1}_{\mathrm{KKT}}(\mathbf{x}, \text{ by }, \boldsymbol{\pi})$. Finally, noting that $1 + (\|\mathbf{x}\|_\infty/\alpha - 1)_+ = \max(\|\mathbf{x}\|_\infty/\alpha, 1)$ completes the proof. □

The first bound (7.65) in Theorem 5.1. is tight since, using Lemma 5.1, we have $\min_{\boldsymbol{\pi} \geq 0} r^{\ell_p}_{\mathrm{KKT}}(\mathbf{x}, \text{ by }, \boldsymbol{\pi}) \longrightarrow r_{\mathrm{PD}}(\mathbf{x}, \text{ by })$ as $\alpha \longrightarrow +\infty$, for any $p \geq 1$. The bounds (7.66) and (7.67) are tight since we have equality in one dimension, *i.e.* $n = 1$. Combining (7.66), (7.54), (7.13) and (7.20) we have:

Theorem 7.9. *Let $\mathcal{K}$ be a polyhedron given by (7.2), and F be a strongly monotone function with parameter $c > 0$. Then VI($\mathcal{K}, F$) admits a unique solution $\boldsymbol{x}^\star$ and; for any $\alpha > 0$, $p > 1$, $\epsilon > 0$, $\boldsymbol{x} \in \mathcal{K}$:*

$$\begin{aligned} &\exists \text{ by }, \boldsymbol{\pi} \in \mathbb{R}^n : \boldsymbol{A}^T \text{ by } \leq F(\boldsymbol{x}), \boldsymbol{\pi} \geq 0, r^{\ell_p}_{KKT}(\boldsymbol{x}, \text{ by }, \boldsymbol{\pi}) \leq \epsilon \\ &\implies \|\boldsymbol{x} - \boldsymbol{x}^\star\|_2 \leq \sqrt{n^{1-1/p} \cdot \epsilon \left(1 + (\|\boldsymbol{x}\|_\infty/\alpha)^{\frac{p}{p-1}}\right)^{\frac{p-1}{p}} / c} \end{aligned} \tag{7.75}$$

For $p = 1$, the bound is $\|\boldsymbol{x} - \boldsymbol{x}^\star\|_2 \leq \sqrt{\epsilon \max(\|\boldsymbol{x}\|_\infty/\alpha, 1)/c}$.

7.8 Implementation

Affine parametrization

For tractability reasons, a classic approach consists in restricting the parametric map $F(\cdot, \mathbf{p})$ to be imputed to a finite dimensional affine parametric model

$$F(\cdot, \mathbf{p}) = F_0(\cdot, \mathbf{p}) + \sum_{i=1}^{K} a_i F_i(\cdot, \mathbf{p}), \quad \mathbf{a} \in \mathcal{A} \subseteq \mathbb{R}^K \tag{7.76}$$

where $F_i(\cdot, \mathbf{p})$, $i = 0, \cdots, K$ are pre-selected basis functions that typically contain prior knowledge on the candidate functions, and $\mathbf{a}$ is imputed in the set of allowable parameter vectors $\mathcal{A}$. For instance, if the true map $F^{\text{true}}(\cdot, \mathbf{p})$ is known to be increasing for all $\mathbf{p}$, then having a parameter space $\mathcal{A} \subseteq \mathbb{R}^K_+$ and increasing basis maps $F_i(\cdot, \mathbf{p})$ given any $(i, \mathbf{p})$ guarantees an increasing parametric map $F(\cdot, \mathbf{p})$ for all $\mathbf{a} \in \mathcal{A}$. In addition, the constant shift $F_0(\cdot, \mathbf{p})$ imposes a normalization on $F(\cdot, \mathbf{p})$ such that trivial solutions are excluded, *e.g.*, null maps where all of $\mathcal{K}$ is solution to the VI problem, and for which both non-negative objectives r_{eq} (7.25) and r_{obs} (7.26) can be minimized to zero.

A nonparametric estimation has also been considered in [20] using kernel methods and regularization methods from statistical learning. The methodology presented in the present chapter can also be extended to this approach.

Block-coordinate descent

Plugging in the affine parametrization (7.76) above, we solve the following WSP:

$$\begin{array}{ll} \min\limits_{\mathbf{a},\mathbf{x},\ \text{by}} & w_{\text{eq}} \Sigma_{j=1}^{N} r_{\text{PD}}(\mathbf{x}^{(j)},\ \text{by}^{(j)}, \mathbf{p}^{(j)} \,|\, \mathbf{a}) + w_{\text{obs}} \Sigma_{j=1}^{N} \phi(g(\mathbf{x}^{(j)}, \mathbf{p}^{(j)}) - \mathbf{z}^{(j)}) \\ \text{s.t.} & \mathbf{x}^{(j)} \in \mathcal{K}(\mathbf{p}^{(j)}), \quad \forall\, j \\ & \mathbf{A}(\mathbf{p}^{(j)})^T\ \text{by}^{(j)} \leq F(\mathbf{x}^{(j)}, \mathbf{p}^{(j)} \,|\, \mathbf{a}), \quad \forall\, j \\ & \mathbf{a} \in \mathcal{A} \end{array}$$

where the dependencies in $\mathbf{a}$ are made explicit. Since the size of the inverse problem increases linearly with the number of observations N, but is separable into N sub-problems with respect to the variables $\{\mathbf{x}^{(j)},\ \text{by}^{(j)}\}_{j=1,\cdots,N}$, we suggest to apply a Block-Coordinate Descent (BCD) algorithm to solve the WSP while avoiding the curse of dimensionality, see Algorithm 7.2. For the BCD, we cyclically update the N vectors $\{\mathbf{x}^{(j)}\}_{j=1,\cdots,N}$, the N vectors $\{\ \text{by}^{(j)}\}_{j=1,\cdots,N}$,

and the parameter vector $\mathbf{a}$. The sub-problems are:

$$\begin{array}{ll} \min\limits_{\mathbf{x}^{(j)}} & w_{\text{eq}}\, F(\mathbf{x}^{(j)}, \mathbf{p}^{(j)} \,|\, \mathbf{a})^T \mathbf{x}^{(j)} + w_{\text{obs}}\, \phi(g(\mathbf{x}^{(j)}, \mathbf{p}^{(j)}) - \mathbf{z}^{(j)}) \\ \text{s.t.} & \mathbf{x}^{(j)} \in \mathcal{K}(\mathbf{p}^{(j)}) \\ & \mathbf{A}(\mathbf{p}^{(j)})^T \text{ by }^{(j)} \leq F(\mathbf{x}^{(j)}, \mathbf{p}^{(j)} \,|\, \mathbf{a}) \end{array} \tag{7.77}$$

$$\min_{\text{by }^{(j)}} \; -\mathbf{b}^T \text{ by }^{(j)} \quad \text{s.t.} \quad \mathbf{A}(\mathbf{p}^{(j)})^T \text{ by }^{(j)} \leq F(\mathbf{x}^{(j)}, \mathbf{p}^{(j)} \,|\, \mathbf{a}) \tag{7.78}$$

$$\begin{array}{ll} \min\limits_{\mathbf{a} \in \mathcal{A}} & \sum_{i=1}^N F(\mathbf{x}^{(j)}, \mathbf{p}^{(j)} \,|\, \mathbf{a})^T \mathbf{x}^{(j)} \\ \text{s.t.} & \mathbf{A}(\mathbf{p}^{(j)})^T \text{ by }^{(j)} \leq F(\mathbf{x}^{(j)}, \mathbf{p}^{(j)} \,|\, \mathbf{a}), \quad \forall\, j \end{array} \tag{7.79}$$

We note that steps 3 and 4 in Algorithm 7.2 can be done in parallel.

Algorithm 7.2 `BCD(·)` Block descent algorithm for the inverse problem

1: **while** stopping criteria not met **do**
2: $t := t + 1$
3: $\mathbf{x}^{(j,t+1)} :=$ solution to (7.77) at $(\text{ by }^{(j)}, \mathbf{a}) = (\text{ by }^{(j,t)}, \mathbf{a}^{(t)})$ for $j = 1, \cdots, N$.
4: $\text{ by }^{(j,t+1)} :=$ solution to (7.78) at $(\mathbf{x}^{(j)}, \mathbf{a}) = (\mathbf{x}^{(j,t+1)}, \mathbf{a}^{(t)})$ for $j = 1, \cdots, N$.
5: $\mathbf{a}^{(t+1)} :=$ solution to (7.79) at $(\mathbf{x}^{(j)}, \text{ by }^{(j)}) = (\mathbf{x}^{(j,t+1)}, \text{ by }^{(j,t+1)})$ for all j.

7.9 Application to Traffic Assignment

Model

A classic application of VI and CO is the traffic assignment problem, see, *e.g.* [130] for more details. Given road network modeled as a directed graph $(\mathcal{V}, \mathcal{E})$, with vertex set $\mathcal{V}$ and directed edge set $\mathcal{E}$, and a set of commodities $\mathcal{W} \subseteq \mathcal{V} \times \mathcal{V}$, a flow rate d_k of a commodity k must be routed from s_k to t_k for each $k = (s_k, t_k) \in \mathcal{C}$. The k-th *commodity flow vector* $\mathbf{x}^{(k)} = [x_e^{(k)}]_{e \in \mathcal{E}} \in \mathbb{R}_+^{\mathcal{E}}$ is feasible if it satisfies the flow equation at every vertex $i \in \mathcal{V}$:

$$\sum_{j\,:\,(j,i) \in \mathcal{E}} x_{(j,i)}^{(k)} - \sum_{j\,:\,(i,j) \in \mathcal{E}} x_{(i,j)}^{(k)} = \begin{cases} -d_k & \text{if} \quad i = s_k \\ d_k & \text{if} \quad i = t_k \\ 0 & \text{otherwise} \end{cases} \tag{7.80}$$

In matrix form, $\mathbf{x}^{(k)}$ is feasible if $\mathbf{N}\mathbf{x}^{(k)} = \mathbf{b}^{(k)}$, $\mathbf{x}^{(k)} \geq 0$, where $\mathbf{N}$ is the node-arc incidence matrix and $\mathbf{b}^{(k)} \in \mathbb{R}^{\mathcal{V}}$ the demand vector associated to commodity k with entries such that $b_{s_k}^{(k)} = -d_k$, $b_{t_k}^{(k)} = d_k$, and $b_i^{(k)} = 0$, $\forall\, i \neq s_k, t_k$. Stacking everything together, we can simply rewrite the flow equations as $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq 0$, where $\mathbf{x} = [x_e^{(k)}]_{e \in \mathcal{E}, k \in \mathcal{C}}$ is the overall flow vector. Following [13], the cost $c_e(x_e)$ of a road segment e only depends on the flow x_e of vehicles on this segment, where x_e is expressed as $x_e = \sum_{k \in \mathcal{C}} x_e^{(k)}$, the sum of all the commodity

flows. The cost functions $c_e(\cdot)$ are assumed to be continuous, positive, non-decreasing, and Beckmann et al. [13] proved that the *User Equilibrium* (UE), defined by [165], exists and is solution to the CO($\mathcal{K}, f$) with potential:

$$f(\mathbf{x}) = \sum_{e \in \mathcal{E}} \int_0^{\sum_{k \in \mathcal{C}} x_e^{(k)}} c_e(u) du \tag{7.81}$$

However, cost functions c_e are in general unknown, other as through empirical modeling such as the BPR function, while total flows $x_e = \sum_{k \in \mathcal{C}} x_e^{(k)}$ are measurable, but only on a small subset of arcs in the network, due to the cost of deploying and maintaining a sensing infrastructure in a large urban area. With $g(\cdot)$ our fixed observation function (due to a fixed sensing infrastructure), we want to estimate delay functions from partial and noisy observations $\mathbf{z}^{(j)} = g(\mathbf{x}^{(j)}) + \mathbf{w}^{(j)}$ of flows $\mathbf{x}^{(j)}$ associated to different traffic demands $\mathbf{b}(\mathbf{p}^{(j)})$ and with noise $\mathbf{w}^{(j)}$, where each superscript j refers to different demand levels, *e.g.*, morning or evening commutes. The imputed delay functions can be used to control or re-design the road network. See Figure 7.2 for an example.

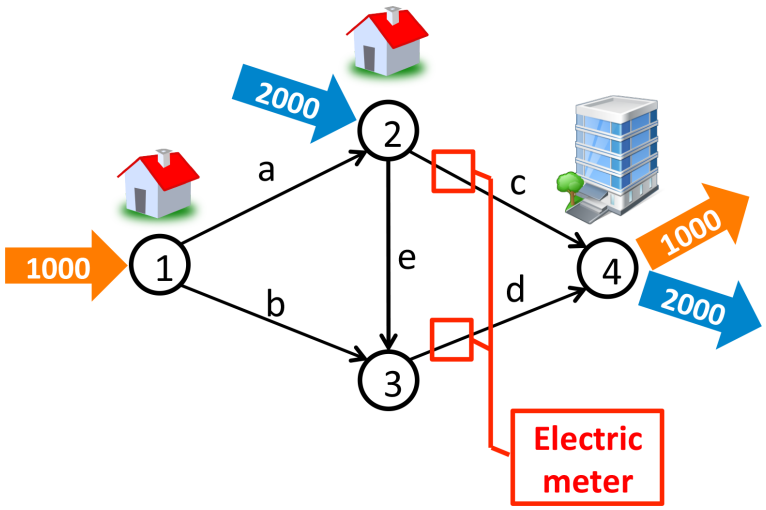

Figure 7.2: Example of a morning commute on a simple road network with arcs $\{a, b, c, d, e\}$, and two commodities 1 and 2 with commodity flows $x_a^{(k)}, x_b^{(k)}, x_c^{(k)}, x_d^{(k)}, x_e^{(k)}$, $k \in \{1, 2\}$. A flow of 1000 veh/hour in c_1 is known to be routed along the shortest paths from nodes 1 to 4 and a flow of 2000 veh/hour in commodity c_2 is routed from 2 to 4, resulting in a UE flow on the network. Given only measurement of $z_1 = (x_c^{(1)} + x_c^{(2)}) + w_1$ and $z_2 = (x_d^{(1)} + x_d^{(2)}) + w_2$ with noise w_1, w_2, how can we impute the delay functions on each arc?

Parametrization

We want to fit polynomial edge cost functions that are positive and non-decreasing. Hence we use the following parametrization, for all $e \in \mathcal{E}$:

$$c_e(x_e \,|\, \mathbf{a}) = d_e + d_e \sum_{i=1}^{K} a_i (x_e/m_e)^i, \quad \mathbf{a} = [a_i]_{i=1}^K \in \mathbb{R}_+^K \tag{7.82}$$

where m_e is the capacity of road segment e (typically proportional to the numer of lanes), and d_e is the known free-flow travel time. Here, d_e is the shift discussed in Section 7.8 to restrict the parameters a_i. The potential function f (which does not depend on the parameter $\mathbf{p}$) is then, using the expression in (7.81):

$$f(\mathbf{x} \,|\, \mathbf{a}) = f_0(\mathbf{x}) + \sum_{i=1}^{K} a_i f_i(\mathbf{x})$$

$$f_i(\mathbf{x}) = \sum_{e \in \mathcal{E}} \frac{d_e}{m_e^i} \int_0^{\sum_{k\in\mathcal{C}} x_e^{(k)}} u^i du = \sum_{e \in \mathcal{E}} \frac{d_e}{m_e^i} \frac{\left(\sum_{k\in\mathcal{C}} x_e^{(k)}\right)^{i+1}}{i+1} \quad i = 0, 1, \cdots, K$$

We are now in position to use our method with the basis map functions:

$$F_i(\mathbf{x}) = \nabla f_i(\mathbf{x}) = [\partial f_i(\mathbf{x})/\partial x_e^{(k)}]_{e\in\mathcal{E}, k\in\mathcal{C}} = \left[d_e \frac{\left(\sum_{k\in\mathcal{C}} x_e^{(k)}\right)^i}{m_e^i} \right]_{e\in\mathcal{E}, k\in\mathcal{C}} \tag{7.83}$$

Numerical experiments

We consider the highway network near Los Angeles with 44 nodes and 122 arcs; see Figure 7.3. The roads characteristics (geometry, capacity, free flow delay) are obtained from OpenStreetMaps. The OD demands $\mathbf{b}$ are based on data from the Census Bureau and calibrated to represent a static morning rush hour model. We consider $N = 4$ equilibria $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}, \mathbf{x}^{(4)}$ associated to four demand vectors $\mathbf{b}(\mathbf{p}^{(j)}) \in \mathbb{R}^{|\mathcal{C}||\mathcal{V}|}$, $j \in \{1, 2, 3, 4\}$ obtained by scaling $\mathbf{b}$ with respective factors .5, 0.8, 1, 1.2. The measurements are obtained by solving the traffic assignment problem:

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{b}(\mathbf{p}),\, \mathbf{x} \geq 0 \tag{7.84}$$

with potential function f given by (7.81), constraints $\mathbf{A}$ given by (7.80), demand vectors $\mathbf{b}(\mathbf{p}^{(j)})$, $j \in \{1, 2, 3, 4\}$, and for two types of delay functions:

$$c^{\text{poly}}(x_e) = d_e(1 + 0.15(x_e/m_e)^4) \tag{7.85}$$

$$c^{\text{hyper}}(x_e) = 1 - 3.5/3 + 3.5/(3 - x_e/m_e) \tag{7.86}$$

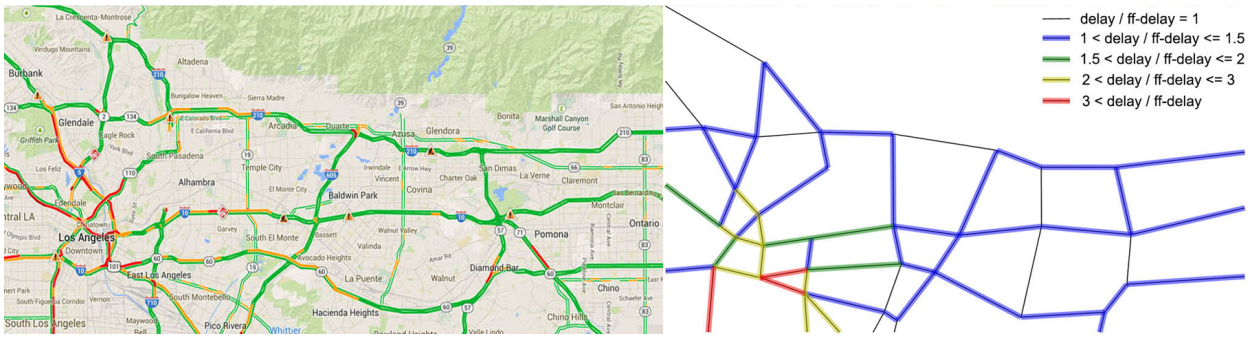

Figure 7.3: Left: Highway network of L.A. in morning rush hour on 2014-06-12 at 9:14 AM from Google Maps; right: The network in UE with the resulting delays under demand 1.2*b. The congested area is near central L.A.

where (7.85) is estimated by the Bureau of Public Roads (BPR), and (7.86) is hyperbolic delay similar to the BPR one. These two functions are considered the ground truth delay functions and we want to recover them from the observations $\mathbf{z}^{(j)} = g(\mathbf{x}^{(j)}) = [\sum_{k\in\mathcal{C}} x_e^{(k)}]_{e\in\mathcal{E}^{\text{obs}}}$, where $\mathcal{E}^{\text{obs}} \subseteq \mathcal{E}$ is the set of observed edge flows. We normalize r_{eq} and r_{obs} and solve the WSP (7.34) using the BCD algorithm discussed in Section 7.8. For $w_{\text{obs}} = 0.001$, 0.01, 0.1, 0.5, 0.9, 0.99, 0.999 and $w_{\text{eq}} = 1 - w_{\text{obs}}$, Figure 7.9 provides the error $\sum_{j=1}^{N} \|\mathbf{x}^{(j)} - \hat{\mathbf{x}}(\mathbf{p}^{(j)})\|$, where $\mathbf{x}^{(j)}$ are the ground-truth equilibrium flows and $\hat{\mathbf{x}}(\mathbf{p}^{(j)})$ the estimated ones.

In a second experiment, we study the sensitivity of our estimation algorithm to four sets of observed links, see Figure 7.5. The parameters $\mathbf{a}$ imputed by our latency inference methodology give a delay function $1 + \sum_{i=1}^{6} a_i x^i$ for each of the four sensor configurations. In case 1, we have a very good match between the estimated delay function and the true one because we observe the entire network, while in case 4, the measurements do not provide additional information because they are already contained in the given OD demands, see Figure 7.5.

7.10 Application to Consumer Utility

Model

We also consider an oligopoly in which n firms produce each one a product indexed by $i = 1, \cdots, n$ with prices $\mathbf{p} = [p_i]_{i=1}^n$. We suppose that the consumer purchases a quantity x_i of product i in order to maximize a non-decreasing and concave utility function $U(\mathbf{x})$ minus the price paid $\mathbf{p}^T\mathbf{x}$, where $\mathbf{x} = [x_i]_{i=1}^n$ is the overall demand, hence the optimization problem

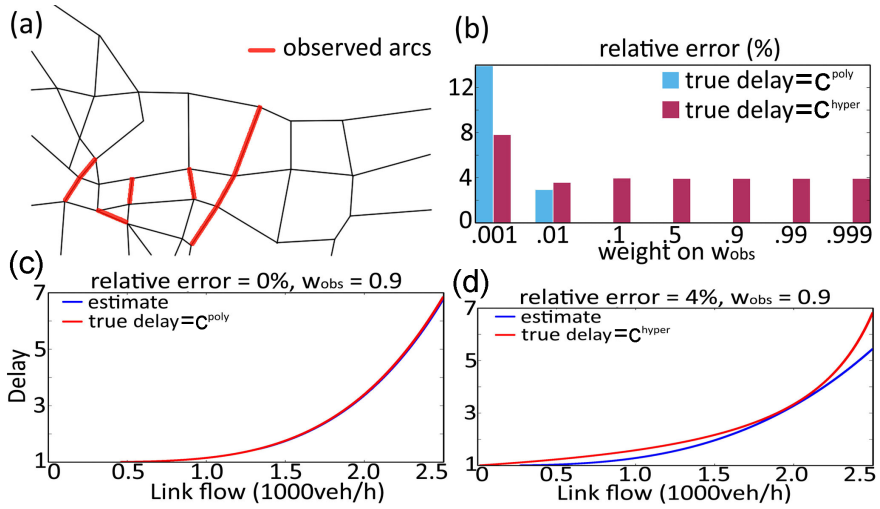

Figure 7.4: Imputation of the delay maps c^{poly}, c^{hyper} with parametric map given by (7.82). The relative error on the flow predicted by the imputed map is small for w_{obs} large enough as shown in (b). With accurate measurements, we suggest to solve the WSP with $w_{\text{obs}} = 0.9$, which gives the estimated cost function for the BPR cost function in (c) and hyperbolic cost function in (d).

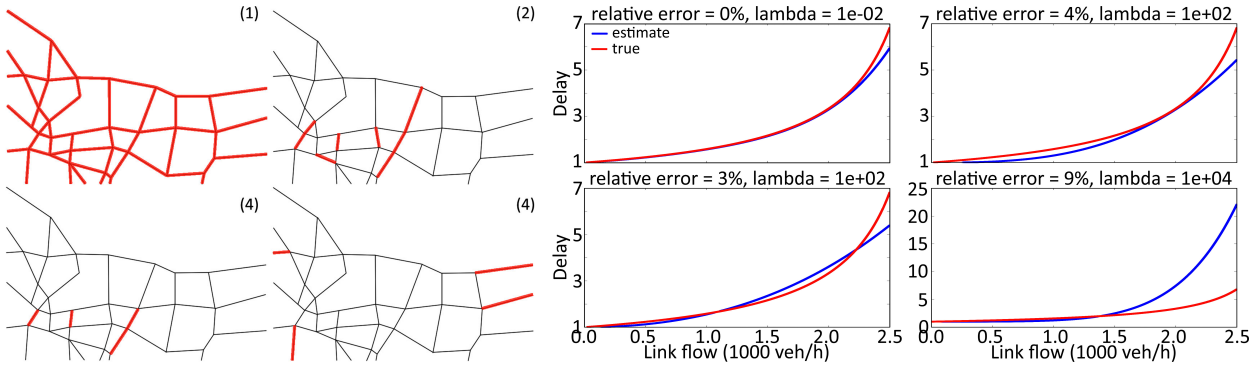

Figure 7.5: Left: the 4 sensor configurations: (1) all arcs are observed; (2) 10 arcs are observed in the congested area; (3) 4 arcs are observed in the congested area; (4) 4 arcs are observed at the boundaries of the region, where the inflows are already known from the OD demands.

and parametric map:

$$\min_{\mathbf{x}\geq 0} f(\mathbf{x}) = \mathbf{p}^T\mathbf{x} - U(\mathbf{x}) \quad \Longrightarrow \quad F(\mathbf{x},\mathbf{p}) = \mathbf{p} - \nabla U(\mathbf{x}) \tag{7.87}$$

However, the utility $U : \mathbb{R}^n \to \mathbb{R}$ is not known in practice, and the inverse problem consists in imputing U based on N observations of pairs $(\mathbf{p}^{(j)}, \mathbf{x}^{(j)})$, $j = 1, \cdots, N$ of prices and

associated demands. The imputed utility U is then used by company producing i to set a price p_i in order to achieve a target consumer demand x_i^{des} in its product. In oligopolies, the price of each product is publicly available and each firm in general knows its own demand x_i, however it may only have partial information on other demands. For example, if there are $n = 5$ firms and consumer demand in product 1 produced by firm 1 is not known, then we only observe the vector $\mathbf{z} = g(\mathbf{x}) = [x_2, x_3, x_4, x_5]^T$.

Parametrization

Similarly to [93], we consider a quadratic parametrization for the utility U, *i.e.* $U(\mathbf{x} \,|\, Q, \mathbf{r}) = \mathbf{x}^T Q \mathbf{x} + 2\mathbf{r}^T \mathbf{x}$, hence the parametric potential is

$$f(\mathbf{x}, \mathbf{p} \,|\, Q, \mathbf{r}) = \mathbf{p}^T \mathbf{x} - \left(\mathbf{x}^T Q \mathbf{x} + 2\mathbf{r}^T \mathbf{x}\right), \quad (Q, \mathbf{r}) \in \mathcal{A} \tag{7.88}$$

$$\mathcal{A} = \{(Q, \mathbf{r}) \,:\, Q\mathbf{x}_{\max} + \mathbf{r} \geq 0,\, \mathbf{r} \geq 0,\, Q \preceq 0\} \tag{7.89}$$

where $\mathcal{A}$ is chosen such that $U(\cdot \,|\, Q, \mathbf{r})$ is concave and non-decreasing on the demand range $[0, \mathbf{x}_{\max}]$. The parametric map $F(\cdot, \mathbf{p} \,|\, Q, \mathbf{r})$ is then:

$$F(\mathbf{x}, \mathbf{p} \,|\, Q, \mathbf{r}) = \mathbf{p} - 2Q\mathbf{x} - 2\mathbf{r} \tag{7.90}$$

Numerical experiments

We consider the case of $n = 5$ firms competing for the same market. At the time period j, let $\mathbf{x}^{(j)} \in \mathbb{R}^5_+$ be the consumer demand in response to the prices $\mathbf{p}^{(j)} \in \mathbb{R}^5_+$ set by each firm, sampled uniformly as i.i.d. random vectors in $[8, 12]^5$. We assume that the third firm only observes the demand $\mathbf{z}^{(j)} = [x_2^{(j)}, \cdots, x_5^{(j)}]^T$ in products from firms 2, 3, 4, 5 over $N = 200$ time periods along with the prices $\mathbf{p}^{(j)}$. The demand $\mathbf{x}^{(j)}$ incurred by prices $\mathbf{p}^{(j)}$ are assumed to be solution of the convex optimization model (7.87) with underlying consumer utility function $U^{\text{real}}(\mathbf{x}) = \mathbf{1}^T \sqrt{\mathbf{A}\mathbf{x} + \mathbf{b}}$. Firm 3 wants to impute U^{real} using the parametric utility given by (7.88). The numerical results are shown in Figure 7.6 with two models for $\mathbf{A} = 50(\mathbf{I} + \mathbf{B})$ in U^{real}: *model 1* where $\mathbf{B}_{ij}$ is sampled uniformly in $[0, 0.3]$ for $i \neq j$, and *model 2* where $\mathbf{B}_{ij}$ is sampled from $0.5 \cdot$Bernoulli(0.3).

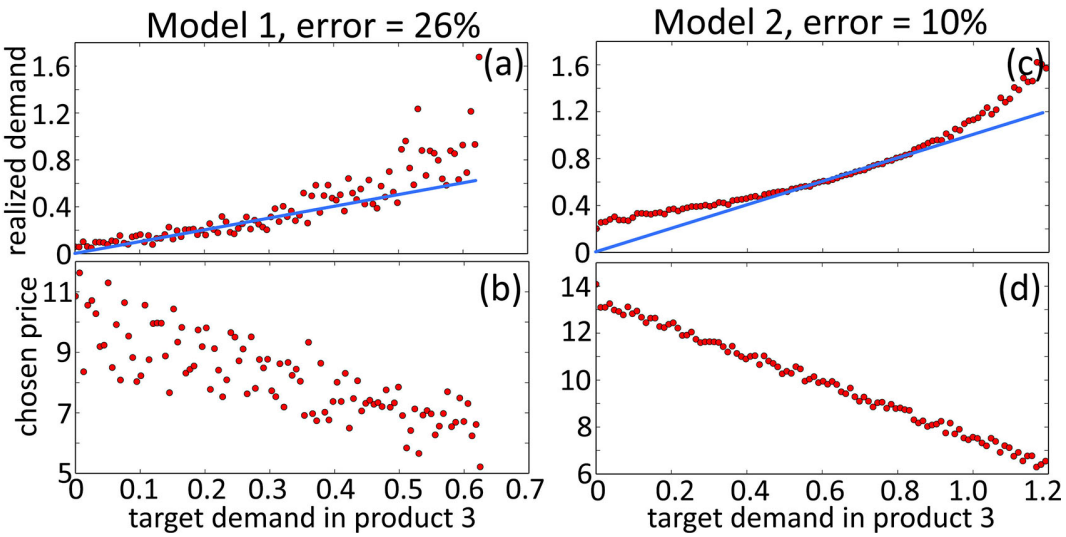

Figure 7.6: Use of the imputed utility to price product 3 for different target demands x_3^{des}. In (b), the prices are scattered due to correlations with other prices in model 1, while in (d), the prices vary linearly with x_3^{des} since the prices in model 2 are more uncorrelated. In (a), (c) the blue line is the $x = y$ line. For both models, the imputed utility performs well with relative errors of 26% and 10% on the training data and target demands $\mathbf{x}_3^{\text{des}}$ close to realized demands $\mathbf{x}_3^{\text{real}}$.

Chapter 8

Statistical learning of an equilibrium: approximation and concentration bounds

8.1 Introduction

In supervised learning, a random response vector $\mathbf{y} \in \mathbb{R}^m$ is generally modeled as an explicit function of the random predictor vector $\mathbf{p}$ in $\mathcal{P}$ (the predictor space). This includes linear and logit models, random forests, neural networks, see *e.g.* [78] for an overview of classic methods. In the present work, we are interested in training and validating a model such that the conditional expectation of $\mathbf{y}$ is an implicit function of $\mathbf{p}$. We focus here on the case when the implicit function is defined as an observation of a solution to a *variational inequality problem VIP(**p**)*, parametrized by the random predictor vector $\mathbf{p}$. Namely, VIP($\mathbf{p}$) consists in finding $\mathbf{x}^\star \in \mathcal{D}(\mathbf{p})$ such that

$$\langle F(\mathbf{x}^\star, \mathbf{p}),\, \mathbf{x} - \mathbf{x}^\star \rangle \geq 0, \quad \forall\, \mathbf{x} \in \mathcal{D}(\mathbf{p}) \tag{8.1}$$

where, for each $\mathbf{p} \in \mathcal{P}$, the parametric domain $\mathcal{D}(\mathbf{p})$ is a compact convex subset of $\mathbb{R}^n$, and $F(\cdot, \mathbf{p})$ is a continuous mapping from $\mathbb{R}^n$ to itself that is strongly monotone with parameter c. Compactness is sufficient for existence, and strong monotonicity is sufficient for uniqueness, of a solution to VIP($\mathbf{p}$). See Appendix B and Appendix C for details. Denoting the solution $\mathbf{x}^\star(\mathbf{p})$, which describes the equilibrium state, the model has thus the form

$$\mathbf{y} = h(\mathbf{x}^\star(\mathbf{p})) + \boldsymbol{\epsilon} \tag{8.2}$$

where h is an observation model continuously mapping from the state space $\mathbb{R}^n$ to an observed space in $\mathbb{R}^m$, and the noise $\boldsymbol{\epsilon} \in \mathbb{R}^m$ is a vector of independent Gaussian random variables with zero mean and variance at most σ^2.

Assuming we know h and $\mathcal{D}(\mathbf{p})$, *i.e.* the structure of the model, and we have available a set of measurements $(\mathbf{p}_i, \mathbf{y}_i)$, $i \in [N]$, *i.e.* the training data, where $[N]$ denotes the set $\{1, \cdots, N\}$, we want to train a parametric map $F(\cdot, \mathbf{p})$ such that each $h(\mathbf{x}^\star(\mathbf{p}_i))$ predicts $\mathbf{y}_i$, and we want to estimate the accuracy of the trained model. The main difficulty stems

from the bilevel structure of the resulting empirical risk minimization problem: the optimal solutions $\mathbf{x}^\star(\mathbf{p}_i)$ are unknown in most problems of interest. However, the approximation error of any feasible candidate $\hat{\mathbf{x}}_i \in \mathcal{D}(\mathbf{p}_i)$ can be bounded using gap functions $g(\hat{\mathbf{x}}_i, \mathbf{p}_i)$ which are related to the VIP($\mathbf{p}_i$) (8.1), and which are often easy to compute [87, §2].

Motivation. The problem of estimating the map F in the VIP($\mathbf{p}$) (8.1) is a very practical one since it emerges in many fields. In control, we may desire to fit a lower complexity controller (one that is easier to automate) from observing outputs of a sophisticated one available through, *e.g.* a human expert or model predictive control [164, 94]. In economics, consumer's purchases are modeled such that they maximize a utility function representing the satisfaction from one's purchases. This function is in general unknown to both the economist and the consumer, but can be learned by observing consumer purchases in response to price changes [94]. In transportation science, selfish routing games have been extensively studied, see [131]. Such models enable to study drivers' routing decisions in a network modeled as a directed graph, in which traveling each edge incurs a cost. Estimating the edge cost functions is a challenging task since they may represent some combination of the actual travel time, the tolls, and disutility from environmental factors, which are not directly observable. In practice, it is often possible to observe, through the sensing infrastructure, the equilibrium flows induced by the selfish routing of agents, and learn the underlying cost functions, see [19, 150]. More generally, many processes involve agents that behave optimally with respect to utility functions, and thus can be modeled as a VIP [61] or a *convex optimization problem (COP)*, which is a well-known specialization of the VIP [27, §4.2.3.]. [85] and [19] use the VIP or COP framework to learn the utility functions in *Nash equilibrium problems*.

Modeling real-world processes as lower complexity VIPs or COPs is a common practice as it enables to leverage powerful mathematical tools for the study of such processes. For example, in economics, knowing the consumer utility function enables to adjust prices to achieve some demand level [94]. In many cases in control, a low complexity controller requires less computation for little performance loss [94, 163]. In transportation science, the selfish behavior of agents (from shorted path routing) leads to an aggregate cost in the network worse than the system's optimum, and which can be analytically quantified [137, 46]. Taxation schemes can be designed to incentivize system optimal drivers' decisions [65, 91].

However, low complexity models rely upon having an accurate approximation of the real ones. For example, system mischaracterizations in selfish routing can cause taxes designed for one problem instance to incentivize inefficient behavior on different, yet closely-related instances [30]. Hence, we want to be able to measure the quality of the learned model. In the present paper, we present a statistical framework for the fitting of equilibrium models using the (standard) *empirical risk minimization* principle, by choosing the fit giving the lowest expected loss (the distance between the observed and predicted outputs) under the empirical measure. Hence, for the class of implicit models (8.2), it is critical to be able to have theoretical guarantees on the quality of the fit. While we discuss the optimization problem for the learning process in Section 8.5, which is adressed in more detail in [94, 19, 150], our main focus is a discussion on the proposed statistical learning framework for the important class of implicit models (8.2), and a *consistency* analysis of the learning problem.

Related work. [19] assume that direct measurements $\hat{\mathbf{x}}_i$ of the states $\mathbf{x}^\star(\mathbf{p}_i)$ are available, *i.e.* the observation mapping h in (8.2) is the identity function, and propose to learn $F(\cdot, \mathbf{p})$ from $(\mathbf{p}_i, \hat{\mathbf{x}}_i)$, $i \in [N]$ such that each $\hat{\mathbf{x}}_i$ is suboptimal for VIP($\mathbf{p}_i$). Good approximation quality is obtained by minimizing gap certificates from optimization. [150] consider the same model as in (8.2) and use some feasible state $\hat{\mathbf{x}}_i \in \mathcal{D}(\mathbf{p}_i)$ as an approximation for $\mathbf{x}^\star(\mathbf{p}_i)$. They propose to train the model such that they have simultaneously small gap certificates, $\hat{\mathbf{x}}_i$ is a good proxy for $\mathbf{x}^\star(\mathbf{p}_i)$, and small residuals $h(\hat{\mathbf{x}}_i) - \mathbf{y}_i$. The resulting trade-off between approximation quality and small observation residuals leads them to use Pareto optimization.

In the particular case when, for each $\mathbf{p} \in \mathcal{P}$, the mapping $F(\cdot, \mathbf{p})$ is the gradient $\nabla_{\mathbf{x}} f$ of a continuously differentiable potential $f(\cdot, \mathbf{p})$, then the parametric variational inequality problem VIP($\mathbf{p}$) reduces to finding $\mathbf{x}^\star \in \mathcal{D}(\mathbf{p})$ such that

$$\langle \nabla_{\mathbf{x}} f(\mathbf{x}^\star, \mathbf{p}),\, \mathbf{x} - \mathbf{x}^\star \rangle \geq 0, \quad \forall\, \mathbf{x} \in \mathcal{D}(\mathbf{p}) \tag{8.3}$$

which means that the feasible region only lies in the half-space where the potential $f(\cdot, \mathbf{p})$ increases. It turns out that monotonicity of $\nabla_{\mathbf{x}} f(\cdot, \mathbf{p})$ is equivalent to convexity of $f(\cdot, \mathbf{p})$ (see Appendix A), and condition (8.3) is the first-order optimality condition for the parametric constrained convex optimization program

$$\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{p}) \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{D}(\mathbf{p}) \tag{8.4}$$

see [27, §4.2.3.] and Appendix B. Thus, learning $F = \nabla_{\mathbf{x}} f$ allows one to learn the potential f by integrating F. The problem of imputing a convex objective is investigated by [94]. They also assume that direct measurements $\hat{\mathbf{x}}_i$ of the states $\mathbf{x}^\star(\mathbf{p}_i)$ are available and learn $f(\cdot, \mathbf{p})$ from $(\mathbf{p}_i, \hat{\mathbf{x}}_i)$, $i \in [N]$ such that each $\hat{\mathbf{x}}_i$ is suboptimal for the convex problem (8.4) at $\mathbf{p}_i$. They measure the approximation quality by using certificates defined as residuals of the Kharush-Kuhn-Tucker (KKT) conditions. To avoid the curse of dimensionality, they search f in an available set of candidates of the form $\{f_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p}) : \boldsymbol{\theta} \in \Theta\}$, where the parameter vector $\boldsymbol{\theta}$, which controls how f depends on the state $\mathbf{x}$ and predictor $\mathbf{p}$, needs to be estimated. [85] use similar ideas but from only one sample $(\mathbf{p}_1, \hat{\mathbf{x}}_1)$. If there exists several $\boldsymbol{\theta}$'s such that $\hat{\mathbf{x}}_1$ satisfies the KKT conditions associated to (8.4) at $\mathbf{p}_1$, then $\boldsymbol{\theta}$ is chosen to be close to a nominal one.

The problem of learning value functions from observations of predictors and responses are also considered in other disciplines. For example, [123, 1, 135] study the inverse reinforcement learning problem where the goal is to learn the reward function from observing the optimal policy, made available though an expert's behavior. Related to this, [167] present a method to learn the Q-values, which are expected discounted rewards for executing actions in a controlled Markov process, from observing the state and action over time.

Contributions. To the best of our knowledge, when addressing the problem of learning the function involved in a variational inequality or convex problem, none of the existing work has considered it in a statistical framework. Thus, our contributions are two-fold: First, the complex bilevel form of the empirical risk minimization problem requires to approximate each equilibrium point $\mathbf{x}^\star(\mathbf{p}_i)$ by a feasible state $\hat{\mathbf{x}}_i$, where sub-optimality certificates are obtained

from optimization theory. By bounding the difference between the *approximate empirical risk (AER)* and the empirical risk, we show that small gap certificates guarantee that the AER is a good proxy for the empirical risk. When there is noise or just one outlier, we show that the method of [94, 19], focusing on a zero AER to the detriment of large gap certificates, can lead to high empirical risks, or poor predictions.

Second, we leverage the proposed statistical framework to derive powerful results on the properties for this class of supervised learning problems. We use results on Lipschitz functions of Gaussian variables to show that small noise variance and good approximation accuracy are sufficient for the AER to concentrate close to the *error-of-fit*, defined as the mean error between the trained model and the true *unknown* one. The AER is thus a good t statistic for testing the quality of the trained model. Most importantly, we derive a sufficient condition on the approximation error and the noise variance for the power of the proposed test to be arbitrarily high if enough samples are available. Hence, our work provides powerful tools for the modeling and validation of complex real-world processes as equilibria.

8.2 Problem statement

Setting. We assume available a set of i.i.d. samples $\{\mathbf{p}_i\}_{i\in[N]}$ from a random predictor vector $\mathbf{p}$ in $\mathcal{P}$ (the predictor space), along with samples $\{\mathbf{y}_i\}_{i\in[N]}$ of the response $\mathbf{y} \in \mathbb{R}^m$. We are also given prior information on the structure of the model: a compact convex parametric domain $\mathcal{D}(\mathbf{p})$ in the state space $\mathbb{R}^n$ for each $\mathbf{p} \in \mathcal{P}$, and a continuous observation mapping h from $\mathbb{R}^n$ to the observed space $\mathbb{R}^m$. With $\mathbf{x}^\star(\mathbf{p})$ the solution to the parametric VIP($\mathbf{p}$) in (8.1), we assume the relationship between $\mathbf{p}_i$ and $\mathbf{y}_i$ to be

$$\mathbf{y}_i = h(\mathbf{x}^\star(\mathbf{p}_i)) + \boldsymbol{\epsilon}_i, \quad i \in [N] \tag{8.5}$$

where the $\boldsymbol{\epsilon}_i$ are i.i.d. samples from a random error vector $\boldsymbol{\epsilon} \in \mathbb{R}^m$. We denote the set of mappings from $\mathbb{R}^n \times \mathcal{P}$ to $\mathbb{R}^n$ by $\mathcal{M}(\mathbb{R}^n \times \mathcal{P},\, \mathbb{R}^n)$. To describe the supervised learning problem more concretely, we assume that the mapping F in (8.1) belongs to an indexed-family

$$\{F_{\boldsymbol{\theta}}(\cdot,\cdot) \,:\, \boldsymbol{\theta} \in \Theta\} \subset \mathcal{M}(\mathbb{R}^n \times \mathcal{P},\, \mathbb{R}^n) \tag{8.6}$$

where $F_{\boldsymbol{\theta}}(\cdot, \mathbf{p})$ is strongly monotone with parameter c for each $(\mathbf{p}, \boldsymbol{\theta}) \in \mathcal{P} \times \Theta$, and Θ is the set of allowable parameters. The parameters $\boldsymbol{\theta}$ control the shape of $F_{\boldsymbol{\theta}}(\cdot,\cdot)$ with respect to the state $\mathbf{x}$ and predictor $\mathbf{p}$.[1] Hence, the function $\mathbf{x}^\star(\cdot)$ belongs to an indexed-family of implicit functions $\{\mathbf{x}^\star_{\boldsymbol{\theta}}(\cdot) : \boldsymbol{\theta} \in \Theta\}$ such that for each $(\mathbf{p}, \boldsymbol{\theta}) \in \mathcal{P} \times \Theta$, $\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p})$ is the unique optimal solution to the VIP($\mathbf{p}$) (8.1) with parametric map $F_{\boldsymbol{\theta}}(\cdot, \mathbf{p})$, which we denote VIP$_{\boldsymbol{\theta}}(\mathbf{p})$.

[1] The parameter $\boldsymbol{\theta}$ can lie within a finite-dimensional space, *e.g.* $\Theta \subset \mathbb{R}^r$, or could lie within some function class, in which case the learning problem is non-parametric.

Goal. We pose the loss function as $\|\mathbf{y} - h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}))\|$, where $\|\cdot\|$ is any norm[2] in the observed space $\mathbb{R}^m$. Hence, the *empirical risk* is defined by

$$\mathcal{L}_N(\boldsymbol{\theta}) := \frac{1}{N}\sum_{i=1}^N \|\mathbf{y}_i - h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}_i))\| \tag{8.7}$$

We want to compute and validate $\hat{\boldsymbol{\theta}} \in \arg\min_{\boldsymbol{\theta}\in\Theta} \mathcal{L}_N(\boldsymbol{\theta})$. Section 5.2 presents an instantiation of the learning problem defined in (8.1), (8.5), (8.6), (8.7) in the case of selfish routing.

8.3 Applications

Routing games

$$\Delta(\mathbf{d}) := \left\{\boldsymbol{\mu} \in \mathbb{R}_+^{\mathcal{P}} : \textstyle\sum_{p\in\mathcal{P}_k} \mu_p^k = d_k,\ \forall\, k \in [K]\right\} \tag{8.8}$$

The path assignment determines the *edge flow* defined as $x_e = \sum_{k=1}^K \sum_{p\in\mathcal{P}_k : e\in p} \mu_p^k$, which can be written compactly as $x_e = (\mathbf{M}\boldsymbol{\mu})_e$ where $\mathbf{M} \in \mathbb{R}^{\mathcal{E}\times\mathcal{P}}$ is an incidence matrix with entries defined as $M_{e,p} = \mathbf{1}_{e\in p}$. For each edge e, the edge flow incurs a cost $c_e(x_e)$, and the cost of choosing a path p is the sum of edge costs along the path, *i.e.* $\sum_{e\in p} c_e(x_e)$. We define the mapping F as the vector of congestion functions

$$F\ :\ \mathbf{x} \in \mathbb{R}_+^{\mathcal{E}} \mapsto F(\mathbf{x}) = (c_e(x_e))_{e\in\mathcal{E}} \tag{8.9}$$

Equilibrium in routing games: Let us define the set $\mathcal{D}(\mathbf{d}) = \mathbf{M}\Delta(\mathbf{d}) = \{\mathbf{M}\boldsymbol{\mu}\ :\ \boldsymbol{\mu} \in \Delta(\mathbf{d})\} \subset \mathbb{R}_+^{\mathcal{E}}$ of feasible edge flows. We say that $\mathbf{x}^\star \in \mathcal{D}(\mathbf{d})$ is a Nash equilibrium if it satisfies the VIP

$$\langle F(\mathbf{x}^\star),\, \mathbf{x} - \mathbf{x}^\star\rangle \geq 0, \quad \forall\, \mathbf{x} \in \mathcal{D}(\mathbf{d}) \tag{8.10}$$

We refer to *e.g.* [131, §3.2] for a full technical discussion on Nash equilibria in routing games. Note that $\mathcal{D}(\mathbf{d})$ is compact convex since it is the image of $\mathbf{M}$ restricted to the compact convex set $\Delta(\mathbf{d})$. In this application, the demand vector $\mathbf{d}$ is the predictor and F is independent from it. Our parametric VIP thus consists in finding $\mathbf{x}^\star(\mathbf{d}) \in \mathcal{D}(\mathbf{d})$ such that (8.10) is satisfied, for a random population demand $\mathbf{d}$.

Learning the congestion functions: We assume available the capacity m_e and the base cost c_e^0 of each edge $e \in \mathcal{E}$ and we measure the edge flows on a subset $\mathcal{A} \subseteq \mathcal{E}$ of the edges, *i.e.* the observation mapping is the projection of $\mathbb{R}_+^{\mathcal{E}}$ into $\mathbb{R}_+^{\mathcal{A}}$ defined by $h\ :\ \mathbf{x} \mapsto (x_e)_{e\in\mathcal{A}}$. This allows us to fully specify model (8.2) instantiated to the routing game.

[2] If the error $\boldsymbol{\epsilon} \in \mathbb{R}^m$ is, *e.g.*, Gaussian with covariance $\boldsymbol{\Sigma} \succ 0$, then $\min \mathcal{L}_N(\boldsymbol{\theta})$ with the quadratic norm $\|\mathbf{z}\|_{\boldsymbol{\Sigma}} = \|\boldsymbol{\Sigma}^{-1/2}\mathbf{z}\|_2$ can be seen as a maximum likelihood problem, or weighted least squares that fixes heteroscedasticity.

Let us define the class of univariate functions

$$\Theta := \{f : \mathbb{R}_+ \to \mathbb{R}_+,\ L\text{-Lipschitz},\ c\text{-strong-monotone},\ f(0) = 0\}$$

We assume that $F(\mathbf{x}) = (c_e(x_e))_{e\in\mathcal{E}}$ belongs to the class

$$\mathcal{F} = \left\{\mathbf{x} \in \mathbb{R}_+^{\mathcal{E}} \mapsto (c_e^0 + f(\tfrac{x_e}{m_e}))_e \,|\, f \in \Theta\right\} \tag{8.11}$$

We suppose that the cost functions are invariant with respect to the normalized edge flows x_e/m_e on each edge e, which is a standard assumption in traffic modeling. The function class (8.11) gives us the indexed family (8.6). For each estimate $\hat{F} \in \mathcal{F}$ learned from samples of population demand and edge flows $\{(\mathbf{d}_i, \mathbf{y}_i)\}_{i\in[N]}$, we want to derive theoretical guarantees on the quality of the fit.

Usage. The learned edge costs $\hat{F}$ is used to quantify the inefficiency of equilibria in routing games [137, 46], and to design taxation schemes to incentivize system optimal decisions [65, 91], hence having an accurate estimate $\hat{F}$ is critical.

Consumer utility

Setting. We consider n products indexed by $i \in [n]$, with prices $\mathbf{p} = (p_i)_{i\in[n]}$ and demand $\mathbf{x} = (x_i)_{i\in[n]}$. Consumer purchases are assumed to solve the COP

$$\min\ \mathbf{p}^T\mathbf{x} - u(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{x} \in \mathbb{R}_+^n$$

where $u : \mathbb{R}_+^n \to \mathbb{R}$ is a concave and non-decreasing utility function that represents the consumer's satisfaction from its purchases. With $\mathcal{S}_-^n$ the set of negative semi-definite matrices of $\mathbb{R}^{n\times n}$, we learn u within the function class

$$\mathcal{F} := \{\mathbf{x} \in \mathbb{R}_+^n \mapsto \tfrac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{r}^T\mathbf{x} \,|\, (\mathbf{Q},\, \mathbf{r}) \in \Theta\}$$
$$\Theta := \{(\mathbf{Q},\, \mathbf{r}) \in \mathcal{S}_-^n \times \mathbb{R}_+^n \,|\, \mathbf{Q}\,\mathbf{x}_{\max} + \mathbf{r} \geq 0\}$$

where $\mathbf{x}_{\max} \in \mathbb{R}_+^n$ is the maximum demand vector. Hence we assume that the utility function is concave quadratic and is increasing, *i.e.* $\nabla u(\mathbf{x}) \geq 0$, for all $\mathbf{x} \in \mathbb{R}_+^n$ such that $\mathbf{x} \leq \mathbf{x}_{\max}$. In this application, the objective function depends the random predictors $\mathbf{p}$ (the prices), while the domain $\mathbb{R}_+^n$ is independent from $\mathbf{p}$. For each estimate $\hat{u} \in \mathcal{F}$, we want to say something about the quality of the fit.

Usage. The estimate $\hat{u}$ can be used to set prices $\mathbf{p}$ to achieve a target demand level $\mathbf{x}$, see [94]. Hence, having theoretical guarantees on the quality of the learned model is of great importance.

Controller fitting

Setting. We consider a dynamical system with state $\mathbf{x}_t \in \mathbb{R}^n$, input $\mathbf{u}_t \in \mathbb{R}^m$, and i.i.d. noise $\mathbf{w}_t \in \mathbb{R}^n$ at time t. The linear dynamics are $\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t$, $t \geq 0$. Given

a convex stage cost function $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, the stochastic control problem consists in finding a control policy $\{\mathbf{u}_t\}_{t\geq 0}$ that minimizes $\limsup_{T\to\infty} \frac{1}{T}\mathbb{E}\sum_{t=1}^{T-1} \ell(\mathbf{x}_t, \mathbf{u}_t)$ with the constraint $\mathbf{F}\,\mathbf{u}_t \leq \mathbf{h}$, $t \geq 0$, where $\mathbf{F} \in \mathbb{R}^{p\times m}$ and $\mathbf{h} \in \mathbb{R}^p$. We refer to [18, 94] for a full technical discussion on stochastic control.

Learning an approximate control. We are given samples of state-control (or input-output) pairs $\{(\mathbf{x}_i, \mathbf{u}_i)\}_{i\in[N]}$ from a suboptimal (but complex) control policy run by a human expert or a computationally expensive controller such as model predictive control [18, 164], and we want to learn a global *approximate value function* $v : \mathbb{R}^n \to \mathbb{R}$ that gives us a lower complexity controller via the optimization program

$$\min_{\mathbf{u}} \ell(\mathbf{x}, \mathbf{u}) + v(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}) \quad \text{s.t.} \quad \mathbf{F}\,\mathbf{u} \leq \mathbf{h} \tag{8.12}$$

The above control policy is known as the *approximate dynamic programming* policy [18] and a standard approach [94] is to learn v in the class $\mathcal{F} := \{\mathbf{z} \mapsto \mathbf{z}^T\mathbf{P}\mathbf{z} \,|\, \mathbf{P} \in \Theta\}$, where the index set Θ is the set $\mathcal{S}^n_+$ of positive semi-definite matrices of $\mathbb{R}^{n\times n}$.

Usage. We can use the program (8.12) with a value function estimate $\hat{v}$ to approximate a policy with a computationally efficient controller, see [94].

8.4 Approximation and risk bounds

Given samples $\{(\mathbf{p}_1, \mathbf{y}_1), \cdots, (\mathbf{p}_N, \mathbf{y}_N)\}$, the main difficulty stems from the *implicit* nature of $\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}_i)$ in the expression of $\mathcal{L}_N(\boldsymbol{\theta})$, because this quantity is not known in practice, since it is the solution of a variational inequality (or optimization) problem. Fortunately, $\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}_i)$ can be approximated by any feasible vector $\mathbf{x}_i \in \mathcal{D}(\mathbf{p}_i)$, and certificates on the approximation quality can be obtained from gap functions commonly used in the convex optimization and variational inequality literature [61, 87]. Examples of gap functions associated to the $\mathrm{VIP}_{\boldsymbol{\theta}}(\mathbf{p})$ include

$$g'_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p}) = \max_{\mathbf{z}\in\mathcal{D}(\mathbf{p})} \langle \mathbf{x} - \mathbf{z},\, F_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p})\rangle \tag{8.13}$$

$$g''_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p}) = \min_{\substack{\boldsymbol{\nu}\in\mathbb{R}^p_+: \\ \mathbf{A}(\mathbf{p})^T\boldsymbol{\nu}\leq F_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{p})}} F_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p})^T\mathbf{x} - \mathbf{b}(\mathbf{p})^T\boldsymbol{\nu} \tag{8.14}$$

where $\mathbf{x} \circ \boldsymbol{\pi} \in \mathbb{R}^n$ is the element-wise product of $\mathbf{x}$ and $\boldsymbol{\pi}$ (Hadamard product). For (8.14), we assume $\mathcal{D}(\mathbf{p})$ polyhedral $\mathcal{D}(\mathbf{p}) = \{\mathbf{x} \in \mathbb{R}^n_+ \,:\, \mathbf{A}(\mathbf{p})\mathbf{x} = \mathbf{b}(\mathbf{p})\}$, where $\mathbf{A}(\mathbf{p}) \in \mathbb{R}^{p\times n}$ and $\mathbf{b}(\mathbf{p}) \in \mathbb{R}^p$ for every predictor $\mathbf{p} \in \mathcal{P}$. Note that the above gap functions are often easy to compute. For example, the computation of (8.13) is a by-product of every iteration of the Frank-Wolfe Algorithm [68, 87] for solving the $\mathrm{VIP}_{\boldsymbol{\theta}}(\mathbf{p})$. Moreover, in most problems of interest, including all the applications presented in [85, 94, 19], the parametric domain $\mathcal{D}(\mathbf{p})$ is polyhedral, and so (8.13) and (8.14) are optimal values of a linear program.

Recall that $F_{\boldsymbol{\theta}}(\cdot, \mathbf{p})$ is strongly convex with parameter c. For each $\mathbf{x} \in \mathcal{D}(\mathbf{p})$, we have the *approximation bounds*

$$\|\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}) - \mathbf{x}\|_2^2 \leq g'_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p})/c = g''_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p})/c \tag{8.15}$$

$$\|\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}) - \mathbf{x}\|_2^2 \leq g'''_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p}) \max(1, \lambda(\mathbf{p}))/c \tag{8.16}$$

where $\lambda(\mathbf{p}) := \operatorname{diam}_{\|\cdot\|_\infty}(\mathcal{D}(\mathbf{p}))$. For completeness, we provide proofs of the above bounds in Appendix D. Thus, the gap functions measure the approximation quality of $\mathbf{x}$. For some increasing function $\phi : \mathbb{R}_+ \to \mathbb{R}_+$, we define a *generic gap function* $g_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p})$ that is ϕ composed with any of the three gaps defined above. To bound $\|\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}) - \mathbf{x}\|$, we choose $g_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p}) := \sqrt{g'_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p})} = \sqrt{g''_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p})}$ so that, for every $(\mathbf{p}, \boldsymbol{\theta}) \in \mathcal{P} \times \Theta$,

$$\|\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}) - \mathbf{x}\|_2 \leq g_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p})/\sqrt{c}, \quad \forall\, \mathbf{x} \in \mathcal{D}(\mathbf{p}) \tag{8.17}$$

Given training data $\{(\mathbf{p}_1, \mathbf{y}_1), \cdots, (\mathbf{p}_N, \mathbf{y}_N)\}$, for every set of feasible vectors $\{\mathbf{x}_i\}_{i\in[N]} \in \mathcal{D}(\mathbf{p}_1) \times \cdots \times \mathcal{D}(\mathbf{p}_N)$, we define the *approximate empirical risk (AER)* as

$$\Phi_N\left(\{\mathbf{x}_i\}_{i\in[N]}\right) := \frac{1}{N} \textstyle\sum_{i=1}^N \|\mathbf{y}_i - h(\mathbf{x}_i)\| \tag{8.18}$$

Using property (8.17) of $g_{\boldsymbol{\theta}}$, we can measure if the AER is good proxy for $\mathcal{L}_N(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N \|\mathbf{y}_i - h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}_i))\|$

Theorem 8.1. *If the observation map h is L-Lipschitz from the normed vector space $(\mathbb{R}^n, \|\cdot\|_2)$ to $(\mathbb{R}^m, \|\cdot\|)$, then $\left|\mathcal{L}_N(\boldsymbol{\theta}) - \Phi_N\left(\{\boldsymbol{x}_i\}_{i\in[N]}\right)\right| \leq \frac{L}{N\sqrt{c}} \sum_{i=1}^N g_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{p}_i)$ for every $\{\boldsymbol{x}_i\}_{i\in[N]} \in \mathcal{D}(\boldsymbol{p}_1) \times \cdots \times \mathcal{D}(\boldsymbol{p}_N)$.*

Proof. Since any vectors $\mathbf{a}, \mathbf{b}$ in a Hilbert space $\mathcal{H}$ satisfy $|\,\|\mathbf{a}+\mathbf{b}\| - \|\mathbf{a}\|\,| \leq \|\mathbf{b}\|$, applying the inequality with $\mathbf{a} = \mathbf{y}_i - h(\mathbf{x}_i)$ and $\mathbf{b} = h(\mathbf{x}_i) - h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}_i))$ gives $|\,\|\mathbf{y}_i - h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}_i))\| - \|\mathbf{y}_i - h(\mathbf{x}_i)\|\,| \leq \|h(\mathbf{x}_i) - h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}_i))\|$ where the right-hand side is bounded by $L\|\mathbf{x}_i - \mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}_i)\|_2 \leq \frac{L}{\sqrt{c}} g_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{p}_i)$. Summing over i, and applying the triangle inequality on the left-hand side gives the claimed bound. □

Hence, the AER is good proxy for $\mathcal{L}_N(\boldsymbol{\theta})$ if we have good certificate accuracy. For larger strong monotonicity parameters c, $F_{\boldsymbol{\theta}}(\cdot, \mathbf{p})$ is less flat thus easier to work with, so we get better bounds. For larger Lipschitz constants L, h has wider variations in the state space $\mathcal{D}(\mathbf{p})$, so we get larger bounds. Theorem 8.1 also gives an upper bound on $\mathcal{L}_N(\boldsymbol{\theta})$:

$$\mathcal{L}_N(\boldsymbol{\theta}) \leq \Phi_N\left(\{\mathbf{x}_i\}_i\right) + \frac{L}{N\sqrt{c}} \textstyle\sum_{i=1}^N g_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{p}_i) \tag{8.19}$$

Theorem 8.1 can be extended to the case when h is just continuous but the parametric domain $\mathcal{D}(\mathbf{p})$ is a subset of a compact set $\mathcal{D}$ for each $\mathbf{p} \in \mathcal{P}$, then h is uniformly continuous in $\mathcal{D}$. For any $\epsilon \in \mathbb{R}_{>0}$, we can bound each term $\|h(\hat{\mathbf{x}}_i) - h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}_i))\|$ by ϵ if each gap function $g_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{p}_i)$ is less than $\sqrt{c}\,\delta(\epsilon)$, where $\epsilon \mapsto \delta(\epsilon) \in \mathbb{R}_{>0}$ depends on h.

We can use the AER to approximate the *population risk* $\mathcal{L}(\hat{\boldsymbol{\theta}}) = \mathbb{E}[\mathbf{y} - h(\mathbf{x}^\star_{\hat{\boldsymbol{\theta}}}(\mathbf{p}))]$, where $\hat{\boldsymbol{\theta}}$ is estimated from the samples $\{(\mathbf{p}_1, \mathbf{y}_1), \cdots, (\mathbf{p}_N, \mathbf{y}_N)\}$. A useful bound is then $|\mathcal{L}(\hat{\boldsymbol{\theta}}) - \Phi_N(\{\mathbf{x}_i\}_i)| \leq |\mathcal{L}(\hat{\boldsymbol{\theta}}) - \mathcal{L}_N(\hat{\boldsymbol{\theta}})| + |\mathcal{L}_N(\hat{\boldsymbol{\theta}}) - \Phi_N(\{\mathbf{x}_i\}_i)|$. However, $\hat{\boldsymbol{\theta}}$ is random since it depends on the samples. To control both terms in the bound, we need results uniform in $\hat{\boldsymbol{\theta}}$. By requiring a uniform accuracy of $\bar{g}$, *i.e.* $g_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{p}_i)$ for every $(i, \boldsymbol{\theta}) \in [N] \times \Theta$, the quantity $|\mathcal{L}_N(\hat{\boldsymbol{\theta}}) - \Phi_N(\{\mathbf{x}_i\}_i)|$ is controlled by Theorem 8.1. The term $|\mathcal{L}(\hat{\boldsymbol{\theta}}) - \mathcal{L}_N(\hat{\boldsymbol{\theta}})|$ can be controlled by the uniform law of large numbers if the implicit function class $\{h(\mathbf{x}^\star_{\boldsymbol{\theta}}(\cdot)) : \boldsymbol{\theta} \in \Theta\}$ is Glivenko-Cantelli [156], or has a small Rademacher or Gaussian complexity [12].

8.5 Empirical risk minimization

The problem of minimizing $\mathcal{L}_N(\boldsymbol{\theta})$ can be viewed as bilevel because a convex optimization problem (or VIP) is embedded within the quantities $\mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}_1), \cdots, \mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}_N)$. Using the gap function $g_{\boldsymbol{\theta}}$ (8.17), the problem becomes explicit

$$\min_{\boldsymbol{\theta} \in \Theta, \{\mathbf{x}_i\}_i} \quad \Phi_N\left(\{\mathbf{x}_i\}_{i \in [N]}\right) \tag{8.20}$$

$$\text{s.t.} \quad g_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{p}_i) = 0,\ \mathbf{x}_i \in \mathcal{D}(\mathbf{p}_i),\ \forall\, i \in [N] \tag{8.21}$$

Smoothing. Problem (8.20)-(8.21) is an instance of an *mathematical program with equilibrium constraints (MPEC)*, see *e.g.* [110]. It is well known that an MPEC is nonsmooth since standard constraint qualifications, such as LICQ or MFCQ, are not satisfied at any feasible point [42]. Hence, minimizing the bound (8.19) on the empirical risk can be seen as solving a penalized (or smooth) form $\min_{\boldsymbol{\theta} \in \Theta,\, \{\mathbf{x}_i\}_i \in \Pi_i \mathcal{D}(\mathbf{p}_i)} \Phi_N\left(\{\mathbf{x}_i\}_{i \in [N]}\right) + \frac{\alpha}{N} \sum_{i=1}^N g_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{p}_i)$ of the MPEC (8.20)-(8.21), where $\alpha \in \mathbb{R}_{>0}$ is the penalty coefficient. Concretely, with $g_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p}) = \phi(g''_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p}))$, see (8.14), and denoting $\mathbf{A}_i := \mathbf{A}(\mathbf{p}_i)$ and $\mathbf{b}_i := \mathbf{b}(\mathbf{p}_i)$, minimizing the bound in (8.19) is equivalent to solving the program, proposed in [150]

$$\begin{aligned} \min_{\boldsymbol{\theta}, \{\mathbf{x}_i\}, \{\boldsymbol{\nu}_i\}} \ & \Phi_N(\{\mathbf{x}_i\}) + \frac{\alpha}{N} \textstyle\sum_{i=1}^N \phi(F_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{p}_i)^T \mathbf{x}_i - \mathbf{b}_i^T \boldsymbol{\nu}_i) \\ \text{s.t.} \ & \mathbf{x}_i \in \mathcal{D}(\mathbf{p}_i),\ \mathbf{A}_i^T \boldsymbol{\nu}_i \leq F_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{p}_i),\ \forall\, i \in [N] \end{aligned}$$

Pareto efficiency. The bound (8.19) on the empirical risk $\mathcal{L}_N(\boldsymbol{\theta})$ is minimal at a Pareto optimal point for the pair of objectives $\Phi_N\left(\{\mathbf{x}_i\}_{i \in [N]}\right)$ and $\frac{1}{N} \sum_{i=1}^N g_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{p}_i)$. Without information on the strongly monotone parameter c and the Lipschitz constant L, we may minimize the bound in (8.19) with different values of the ratio $L/\sqrt{c}$, thus exploring a Pareto curve, see Algorithm 8.1, and *e.g.* [116]. This approach was proposed by [150], but without the statistical interpretation of (8.19).

Inverse problems: [94, 19] investigate the case when the observation mapping h is the identity function, and learn $F_{\boldsymbol{\theta}}$ from predictor samples $\mathbf{p}_i$ and direct observations $\mathbf{y}_i = \mathbf{x}^\star_{\boldsymbol{\theta}}(\mathbf{p}_i) + \boldsymbol{\epsilon}_i$. Correcting the $\mathbf{y}_i$'s so that $\mathbf{y}_i \in \mathcal{D}(\mathbf{p}_i)$, for every $i \in [N]$, they solve *the inverse problem* $\min_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^N g_{\boldsymbol{\theta}}(\mathbf{y}_i, \mathbf{p}_i)$. Relating to (8.19), they implicitly solve

Algorithm 8.1 weighted sum method for Pareto optimization

Choose set of weights $\mathcal{W}$ in $(0,1)$
Normalize $\Phi_N\left(\{\mathbf{x}_i\}_{i\in[N]}\right)$ and $\frac{1}{N}\sum_{i=1}^N g_i(\hat{\mathbf{x}}_i,\boldsymbol{\theta})$
for $w \in \mathcal{W}$:
  Minimize $w\,\Phi_N\left(\{\mathbf{x}_i\}_i\right) + (1-w)\frac{1}{N}\sum_{i=1}^N g_{\boldsymbol{\theta}}(\mathbf{x}_i,\mathbf{p}_i)$
Check values of $\Phi_N\left(\{\mathbf{x}_i\}_i\right)$ and $\frac{1}{N}\sum_{i=1}^N g_{\boldsymbol{\theta}}(\mathbf{x}_i,\mathbf{p}_i)$

$\min_{\boldsymbol{\theta}\in\Theta,\{\mathbf{x}_i\}_i} \sum_{i=1}^N g_{\boldsymbol{\theta}}(\mathbf{x}_i,\mathbf{p}_i)$ s.t. $\sum_{i=1}^N \|\mathbf{y}_i - \mathbf{x}_i\| = 0$, *i.e.* the gap functions are minimized with the constraint that the AER is zero. This approach is motivated by tractability since the *inverse problem* can be made convex, see [19, 94]. Concretely, with $F_{\boldsymbol{\theta}}(\cdot,\mathbf{p}_i) = \sum_{j=1}^r \theta_j F_j(\cdot,\mathbf{p}_i)$ given basis mappings $F_j : \mathbb{R}^n \times \mathcal{P} \to \mathbb{R}^n$, and objective $\sum_{i=1}^N \phi(g''_{\boldsymbol{\theta}}(\mathbf{x}_i,\mathbf{p}_i))$ with ϕ convex, the inverse problem is convex,

$$\min_{\boldsymbol{\theta}\in\Theta,\{\boldsymbol{\nu}_i\}_i} \sum_{i=1}^N \phi(F_{\boldsymbol{\theta}}(\mathbf{y}_i,\mathbf{p}_i)^T\mathbf{y}_i - \mathbf{b}_i^T\boldsymbol{\nu}_i) \tag{8.22}$$

$$\text{s.t.} \quad \mathbf{A}_i^T\boldsymbol{\nu}_i \le F_{\boldsymbol{\theta}}(\mathbf{y}_i,\mathbf{p}_i), \quad \forall\, i \in [N] \tag{8.23}$$

Noise sensitivity. However, random errors may yield a large objective value in the program (8.22)-(8.23) at an optimum $\hat{\boldsymbol{\theta}}$, hence large gap functions $g''_{\hat{\boldsymbol{\theta}}}(\mathbf{y}_i,\mathbf{p}_i)$, *i.e.* the observations $\mathbf{y}_i$ do not approximate $\mathbf{x}^\star_{\hat{\boldsymbol{\theta}}}(\mathbf{p}_i)$ well. This causes the learned model, or the learned $\hat{\boldsymbol{\theta}}$, to have low prediction accuracy, *i.e.* the predicted outcomes $\hat{\mathbf{y}}_i \approx \mathbf{x}^\star_{\hat{\boldsymbol{\theta}}}(\mathbf{p}_i)$ are far from the observed ones $\mathbf{y}_i = \hat{\mathbf{x}}_i$. From the risk minimization perspective, a zero AER reduces the bound in Theorem 8.1 to $\mathcal{L}_N(\hat{\boldsymbol{\theta}}) \le \frac{L}{N\sqrt{c}}\sum_{i=1}^N g_{\hat{\boldsymbol{\theta}}}(\mathbf{y}_i,\mathbf{p}_i)$, hence large gap functions despite a zero AER do not guarantee a small risk.

To illustrate the sensitivity of the inverse problems (8.22)-(8.23) to noise, consider the problem of learning $\hat{\theta} \in \mathbb{R}$ in the univariate program $\min_{x\in\mathbb{R}_+}(x-\theta^\star)^2$, with $\theta^\star \in \mathbb{R}_+$, from observations $y_i = x^\star(\theta^\star) + \epsilon_i = \theta^\star + \epsilon_i$, $i \in [N]$. Assume there is one measurement error $\epsilon_1 = -\alpha$ for $\alpha \in [0,\theta^\star]$, and $\epsilon_i = 0$ for $i = 2,\cdots,N$. Solving (8.22)-(8.23) instantiated to the present problem gives $\hat{\theta} = \theta^\star - \alpha$ and large risk and gap $\mathcal{L}_N(\hat{\theta}) = \frac{N-1}{N}\,\alpha$, with the norm in the AER and $\mathcal{L}_N(\theta)$ instantiated to the absolute value and $\phi = \mathrm{Id}$. Hence, having a better trade-off between the AER and the approximation accuracy may give better results. If we use the weighted sum method described in Algorithm 1 on the present example, we get (see Appendix E for more details),

$$\begin{aligned} w \le \tfrac{(\theta^\star-\alpha)\,(N-1)}{1+(\theta^\star-\alpha)\,(N-1)} &\implies \mathcal{L}_N(\hat{\theta}) = \tfrac{\alpha}{N} \\ w > \tfrac{\theta^\star\,(N-1)}{1+\theta^\star\,(N-1)} &\implies \mathcal{L}_N(\hat{\theta}) = \tfrac{N-1}{N}\alpha \end{aligned}$$

8.6 Concentration bounds

A central problem in supervised learning involves validating the learned model. Let us define the *true model* as

$$\mathbf{y} = h(\mathbf{z}(\mathbf{p})) + \boldsymbol{\epsilon} \tag{8.24}$$

for a random error $\boldsymbol{\epsilon} \in \mathbb{R}^m$ with variance at most σ^2, and an *unknown* mapping $\mathbf{z} : \mathcal{P} \to \mathbb{R}^n$ such that $\mathbf{z}(\mathbf{p}) \in \mathcal{D}(\mathbf{p})$ for each $\mathbf{p} \in \mathcal{P}$. Note that $\mathbf{z}(\cdot)$ is different from $\mathbf{x}^\star(\cdot)$ in (8.1) and (8.2) because it may not be solution to a VIP. We recall the learning problem of the present paper. We observe N i.i.d. samples $\mathbf{p}_i$, $i \in [N]$ of the random predictor vector $\mathbf{p} \in \mathcal{P}$ and the resulting response vectors $\mathbf{y}_i \in \mathbb{R}^m$, $i \in [N]$. From (8.24), the relationship between $\mathbf{p}_i$ and $\mathbf{y}_i$ is given by

$$\mathbf{y}_i = h(\mathbf{z}(\mathbf{p}_i)) + \boldsymbol{\epsilon}_i, \quad \forall\, i \in [N] \tag{8.25}$$

where $\boldsymbol{\epsilon}_i$ are i.i.d. samples from the noise vector $\boldsymbol{\epsilon}$. We have available the parametric convex compact domain $\mathcal{D}(\mathbf{p}) \subset \mathbb{R}^n$, the L-Lipschitz observation mapping $h : \mathbb{R}^n \to \mathbb{R}^m$, and an indexed-family $\mathcal{F} = \{(\mathbf{x}, \mathbf{p}) \mapsto F_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{p}) : \boldsymbol{\theta} \in \Theta\}$ such that the mapping $F_{\boldsymbol{\theta}}(\cdot, \mathbf{p})$ is strongly monotone with parameter c for each $(\mathbf{p}, \boldsymbol{\theta}) \in \mathcal{P} \times \Theta$. We apply one of the methods presented in Section 8.5 to learn the parameter vector $\boldsymbol{\theta}$. Let $\{\hat{\mathbf{x}}_i\}_{i\in[N]}$ be the estimated state vectors and $\hat{\boldsymbol{\theta}}$ the estimated parameters, from which we define the *maximal approximation error* $\bar{g} = \max_{i\in[N]} g_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_i, \mathbf{p}_i)$, where $g_{\boldsymbol{\theta}}$ satisfies (8.17) from the analysis in Section 8.4. We can also interpret $\bar{g}$ as a desired level of accuracy.

We now equip the predictor space $\mathcal{P}$ with a structure of measure space $(\mathcal{P}, \Sigma, \mu)$ where Σ is a σ-algebra of measurable sets, and μ is a probability measure. Under μ, we assume the essential supremum $s(\hat{\boldsymbol{\theta}})$ and the expected value $d(\hat{\boldsymbol{\theta}})$ of $\mathbf{p} \mapsto \|h(\mathbf{x}^\star_{\hat{\boldsymbol{\theta}}}(\mathbf{p})) - h(\mathbf{z}(\mathbf{p}))\|$ over $\mathcal{P}$ finite,[3]

$$s(\hat{\boldsymbol{\theta}}) = \operatorname{ess\,sup} \|h(\mathbf{x}^\star_{\hat{\boldsymbol{\theta}}}(\mathbf{p})) - h(\mathbf{z}(\mathbf{p}))\| \tag{8.26}$$

$$d(\hat{\boldsymbol{\theta}}) = \mathbb{E}\left[\|h(\mathbf{x}^\star_{\hat{\boldsymbol{\theta}}}(\mathbf{p})) - h(\mathbf{z}(\mathbf{p}))\|\right] \tag{8.27}$$

This allows us to define the *error-of-fit (EOF)*, $d(\hat{\boldsymbol{\theta}})$, as the expected distance between the responses of the true and learned models. Note that we do not make any assumptions on the observation mapping h other than Lipschitz continuity, and can only hope to match the responses of both models, with disregard for the underlying equilibrium states they describe. We have the following sub-Gaussian concentration property of the AER, $\Phi_N\left(\{\mathbf{x}_i\}_{i\in[N]}\right)$, in an interval $[d(\hat{\boldsymbol{\theta}}) - \delta,\, d(\hat{\boldsymbol{\theta}}) + \delta]$, where the associated *sub-Gaussian parameter* Ω and the

[3]Showing continuity of $\mathbf{p} \mapsto \|h(\mathbf{x}^\star_{\hat{\boldsymbol{\theta}}}(\mathbf{p})) - h(\mathbf{z}(\mathbf{p}))\|$ requires sensitivity analysis [133]. In a more general setting, we do not assume continuity, and essential bound implies that the function values are less than $s(\hat{\boldsymbol{\theta}})$ almost surely.

model error δ are given by

$$\Omega = \sigma + \tfrac{s(\hat{\boldsymbol{\theta}})}{2} \tag{8.28}$$

$$\delta = \sigma\sqrt{m} + \tfrac{L}{\sqrt{c}}\bar{g} \tag{8.29}$$

Theorem 8.2. *Assume* $\{\boldsymbol{\epsilon}_i\}_{i\in[N]}$ *are i.i.d. samples from a vector* $\boldsymbol{\epsilon} \in \mathbb{R}^m$ *of independent Gaussian random variables with zero mean and variance at most* σ^2*, and the norm in the observed space* $\mathbb{R}^m$*, used in the definitions of* $\mathcal{L}_N(\boldsymbol{\theta})$ *and the AER, is Euclidean. If* $d(\hat{\boldsymbol{\theta}}) > 0$*, for every* $t \geq 0$

$$\mathbb{P}\left[\left|\Phi_N\left(\{\boldsymbol{x}_i\}_{i\in[N]}\right) - d(\hat{\boldsymbol{\theta}})\right| \geq t + \delta\right] \leq 3\exp\left\{-\tfrac{Nt^2}{2\Omega^2}\right\}$$

Remarkably, Ω does not depend on the dimension of the observed space $\mathbb{R}^m$, hence the contribution of the noise vector $\boldsymbol{\epsilon} \in \mathbb{R}^m$ in the concentration inequality is that of a scalar Gaussian with variance σ^2. The results in Theorem 8.2 also imply: smaller noise variance σ or smaller error $s(\hat{\boldsymbol{\theta}})$ guarantee a smaller Ω, hence stronger concentration in t and N; and smaller variance σ or smaller approximation error $\bar{g}$ guarantee a smaller δ, hence a concentration in a smaller neighborhood of $d(\hat{\boldsymbol{\theta}})$. Theorem 2 enables to approximate the empirical distribution of the AER. We also derive tail bounds on the distribution of the AER if the fit is perfect.

Theorem 8.3. *Under the assumptions of Theorem 8.2, if* $d(\hat{\boldsymbol{\theta}}) = 0$*, then for every* $t \geq 0$

$$\mathbb{P}\left[\Phi_N\left(\{\boldsymbol{x}_i\}_{i\in[N]}\right) \geq t + \delta\right] \leq \exp\left\{-\tfrac{Nt^2}{2\sigma^2}\right\}$$

Note that Theorem 8.3 is not a direct consequence of Theorem 8.2 with $d(\hat{\boldsymbol{\theta}}) = 0$, since the concentration parameter is smaller. We defer the proofs of both theorems to Appendix F. They use results on the concentration of Lipschitz functions of standard Gaussian variables, see *e.g.* [25, Theorem 5.6]. We extend the results to the case when the observed space is equipped with the p-norm for some $p \in [1, \infty)$. This is of interest since it gives concentration results for the ℓ_1-norm, used for robust learning, see *e.g.* [27, §6.1.2.]. In particular, Proposition 8.1 claims that Theorem 8.2 holds with $\Omega = \sigma\sqrt{m} + \frac{s(\hat{\boldsymbol{\theta}})}{2}$ and $\delta = \sigma\, m + \frac{L}{\sqrt{c}}\bar{g}$ with the ℓ_1-norm.

Proposition 8.1. *Suppose the assumptions of Theorem 8.2 hold, with the difference that the observed space* $\mathbb{R}^m$ *is equipped with the p-norm. If* $p \geq 2$*, the result of Theorem 8.2 holds. If* $p \in [1, 2)$*, the result of Theorem 8.2 holds with* $\Omega = \sigma\, m^{1/p-1/2} + \frac{s(\hat{\boldsymbol{\theta}})}{2}$ *and* $\delta = \sigma\, m^{1/p} + \frac{L}{\sqrt{c}}\bar{g}$*.*

8.7 Hypothesis testing and statistical power

The concentration inequalities derived in Section 8.6 are useful approximations of the distribution of the AER, $\Phi_N\left(\{\mathbf{x}_i\}_{i\in[N]}\right) = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{y}_i - h(\hat{\mathbf{x}}_i)\|$. This allows us to measure the quality of the learned model $\hat{\boldsymbol{\theta}}$.

Test definition. Theorem 8.2 states that the AER concentrates in a neighborhood of the EOF, $d(\hat{\boldsymbol{\theta}}) = \mathbb{E}\left[\|h(\mathbf{x}^\star_{\hat{\boldsymbol{\theta}}}(\mathbf{p})) - h(\mathbf{z}(\mathbf{p}))\|\right]$. Hence large values of the EOF are likely to cause large values of the AER, and this leads us to pose the AER as a t-statistic for testing the hypothesis that the learned model is accurate. To formalize, We pose the *null hypothesis* that the true model and the learned model coincide exactly

$$H_0: \quad d(\hat{\boldsymbol{\theta}}) = 0$$

Under H_0, Theorem 8.3 states the following, for every $t \geq 0$,
$\mathbb{P}\left[\Phi_N(\{\hat{\mathbf{x}}_i\}_i) \geq t + \delta\right] \leq \exp\left\{-\frac{Nt^2}{2\sigma^2}\right\}$, where we recall $\delta = \sigma\sqrt{m} + \frac{L}{\sqrt{c}}\bar{g}$. Hence we can approximate the empirical distribution of the observed p-value, defined as

$$\text{p-value} = \mathbb{P}\left[\Phi_N\left(\{\mathbf{x}_i\}_{i\in[N]}\right) - \delta \geq t\right] \tag{8.30}$$

Define a significance level $\alpha \in (0,1)$. From (8.30), the following condition is sufficient for the p-value to be less than α. We reject H_0 if it is satisfied:

$$\Phi_N\left(\{\mathbf{x}_i\}_{i\in[N]}\right) - \delta \geq t_\alpha, \quad t_\alpha = \sigma\sqrt{\tfrac{2\ln(1/\alpha)}{N}} \tag{8.31}$$

Note that (8.31) gets more sensitive to large values of the AER when the sample size N increases. The significance criterion (8.31) guarantees that the probability of a type I error, rejecting H_0 while it is true, is at most α. If the test rejects H_0, we can expect to either have a poor estimate $\hat{\boldsymbol{\theta}}$, or a parametric class of implicit functions $\{\mathbf{x}^\star_{\boldsymbol{\theta}}(\cdot) : \boldsymbol{\theta} \in \Theta\}$ that is too restrictive or that is not a good model for $\mathbf{z}(\cdot)$.

Statistical power. The alternative hypothesis is defined as

$$H_1: \quad d(\hat{\boldsymbol{\theta}}) > 0$$

and we want to estimate the power of the test, defined as the probability of accepting the alternative hypothesis when it is true, formally $\mathbb{P}[\text{reject } H_0 \,|\, H_1]$. It is also the probability of not making a type II error. Using Theorem 8.2, Lemma 8.1 gives a lower bound on it, where the positive part of $x \in \mathbb{R}$ is denoted by x_+. Combining with a second lemma, we derive the result in Theorem 8.4.

Lemma 8.1. *Let $\tau_\alpha = [d(\hat{\boldsymbol{\theta}}) - 2\,\delta - t_\alpha]_+$ and $\alpha \in (0,1)$. The test defined in* (8.31) *with significance level α, has power*

$$\mathbb{P}[\textit{reject } H_0 \,|\, H_1] \geq 1 - 3\exp\left\{-\tfrac{N\tau_\alpha^2}{2\,\Omega^2}\right\} \tag{8.32}$$

Proof. Under the alternative hypothesis H_1, $d(\hat{\boldsymbol{\theta}}) > 0$, Theorem 8.2 states that $\Phi_N\left(\{\mathbf{x}_i\}_{i\in[N]}\right)$ concentrates in a neighborhood $[d(\hat{\boldsymbol{\theta}}) - \delta,\, d(\hat{\boldsymbol{\theta}}) + \delta]$ of the EOF. In view of (8.31), we expect the power of the test to rise with the sample size N if $d(\hat{\boldsymbol{\theta}}) - \delta > t_\alpha + \delta$. This leads us to

pose the positive part of the difference $\tau_\alpha = [d(\hat{\boldsymbol{\theta}}) - 2\delta - t_\alpha]_+$. If $\tau_\alpha = 0$, the right-hand side of (8.32) is negative and the inequality is true. If $\tau_\alpha > 0$, $t_\alpha + \delta = d(\hat{\boldsymbol{\theta}}) - \delta - \tau_\alpha$, then, using Theorem 8.2

$$\begin{aligned}
\mathbb{P}[\text{reject } H_0 \,|\, H_1] &= \mathbb{P}\left[\Phi_N\left(\{\mathbf{x}_i\}_{i\in[N]}\right) \geq t_\alpha + \delta\right] \\
&= \mathbb{P}\left[\Phi_N\left(\{\mathbf{x}_i\}_{i\in[N]}\right) \geq d(\hat{\boldsymbol{\theta}}) - \delta - \tau_\alpha\right] \\
&\geq \mathbb{P}\left[|\Phi_N\left(\{\mathbf{x}_i\}_{i\in[N]}\right) - d(\hat{\boldsymbol{\theta}})| \leq \delta + \tau_\alpha\right] \\
&\geq 1 - 3\exp\left\{-\tfrac{N\tau_\alpha^2}{2\Omega^2}\right\}
\end{aligned}$$

□

Lemma 8.2. *Assume $d(\hat{\boldsymbol{\theta}}) > 2\,\delta$ and let $\beta \in (0,1)$. Then the inequality $1 - 3\exp\left\{-\frac{N\tau_\alpha^2}{2\Omega^2}\right\} \geq 1-\beta$ is equivalent to $\sqrt{N} \geq \frac{\Omega\sqrt{2\ln(3/\beta)}+\sigma\sqrt{2\ln(1/\alpha)}}{d(\hat{\boldsymbol{\theta}})-2\delta}$*

Proof. We start with the following equivalence $1 - 3\exp\left\{-\frac{N\tau_\alpha^2}{2\,\Omega^2}\right\} \geq 1-\beta \iff \tau\sqrt{N} \geq \Omega\sqrt{2\ln(3/\beta)}$. Since the term in the right-hand side is positive because $\beta \in (0,1)$, we can substitute τ_α with $d(\hat{\boldsymbol{\theta}}) - 2\,\delta - \sigma\sqrt{\frac{2\ln(1/\alpha)}{N}}$. By re-arranging the terms such that $\sqrt{N}(d(\hat{\boldsymbol{\theta}}) - 2\,\delta)$ is on the left-hand side of the inequality, and dividing by $d(\hat{\boldsymbol{\theta}}) - 2\,\delta$ without changing the sense of the inequality because it is positive, we obtain the equivalence. □

Theorem 8.4. *Let $\alpha, \beta \in (0,1)$. Given the test defined in* (8.30) *and* (8.31) *with significance level α, its power is at least $1-\beta$ if we have the following sufficient conditions*

$$d(\hat{\boldsymbol{\theta}}) > 2\,\delta \tag{8.33}$$

$$\sqrt{N} \geq \frac{\Omega\sqrt{2\ln(3/\beta)}+\sigma\sqrt{2\ln(1/\alpha)}}{d(\hat{\boldsymbol{\theta}})-2\delta} \tag{8.34}$$

Theorem 8.4 is a direct implication from Lemma 8.2 and Lemma 8.1. Under the null hypothesis, the AER concentrates inside $[0,\delta]$ from Theorem 8.3, and under the alternate hypothesis the AER concentrates inside $[d(\hat{\boldsymbol{\theta}})-\delta,\, d(\hat{\boldsymbol{\theta}})+\delta]$ from Theorem 8.2. We then expect to have a high statistical power when both of these intervals are disjoint, which is equivalent to having $d(\hat{\boldsymbol{\theta}}) > 2\,\delta$. Theorem 8.4 formalizes this intuition by stating a sufficient condition on the sample size, on the significance α, and on the power $1-\beta$. If higher significance or power is desired, *i.e.* smaller α or β, (8.34) states that a larger sample size may be required. We have the same implication if δ is larger, due to *e.g.* larger noise variance σ or larger approximation error $\bar{g}$. If δ is large enough so that $d(\hat{\boldsymbol{\theta}}) < 2\,\delta$, we may loose the ability to reject the learned model if H_1 is true. A more optimistic view is that having $d(\hat{\boldsymbol{\theta}}) < 2\,\delta$ means that the EOF is small, and we may not want to reject the learned model anymore, independently from whether H_1 is true or false. Finally, Theorem 8.4 can be interpreted as: *for the test defined in* (8.30) *and* (8.31)*, the condition $d(\boldsymbol{\theta}) > 2\delta$ is sufficient to get an arbitrary high statistical power provided that we have enough samples.*

8.8 Concluding remarks

The proposed statistical framework, and the resulting analysis and t-statistic have far-reaching potential on the modeling and validation of complex real-world processes as equilibria, since variational inequality and convex optimization problems have a wide variety of applications.

To obtain a concentration of measure for the AER that is the one of a scalar variable, we assumed Gaussian errors and used results on the concentration of Lipschitz functions of Gaussian variables. However, it remains an open question whether or not a similar property holds for sub-Gaussian variables. In the case of distribution-free bounded random errors, dimensionless concentration results can still be obtained by using bounded differences inequalities, see *e.g.* [25, Theorem 6.10].

Part III

Estimating traffic flow on the highway and arterial networks

Chapter 9

State Estimation for Polyhedral Hybrid Systems

This chapter investigates the problem of estimating the state of discretized hyperbolic scalar partial differential equations. It uses a Godunov scheme to discretize the so-called *Lighthill-Whitham-Richards* equation with a triangular flux function, and proves that the resulting nonlinear dynamical system can be decomposed in a *piecewise affine* manner. Using this explicit representation, the system is written as a switching dynamical system, with a state space partitioned into an exponential number of polyhedra in which one mode is active. We propose a feasible approach based on the interactive multiple model (IMM) which is a widely used algorithm for estimation of hybrid systems in the scientific community. The number of modes is reduced based on the geometric properties of the polyhedral partition. The *k-means* algorithm is also applied on historical data to partition modes into clusters. The performance of these algorithms are compared to the extended Kalman filter and the ensemble Kalman filter in the context of Highway Traffic State Estimation. In particular, we use sparse measurements from loop detectors along a section of the I-880 to estimate the state density for our numerical experiments.

9.1 Introduction

Partial Differential Equations (PDEs) are often used in traffic as density based traffic models because they provide a concise mathematical model to capture essential properties of a wide variety of phenomena such as fluid flow, heat, and electrodynamics. Based on the conservation of flow, the *Lighthill-Whitham-Richards* (LWR) PDE [109, 136] and its discretization using the Godunov scheme [100, 106, 146] have been widely used in the scientific community for modelling traffic, they also known as the *Cell Transmission Model* (CTM) [50, 51] in the transportation literature. State of the art traffic estimation techniques for this model include the application of the *extended Kalman filter* (EKF) to the LWR PDE by Schreiter et al. [141], and to non-scalar traffic model by Papageorgiou [128]. The application of the EKF to

the LWR PDE model is problematic due to the non-differentiability of its discretization, a problem which has been partially addressed in [23] and [151]. The *ensemble Kalman filter* (EnKF) has also been applied to a velocity-based model in [170], in order to circumvent the difficulties of non-differentiability of numerical solutions to these PDEs such as the one presented in this chapter.

The Godunov scheme applied to the LWR model for a triangular flux function can be proven to lead to a *piecewise affine* (PWA) hybrid system, which is one of the contributions of this chapter. Each cell of the discretized system switches between several linear models. We define this new class of systems as *multicellular hybrid systems*. The resulting switching-mode dynamical system combines discrete dynamics modeled by a finite automaton for the *transitions* between the modes and continuous dynamics in the form of linear discretized dynamical systems. Estimation of hybrid systems has been widely studied in past work [104, 105]. In particular, such techniques have been successfully used for aircraft tracking in [83] in which Bar-Shalom's *interacting multiple model* (IMM) algorithm was used [9]. Similar hybrid estimation algorithms and their applications are described in [118, 147, 80]. While the IMM algorithm seems a natural approach for the estimation of hybrid systems, it is intractable when applied to the discretized LWR PDE (thus highway models) because the combination of the modes of each cell induce an exponential number of modes. A priori, each cell of the discretized model can be in seven different modes, which leads to 7^n modes, where n is the dimension of the state thus creating serious computational challenges in the estimation problem. One possible way to address this is with the *mixture Kalman filter* algorithm [40] which handles this complexity by randomly sampling in the space of modes.

Our work contains four contributions. To the best of our knowledge, this is the first time that an explicit piecewise affine decomposition of the Godunov is formulated. 1) For a fixed mode vector $\boldsymbol{m}$, the Godunov scheme is locally affine, and we have an explicit formulation of the linear dynamics. 2) The domains of the mode vectors $\mathrm{Dom}(\boldsymbol{m})$ are also expressed with explicit linear constraints, and they form a polyhedral partition of the state space. Even though the IMM is a natural algorithm for hybrid estimation, it is not tractable because of the exponential number of modes. Hence the second contribution consists in proposing two methods: 3) The first one takes advantage of the geometric properties of the space of modes to reduce the set of modes to the mode of the current estimate and its adjacent modes. 4) The second one uses a clustering algorithm on historical data to reduce the set of modes to a representative sets: then the reduced model only switches between these modes.

The rest of the chapter is organized as follow: Section 9.2 presents the mathematical model used and unravels the PWA expression of the Godunov scheme. Section 9.3 presents the polyhedral properties of the space of modes. Section 9.4 shows that the IMM applied to the discretized system is not tractable. Section 9.5 presents feasible algorithms inspired from IMM using the PWA character of the Godunov scheme and *k-means*.[1]

[1]Code available here: https://github.com/jeromethai/hybrid-LWR-estimation

9.2 A Hybrid Automaton

The LWR Model

Lighthill, Whitham in 1955 [109], and Richards in 1956 [136] introduced a macroscopic dynamic model of traffic based on conservation of vehicles, using Greenshields' hypothesis [73] of a static flow/density relationship (9.1), known as the *flux function*:

$$q(x,t) = Q(\rho(x,t)) \tag{9.1}$$

where $\rho(x,t)$ and $q(x,t)$ denote the density and the flow of vehicles at location x and time t respectively. The flux function Q is assumed to be a function of the density only. The conservation of mass can be rewritten as follows:

$$\begin{array}{l} \frac{\partial \rho(x,t)}{\partial t} + \frac{\partial Q(\rho(x,t))}{\partial x} = 0, \quad \forall\, (x,t) \in [0,L] \times \mathbb{R}_+ \\ \rho(0,t) = u(t), \quad \rho(L,t) = d(t) \quad \forall\, t \in \mathbb{R}_+ \\ \rho(x,0) = \rho_0(x), \quad \forall\, x \in [0,L] \end{array} \tag{9.2}$$

where $u(t)$, $d(t)$ are the upstream and downstream densities respectively, and $\rho_0(x)$ is the initial state [109, 136]. This equation is commonly known as the *Lighthill-Whitham-Richards*, or LWR, model. Different flux functions have been suggested.

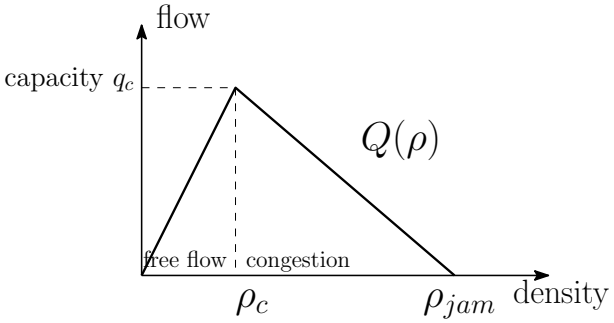

Figure 9.1: Speed and flow relationships for triangular flux function.

At each boundary, the ability to prescribe the value of the solution depends on the sign of the characteristic curve (if it is entering the domain, it can be done in the strong sense, otherwise it cannot be done). Thus, in order for the problem to be well posed, one needs to prescribe the boundary conditions in the weak sense, and they can either apply at the two boundaries, at one boundary or at none of the boundaries, depending on the value of the function in the interior of the domain. This result is described in detail in [10] for a compact domain. It was later instantiated for specific PDEs, in particular in the work of [103], and in the specific case of traffic (concave flux function) in [146].

Assumptions and notations

In the rest of the chapter, we will focus on the analysis of the Godunov scheme, which is a conservative numerical scheme for solving PDE. We assume that traffic densities are between 0 and ρ_{jam}, *i.e.* the density $\rho(x,t)$ is in $[0,\, \rho_{\text{jam}}]$ for all x, t.

The widely-used *triangular flux function* described in [50] is also chosen for our dynamic model and results are derived from it. It is a function of the density ρ. It assumes a constant velocity in free-flow and a hyperbolic velocity in congestion as shown in Figure 9.1.

$$Q(\rho) = \begin{cases} v_f \rho & \text{if } \rho \leq \rho_c \\ -\omega_f \left(\rho - \rho_{\text{jam}}\right) & \text{if } \rho > \rho_c \end{cases} \tag{9.3}$$

where $\omega_f = v_f \rho_c / (\rho_{\text{jam}} - \rho_c)$ is the backward propagation wave speed.

We also assume for simplicity and clarity that the segment of road we are modeling is homogeneous, i.e. the parameters of the flux function ω_f, v_f, ρ_{jam}, ρ_c, q_c are uniform along the cells of the discretized road. All the results derived in the rest of the chapter still remain valid for an heterogeneous road.

The Godunov scheme

A seminal numerical method to solve the above equations is given by the Godunov scheme, which is based on exact solutions to Riemann problems [71, 72]. This leads to the construction of a nonlinear discrete time dynamical system. The Godunov discretization scheme is applied on the LWR PDE, where the discrete time step Δt is indexed by t, and the discrete space step Δx is indexed by i:

$$\rho_i^{t+1} = \rho_i^t - \frac{\Delta t}{\Delta x}\left(G(\rho_i^t, \rho_{i+1}^t) - G(\rho_{i-1}^t, \rho_i^t)\right), \quad i = 1, \cdots, n \tag{9.4}$$

In order to ensure numerical stability, the time and space steps are coupled by the CFL condition [106]: $c_{\max}\frac{\Delta t}{\Delta x} \leq 1$ where $c_{\max}$ denotes the maximal characteristic speed.

The Godunov flux can be expressed as the minimum of the *sending flow* $S(\rho)$ from the upstream cell and the *receiving flow* $R(\rho)$ from the downstream cell through a boundary connecting two cells of a homogeneous road (i.e. the upstream and downstream cells have the same characteristics). For the triangular flux function:

$$\begin{aligned} G(\rho_1, \rho_2) &= \min(S(\rho_1), R(\rho_2)) \\ S(\rho) &= \begin{cases} Q(\rho) = v_f \rho & \text{if } \rho \leq \rho_c \\ q_c & \text{if } \rho > \rho_c \end{cases} \\ R(\rho) &= \begin{cases} q_c & \text{if } \rho \leq \rho_c \\ Q(\rho) = -\omega_f \left(\rho - \rho_{\text{jam}}\right) & \text{if } \rho > \rho_c \end{cases} \end{aligned} \tag{9.5}$$

where ρ_1 is the density of the cell upstream and ρ_2 is the density of the cell downstream.

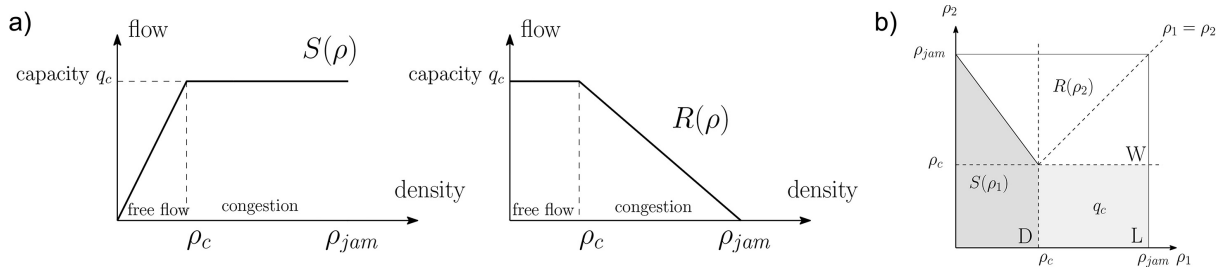

Figure 9.2: a) Sending and receiving flows for triangular flux function. b) Values of $G(\rho_1, \rho_2)$ in the space $[0,\, \rho_{\text{jam}}]^2$.

As shown in Figure 9.2.a), the application of the Godunov scheme to the flux functions introduces intuitive concepts of *supply* and *demand* at the boundary connecting two cells.

Given the partition of the space $[0,\, \rho_{\text{jam}}]^2$ in different regions **W**, **L**, and **D** as shown in Figure 9.2.b), the function $G(\rho_1, \rho_2)$ takes different values.

Lemma 9.1. *With a triangular flux function, the Godunov flux* $(\rho_1,\, \rho_2) \in [0,\, \rho_{jam}]^2 \mapsto G(\rho_1,\, \rho_2)$ *is piecewise affine:*

$$
\begin{aligned}
G(\rho_1, \rho_2) &= \begin{cases} -\omega_f\,(\rho_2 - \rho_{\text{jam}}) & \text{if } (\rho_1, \rho_2) \in \boldsymbol{W} \\ q_c & \text{if } (\rho_1, \rho_2) \in \boldsymbol{L} \\ v_f \rho_1 & \text{if } (\rho_1, \rho_2) \in \boldsymbol{D} \end{cases} \\
\boldsymbol{W} &:= \{(\rho_1, \rho_2) \mid \rho_2 + \tfrac{v_f}{w_f}\rho_1 > \rho_{\text{jam}}\ ,\ \rho_2 > \rho_c\} \\
\boldsymbol{L} &:= \{(\rho_1, \rho_2) \mid \rho_1 > \rho_c\ ,\ \rho_2 \leq \rho_c\} \\
\boldsymbol{D} &:= \{(\rho_1, \rho_2) \mid \rho_2 + \tfrac{v_f}{w_f}\rho_1 \leq \rho_{\text{jam}}\ ,\ \rho_1 \leq \rho_c\}
\end{aligned}
\tag{9.6}
$$

Proof. We recall that $(\rho_1,\, \rho_2) \in [0,\, \rho_{\text{jam}}]^2$. Equations (9.5) imply:

$$
\begin{aligned}
&\rho_1,\, \rho_2 \leq \rho_c && \Longrightarrow G(\rho_1, \rho_2) = \min(v_f\rho_1,\, q_c) = v_f\rho_1 \\
&\rho_1,\, \rho_2 \geq \rho_c && \Longrightarrow G(\rho_1, \rho_2) = \min(q_c,\, -w_f(\rho_2 - \rho_{\text{jam}})) \\
& && = -w_f(\rho_2 - \rho_{\text{jam}}) \\
&\rho_1 \geq \rho_c,\, \rho_2 \leq \rho_c && \Longrightarrow G(\rho_1,\, \rho_2) = \min(q_c,\, q_c) = q_c \\
&\rho_1 \leq \rho_c,\, \rho_2 \geq \rho_c && \Longrightarrow G(\rho_1,\, \rho_2) = \min(v_f\rho_1,\, -w_f(\rho_2 - \rho_{\text{jam}}))
\end{aligned}
$$

The third implication proves our result for the region **L**. Then, given $\rho_1 \leq \rho_c$, $\rho_2 \geq \rho_c$, $G(\rho_1,\, \rho_2) = v_f\rho_1 \iff v_f\rho_1 \leq -w_f(\rho_2 - \rho_{\text{jam}}) \iff \rho_2 + \frac{v_f}{w_f}\rho_1 \leq \rho_{\text{jam}}$. Finally, we note that $\{\rho_1 \leq \rho_c,\, \rho_2 \leq \rho_c\} \cup \{\rho_2 + \frac{v_f}{w_f}\rho_1 \leq \rho_{\text{jam}}\} = \{\rho_2 + \frac{v_f}{w_f}\rho_1 \leq \rho_{\text{jam}}\ ,\ \rho_1 \leq \rho_c\}$ hence the definition of **D** in (9.6). The result for **W** follows similarly. □

Godunov scheme as a Hybrid Automaton

We now consider an entire link divided into n cells and *we add two ghost cells on the left and right sides of the domain.* Hence, the discrete state space is indexed by $i = 0,\, 1,\, \cdots,\, n+1$,

the state of the system is $\boldsymbol{\rho} = [\rho_0, \cdots, \rho_{n+1}]^T \in [0,\, \rho_{\text{jam}}]^{n+2}$, and the dimension is $n+2$. The density at cell i and time t is then ρ_i^t, the i-th entry of vector $\boldsymbol{\rho}$, and the values of ρ_0^t and ρ_{n+1}^t are given by the prescribed boundary conditions to be imposed on the left and right side of the domain respectively, *i.e.* $\rho_0^t = u(t)$ and $\rho_{n+1}^t = d(t)$ for all t where $u(t)$ and $d(t)$ are the upstream and downstream densities respectively.

In the rest of the section we present a simple analysis for the formulation of the discretized system as a *piecewise affine* Autonomous Hybrid Automaton. We will sometimes use the lighter notation $\rho_i^+ = \rho_i - \alpha\,(G(\rho_i, \rho_{i+1}) - G(\rho_{i-1}, \rho_i))$ for the Godunov scheme (9.4) with $\alpha = \Delta t/\Delta x$. We rewrite equations (9.6) in the state space $[0,\, \rho_{\text{jam}}]^{n+2}$.

$$G(\rho_i, \rho_{i+1}) = \begin{cases} -\omega_f\,(\rho_{i+1} - \rho_{\text{jam}}) & \text{if } \boldsymbol{\rho} \in \mathbf{W}_{i+1/2} \\ q_c & \text{if } \boldsymbol{\rho} \in \mathbf{L}_{i+1/2} \quad \text{for } i = 0, \cdots n \\ v_f \rho_i & \text{if } \boldsymbol{\rho} \in \mathbf{D}_{i+1/2} \end{cases} \tag{9.7}$$

where $\mathbf{W}_{i+1/2}$, $\mathbf{L}_{i+1/2}$, $\mathbf{D}_{i+1/2}$, $i = 0, \cdots, n$, are $3(n+1)$ polyhedra in $[0,\, \rho_{\text{jam}}]^{n+2}$:

$$\begin{array}{ll} \mathbf{W}_{i+1/2} & = \{\boldsymbol{\rho} \in [0,\, \rho_{\text{jam}}]^{n+2} \mid \rho_{i+1} + \frac{v_f}{w_f}\rho_i > \rho_{\text{jam}}\ ,\ \rho_{i+1} > \rho_c\} \\ \mathbf{L}_{i+1/2} & = \{\boldsymbol{\rho} \in [0,\, \rho_{\text{jam}}]^{n+2} \mid \rho_i > \rho_c\ ,\ \rho_{i+1} \le \rho_c\} \\ \mathbf{D}_{i+1/2} & = \{\boldsymbol{\rho} \in [0,\, \rho_{\text{jam}}]^{n+2} \mid \rho_{i+1} + \frac{v_f}{w_f}\rho_i \le \rho_{\text{jam}}\ ,\ \rho_i \le \rho_c\} \end{array} \tag{9.8}$$

We note that we can express the polyhedra $\mathbf{W}_{i+1/2}$, $\mathbf{L}_{i+1/2}$, $\mathbf{D}_{i+1/2}$ in vector form:

$$\begin{array}{ll} \mathbf{W}_{i+1/2} & = \{\boldsymbol{\rho} \mid \boldsymbol{d}(1) \cdot [\rho_i, \rho_{i+1}, 1]^T > 0\ ,\ \boldsymbol{d}(3) \cdot [\rho_i, \rho_{i+1}, 1]^T > 0\} \\ \mathbf{L}_{i+1/2} & = \{\boldsymbol{\rho} \mid \boldsymbol{d}(2) \cdot [\rho_i, \rho_{i+1}, 1]^T > 0\ ,\ \boldsymbol{d}(3) \cdot [\rho_i, \rho_{i+1}, 1]^T \le 0\} \\ \mathbf{D}_{i+1/2} & = \{\boldsymbol{\rho} \mid \boldsymbol{d}(1) \cdot [\rho_i, \rho_{i+1}, 1]^T \le 0\ ,\ \boldsymbol{d}(2) \cdot [\rho_i, \rho_{i+1}, 1]^T \le 0\} \end{array} \tag{9.9}$$

with coefficients

$$\begin{array}{l} \boldsymbol{d}(1) = [(\rho_{\text{jam}} - \rho_c)/\rho_c,\, 1,\, -\rho_{\text{jam}}] \\ \boldsymbol{d}(2) = [1,\, 0,\, -\rho_c] \\ \boldsymbol{d}(3) = [0,\, 1,\, -\rho_c] \end{array} \tag{9.10}$$

Combining the Godunov scheme (9.4) and the Godunov flux in PWA form (9.7):

Lemma 9.2. *With a triangular flux function, the Godunov scheme at cell $i \in \{1, \cdots, n\}$ can be formulated as a Hybrid Automaton with linear components:*

- *mode* $m_i \in Q$ *with* $Q := \{1, \cdots, 9\}^2$
- *state* $\rho_i \in [0,\, \rho_{\text{jam}}]$
- *inputs* $(\rho_{i-1}^t,\, \rho_{i+1}^t) \in [0,\, \rho_{\text{jam}}]^2, \quad t \ge 0$
- *discrete dynamics* $\rho_i^{t+1} = \boldsymbol{L}(m_i) \cdot [\rho_{i-1}^t,\, \rho_i^t,\, \rho_{i+1}^t]^T + w(m_i) \quad$ *if* $\quad (\rho_{i-1}^t,\, \rho_i^t,\, \rho_{i+1}^t) \in P(\text{Dom}(m_i))$ *where* $L(\cdot) : Q \mapsto \mathbb{R}^3$ *and* $w(\cdot) : Q \mapsto \mathbb{R}$ *are defined in Table 9.1, and* $P(\cdot)$ *is the projection operator onto* $Vect(e_{i-1},\, e_i,\, e_{i+1})$.

m_i	$\mathrm{Dom}(m_i)$	$\boldsymbol{L}(m_i)$	$w(m_i)$	$\rho_i^{t+1} = \boldsymbol{L}(m_i) \cdot [\rho_{i-1}^t,\, \rho_i^t,\, \rho_{i+1}^t]^T + w(m_i)$
1	$\mathbf{W}_{i-1/2} \cap \mathbf{W}_{i+1/2}$	$\boldsymbol{L}(1) = [0,\ 1-\alpha w_f,\ \alpha w_f]$	$w(1) = 0$	$\rho_i^{t+1} = (1-\alpha\omega_f)\rho_i^t + \alpha\omega_f\rho_{i+1}^t$
2	$\mathbf{W}_{i-1/2} \cap \mathbf{L}_{i+1/2}$	$\boldsymbol{L}(2) = [0,\ 1-\alpha w_f,\ 0]$	$w(2) = \alpha w_f \rho_c$	$\rho_i^{t+1} = (1-\alpha\omega_f)\rho_i^t + \alpha\omega_f\rho_c$
3	$\mathbf{L}_{i-1/2} \cap \mathbf{W}_{i+1/2}$	$\boldsymbol{L}(3) = [0,\ 1,\ \alpha w_f]$	$w(3) = -\alpha w_f \rho_c$	$\rho_i^{t+1} = \rho_i^t + \alpha\omega_f\rho_{i+1}^t - \alpha\omega_f\rho_c$
4	$\mathbf{L}_{i-1/2} \cap \mathbf{D}_{i+1/2}$	$\boldsymbol{L}(4) = [0,\ 1-\alpha v_f,\ 0]$	$w(4) = \alpha v_f \rho_c$	$\rho_i^{t+1} = (1-\alpha v_f)\rho_i^t + \alpha v_f\rho_c$
5	$\mathbf{D}_{i-1/2} \cap \mathbf{W}_{i+1/2}$	$\boldsymbol{L}(5) = [\alpha v_f,\ 1,\ \alpha w_f]$	$w(5) = -\alpha w_f \rho_{\mathrm{jam}}$	$\rho_i^{t+1} = \alpha v_f\rho_{i-1}^t + \rho_i^t + \alpha\omega_f\rho_{i+1}^t - \alpha\omega_f\rho_{\mathrm{jam}}$
6	$\mathbf{D}_{i-1/2} \cap \mathbf{L}_{i+1/2}$	$\boldsymbol{L}(6) = [v_f,\ 1,\ 0]$	$w(6) = -\alpha v_f \rho_c$	$\rho_i^{t+1} = \alpha v_f\rho_{i-1}^t + \rho_i^t - \alpha v_f\rho_c$
7	$\mathbf{D}_{i-1/2} \cap \mathbf{D}_{i+1/2}$	$\boldsymbol{L}(7) = [v_f,\ 1-\alpha v_f,\ 0]$	$w(7) = 0$	$\rho_i^{t+1} = \alpha v_f\rho_{i-1}^t + (1-\alpha v_f)\rho_i^t$
8	$\mathbf{W}_{i-1/2} \cap \mathbf{D}_{i+1/2}$	$\boldsymbol{L}(8) = [0,\ 1-\alpha v_f-\alpha w_f,\ 0]$	$w(8) = \alpha w_f \rho_{\mathrm{jam}}$	$\rho_i^{t+1} = (1-\alpha v_f-\alpha w_f)\rho_i^t + \alpha w_f\rho_{\mathrm{jam}}$
9	$\mathbf{L}_{i-1/2} \cap \mathbf{L}_{i+1/2}$	$\boldsymbol{L}(9) = [0,\ 0,\ 0]$	$w(9) = 0$	$\rho_i^{t+1} = \rho_i^t$

Table 9.1: Godunov scheme w.r.t. discrete states m_i at cell i, *e.g.*, if $\boldsymbol{\rho} \in \mathrm{Dom}(\{m_i = 4\}) = \mathbf{L}_{i-1/2} \cap \mathbf{D}_{i+1/2} = \{\boldsymbol{\rho} \mid \rho_{i-1} > \rho_c,\, \rho_i \le \rho_c,\, \rho_{i+1} + \frac{v_f}{w_f}\rho_i \le \rho_{\mathrm{jam}}\}$, then $\rho_i^{t+1} = \boldsymbol{L}_4 \cdot [\rho_{i-1}^t,\, \rho_i^t,\, \rho_{i+1}^t]^T + w_4 = (1-\alpha v_f)\rho_i^t + \alpha v_f\rho_c$.

- *domain of the modes* $\mathrm{Dom}(m_i)$ *defined in the Table 9.1 and (9.8).*

We note that $\mathrm{Dom}(m_i)$ *refers to the subset of* $\mathbb{R}^{n+2}$ *in which the mode of cell* i *is* m_i*. Since the linear constraints that define* $\mathrm{Dom}(m_i)$ *(see Table 9.1) only concern variables* ρ_{i-1}, ρ_i, ρ_{i+1}*, the projection onto* $Vect(e_{i-1},\, e_i,\, e_{i+1})$ *contains all the information on the shape of* $\mathrm{Dom}(m_i)$*.*

Proof. We prove the result for $m_i = 4$, the other cases follow similarly. When $\boldsymbol{\rho} \in \mathrm{Dom}(\{m_i = 4\}) = \mathbf{L}_{i-1/2} \cap \mathbf{D}_{i+1/2}$ following the definition of $\mathrm{Dom}(m_i)$ in Table 9.1, we have $G(\rho_{i-1}, \rho_i) = q_c$ and $G(\rho_i,\, \rho_{i+1}) = v_f\rho_i$ from (9.7) then

$$\begin{aligned} \rho_i^+ &= \rho_i - \alpha\left(G(\rho_i, \rho_{i+1}) - G(\rho_{i-1}, \rho_i)\right) \\ &= \rho_i - \alpha(v_f\rho_i - q_c) = (1-\alpha v_f)\rho_i + \alpha q_c \end{aligned}$$

hence $\rho_i^+ = \boldsymbol{L}(4) \cdot [\rho_{i-1},\, \rho_i,\, \rho_{i+1}]^T + w(4)$ with $\boldsymbol{L}(4) := [0,\, 1-\alpha v_f,\, 0]$ and $w(4) := \alpha v_f \rho_c$ following the definitions of $\boldsymbol{L}(m_i)$ and $w(m_i)$ in Table 9.1. □

We note that the condition $(\rho_{i-1}^t,\, \rho_i^t,\, \rho_{i+1}^t) \in P(\mathrm{Dom}(m_i))$ in the discrete dynamics is a reset relation at each time step: the mode at time t is directly given by state $\boldsymbol{\rho}^t$.

Discretized system as a Hybrid system

The mode of each cell can be listed in a *vector* $\boldsymbol{m} \in \{1, \cdots, 9\}^n$ in which the i-th entry is the discrete state at cell i. We call it the *mode vector*. As a result, the domain of the mode vector $\boldsymbol{m} \in \{1, \cdots, 9\}^n$ is:

$$\mathrm{Dom}(\boldsymbol{m}) = \bigcap_{i=1}^{n} \mathrm{Dom}(m_i) \tag{9.11}$$

[2]In this description, the mode m_i takes on values in a finite set $Q = \{1, \cdots, 9\}$ for completeness. We will see in Section 9.3 that the modes $m_i = 8$ and $m_i = 9$ are *not accepted*.

For example, if $n = 2$, then the state $\boldsymbol{\rho} = [\rho_0, \rho_1, \rho_2, \rho_3]$ is in $[0,\ \rho_{\text{jam}}]^4$ with boundary cells ρ_0 and ρ_3 and the mode vector $\boldsymbol{m}$ is in $\{1, \cdots, 9\}^4$. More specifically:

$$\begin{aligned}
&\text{Dom}(\{\boldsymbol{m} = (2,3)\}) \\
&= \text{Dom}(\{m_1 = 2\}) \cap \text{Dom}(\{m_2 = 3\}) \\
&= (\mathbf{W}_{1/2} \cap \mathbf{L}_{1+1/2}) \cap (\mathbf{L}_{1+1/2} \cap \mathbf{W}_{2+1/2}) \\
&= \mathbf{W}_{1/2} \cap \mathbf{L}_{1+1/2} \cap \mathbf{W}_{2+1/2} \\
&= \{\boldsymbol{\rho} \in [0,\ \rho_{\text{jam}}]^4 \mid \rho_1 + \tfrac{v_f}{w_f}\rho_0 > \rho_{\text{jam}},\ \rho_1 > \rho_c, \rho_2 \leq \rho_c, \\
&\qquad \rho_3 + \tfrac{v_f}{w_f}\rho_2 > \rho_{\text{jam}}\}
\end{aligned} \tag{9.12}$$

We will show later that the subsets $\text{Dom}(\boldsymbol{m})$'s form a partition of $[0, \rho_{\text{jam}}]^{n+2}$.

For each mode vector $\boldsymbol{m}$, we construct the matrix $A_{\boldsymbol{m}} \in \mathbb{R}^{(n+2)\times(n+2)}$, and the row vectors $b_{\boldsymbol{m}}$, $c^t \in \mathbb{R}^{n+2}$ in the form:

$$A_{\boldsymbol{m}} = \begin{bmatrix} 0 & \cdots & 0 \\ L(m_1) & & \\ & \ddots & \\ & & L(m_n) \\ 0 & \cdots & 0 \end{bmatrix},\ b_{\boldsymbol{m}} = \begin{bmatrix} 0 \\ w(m_1) \\ \vdots \\ w(m_n) \\ 0 \end{bmatrix},\ c^t = \begin{bmatrix} u(t) \\ 0 \\ \vdots \\ 0 \\ d(t) \end{bmatrix} \tag{9.13}$$

where $L(m_i)$, $w(m_i)$ are defined in Table 9.1, and $u(t)$, $d(t)$ are the upstream and downstream densities respectively. This leads to one of the main results of the chapter:

Proposition 9.1. *The discretized LWR equation using the Godunov scheme and with a triangular flux function is an Autonomous Hybrid Automaton with affine components:*

- *discrete state* $\boldsymbol{m} \in \{1, \cdots, 9\}^n$
- *state* $\boldsymbol{\rho}^t \in [0,\ \rho_{\text{jam}}]^{n+2}$ *at time* t
- *inputs* $(u(t), d(t)) \in [0,\ \rho_{\text{jam}}]^2$
- *discrete dynamics* $\boldsymbol{\rho}^{t+1} = A_{\boldsymbol{m}}\boldsymbol{\rho}^t + b_{\boldsymbol{m}} + c^t \quad$ *if* $\quad \boldsymbol{\rho}^t \in \text{Dom}(\boldsymbol{m})$
- *domain of the discrete states* $\text{Dom}(\boldsymbol{m})$ *defined in (9.11).*

Proof. The formulation as a Hybrid Automaton is obtained by stacking the states and modes in the Hybrid Automaton formulation of the Godunov scheme into a vector, and the linear transformations into a matrix. □

Finally, we note that the condition $\boldsymbol{\rho}^t \in \text{Dom}(\boldsymbol{m})$ in the discrete dynamics is a reset relation at each time step: the mode at time t is directly given by state $\boldsymbol{\rho}^t$.

Algorithm 9.1 Find the mode vector: **rho2m($\boldsymbol{\rho}$)**. The parameters $\boldsymbol{d}(1)$, $\boldsymbol{d}(2)$, $\boldsymbol{d}(3) \in \mathbb{R}^3$ in equation (9.10) describe the domain of each mode vector (see Table 9.1 and (9.9), (9.10), (9.11))

Require: current state $\boldsymbol{\rho} = [\rho_0, \cdots, \rho_{n+1}] \in [0, \rho_{\text{jam}}]^{n+2}$
1. **for** $i \in \{0, \cdots, n\}$:
2. $\quad \mathbf{x} = [\rho_i,\ \rho_{i+1},\ 1]^T$
3. $\quad I = [\boldsymbol{d}(1)x > 0,\ \boldsymbol{d}(2)x > 0,\ \boldsymbol{d}(3)x > 0] \in \{0,\ 1\}^3$
4. $\quad$ **if** $\boldsymbol{I}(1) \wedge \boldsymbol{I}(3)$ **then** $s(i) = W$ $\quad \backslash\backslash \boldsymbol{\rho} \in \mathbf{W}_{i+1/2}$
5. $\quad$ **if** $\boldsymbol{I}(2) \wedge \neg\boldsymbol{I}(3)$ **then** $s(i) = L$ $\quad \backslash\backslash \boldsymbol{\rho} \in \mathbf{L}_{i+1/2}$
6. $\quad$ **if** $\neg\boldsymbol{I}(1) \wedge \neg\boldsymbol{I}(2)$ **then** $s(i) = D$ $\quad \backslash\backslash \boldsymbol{\rho} \in \mathbf{D}_{i+1/2}$
7. **for** $i \in \{0, \cdots, n\}$:
8. $\quad$ **if** $\{s(i) = W\} \wedge \{s(i+1) = W\}$ **then** $m_i = 1$
9. $\quad$ **if** $\{s(i) = W\} \wedge \{s(i+1) = L\}$ **then** $m_i = 2$
10. $\quad$ **if** $\{s(i) = L\} \wedge \{s(i+1) = W\}$ **then** $m_i = 3$
11. $\quad$ **if** $\{s(i) = L\} \wedge \{s(i+1) = D\}$ **then** $m_i = 4$
12. $\quad$ **if** $\{s(i) = D\} \wedge \{s(i+1) = W\}$ **then** $m_i = 5$
13. $\quad$ **if** $\{s(i) = D\} \wedge \{s(i+1) = L\}$ **then** $m_i = 6$
14. $\quad$ **if** $\{s(i) = D\} \wedge \{s(i+1) = D\}$ **then** $m_i = 7$
15. **return** $\boldsymbol{m} = [m_1, \cdots, m_n] \in \{1, \cdots, 7\}^n$

9.3 Description of the mode vectors

Accepted mode vectors

The following analysis is motivated by the fact that $\text{Dom}(\boldsymbol{m}) = \emptyset$ for some values of $\boldsymbol{m}$, which means that some of the mode vectors $\boldsymbol{m}$'s are not *accepted* by the system.

Definition 9.1. *We say that a mode vector $\boldsymbol{m}$ is* accepted *by the system if and only if its domain* $\text{Dom}(\boldsymbol{m})$ *is not empty.*

Proposition 9.2. *The mode vector $\boldsymbol{m} \in \{1, \cdots, 9\}^n$ is accepted by the system if and only if we have the following two conditions*

$$m_i \in \{1, \cdots, 7\},\ \forall\, i \in \{1, \cdots, n\} \tag{9.14}$$

$$\forall\, i \in \{1, \cdots, n-1\}, \quad m_{i+1} \in \begin{cases} \{1, 2\} & \text{if } m_i \in \{1, 3, 5\} \\ \{3, 4\} & \text{if } m_i \in \{2, 6\} \\ \{5, 6, 7\} & \text{if } m_i \in \{4, 7\} \end{cases} \tag{9.15}$$

Proof. From (9.8), it can be seen that $\mathbf{W}_{i-1/2} \cap \mathbf{D}_{i+1/2} = \mathbf{L}_{i-1/2} \cap \mathbf{L}_{i+1/2} = \emptyset$ for all $i = 1, \cdots, n$. Hence $\text{Dom}(\{m_i = 8\}) = \text{Dom}(\{m_i = 9\}) = \emptyset$ (see Table 9.1). In other words, $\boldsymbol{m}$ is not accepted if it has an entry in $\{8, 9\}$ which gives the first condition.

We note that for a fixed i, the polyhedra $\mathbf{W}_{i+1/2}$, $\mathbf{L}_{i+1/2}$, $\mathbf{D}_{i+1/2}$ partition $[0,\ \rho_{\text{jam}}]^{n+2}$. Since $\text{Dom}(m_i) \cap \text{Dom}(m_{i+1})$ is of the form:

$$\begin{aligned}&\text{Dom}(m_i) \cap \text{Dom}(m_{i+1})\\ &= (\mathbf{P}_{i-1/2} \cap \mathbf{P}_{i+1/2}) \cap (\mathbf{P}'_{i+1/2} \cap \mathbf{P}_{i+1+1/2}) \subset \mathbf{P}_{i+1/2} \cap \mathbf{P}'_{i+1/2}\end{aligned}$$

with $\mathbf{P}_{i+1/2},\ \mathbf{P}'_{i+1/2} \in \{\mathbf{W}_{i+1/2},\ \mathbf{L}_{i+1/2},\ \mathbf{D}_{i+1/2}\}$, then $\boldsymbol{m}$ is accepted if $\mathbf{P}_{i+1/2} = \mathbf{P}'_{i+1/2}$. In other words, $\text{Dom}(m_i) = \mathbf{P}_{i-1/2} \cap \mathbf{P}_{i+1/2}$ and $\text{Dom}(m_{i+1}) = \mathbf{P}'_{i+1/2} \cap \mathbf{P}_{i+1+1/2}$ must overlap. This gives condition (9.15).

Reciprocally, if $\boldsymbol{m}$ satisfies conditions (9.14) and (9.15), then we have overlaps between $\text{Dom}(m_i)$ and $\text{Dom}(m_{i+1})$. Hence $\text{Dom}(\boldsymbol{m})$ is of the form

$$\begin{aligned}\text{Dom}(\boldsymbol{m}) &= \textstyle\bigcap_{i=0}^{n} \mathbf{P}_{i+1/2}\\ \mathbf{P}_{i+1/2} &\in \{\mathbf{W}_{i+1/2},\ \mathbf{L}_{i+1/2},\ \mathbf{D}_{i+1/2}\},\ i = 0, \cdots, n\end{aligned} \tag{9.16}$$

The intersection of any pair of two consecutive polyhedra in (9.16) has to be among the first seven subsets in Table 9.1. Hence for all $i = 1, \cdots, n$, the projection of $\text{Dom}(\boldsymbol{m})$ onto $Vect(e_{i-1},\ e_i,\ e_{i+1})$ is one of the 7 subsets of $\mathbb{R}^3$ shown in Figure 9.3, which are all non empty. Hence $\text{Dom}(\boldsymbol{m})$ is the product of nonempty spaces, hence it is nonempty. □

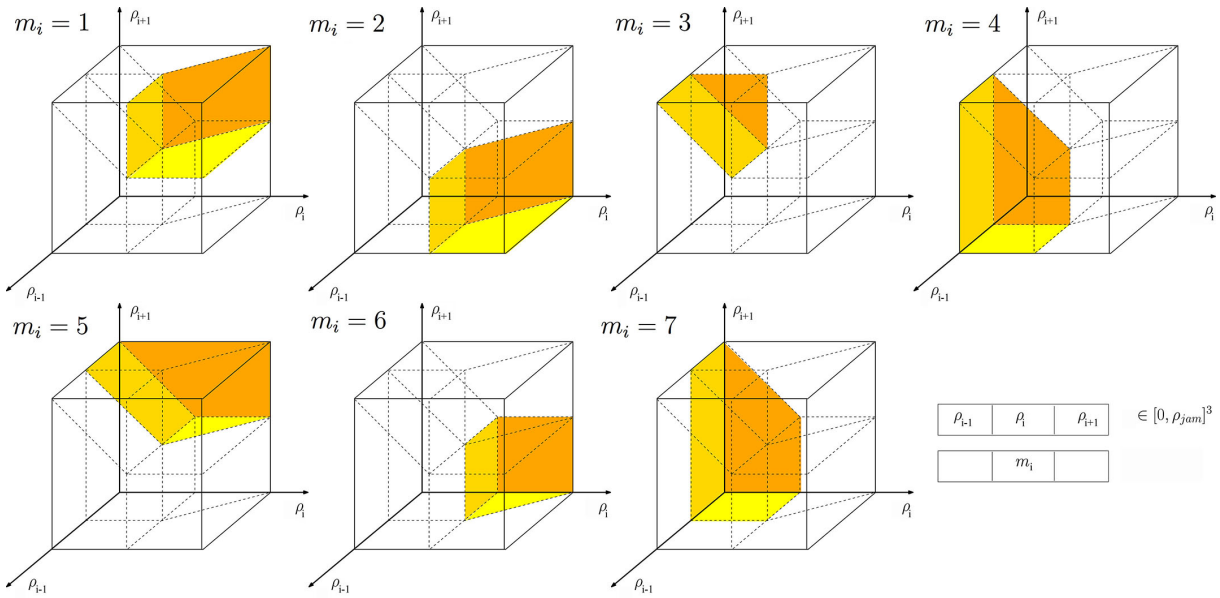

Figure 9.3: Projection of $\text{Dom}(m_i)$ onto $Vect(e_{i-1},\ e_i,\ e_{i+1})$ for $i \in \{1, \cdots, 7\}$. For example, in the top left figure, if $(\rho_{i-1},\ \rho_i,\ \rho_{i+1})$ is in the orange polyhedron, then $\boldsymbol{\rho} \in \mathbf{W}_{i-1/2} \cap \mathbf{W}_{i+1/2} = \text{Dom}(\{m_i = 1\})$, the mode m_i is 1 (see Table 9.1).

From the analysis above, we also conclude that under conditions (9.14) and (9.15), the domain of an accepted mode vector can be decomposed in the form (9.16). This is illustrated in the derivation of $\text{Dom}(\{\boldsymbol{m} = (2, 3)\})$ in example (9.12) above.

From (9.14), the space of discrete states of the Godunov scheme at each cell is reduced to $\{1, \cdots, 7\}$ and the space in which the mode vector $\boldsymbol{m}$ lies is reduced to $\{1, \cdots, 7\}^n$.

Definition 9.2. *For an accepted mode vector* $\boldsymbol{m}$ *and the associated* $\mathrm{Dom}(\boldsymbol{m}) = \bigcap_{i=0}^{n} \boldsymbol{P}_{i+1/2}$, *a* mode string $\boldsymbol{s} = s(0)s(1)s(2)\cdots s(n)$ *is associated with* $\boldsymbol{m}$ *if* $s(i) = W$ *(resp.* L, D*) if* $\boldsymbol{P}_{i+1/2} = \boldsymbol{W}_{i+1/2}$ *(resp.* $\boldsymbol{L}_{i+1/2}$, $\boldsymbol{D}_{i+1/2}$*) and a mode string is accepted if and only if* $s(i)s(i+1) \in \{WW, WL, LW, LD, DW, DL, DD\}$ *for all* i*, from the analysis done in Proposition 9.2.*

Proposition 9.3. *The number of accepted mode vectors is asymptotically* $3.1778 \cdot (2.2470)^n$.

Proof. We count the number of accepted mode strings recursively on the length k of the string. Let N_k be the number of accepted strings, Figure 9.4 shows the 16 accepted strings of length 3. Let us denote by w_k (resp. l_k, d_k) the number of accepted strings which last element is W (resp. L, D). Then for all $k \geq 0$

$$\begin{aligned} w_0 &= l_0 = d_0 = 1 \\ w_{k+1} &= w_k + l_k + d_k \\ l_{k+1} &= w_k + d_k \\ d_{k+1} &= l_k + d_k \end{aligned}$$

$$\Longrightarrow \quad \begin{bmatrix} w_k \\ l_k \\ d_k \end{bmatrix} = A^k \times \begin{bmatrix} w_0 \\ l_0 \\ d_0 \end{bmatrix}, \; A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{9.17}$$

hence, $N_k = w_k + l_k + d_k = e^T A^k e$ with $e^T = [1\,1\,1]$. Diagonalizing the matrix A gives $A = VDV^{-1}$ with $D := \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3)$ where λ_1, λ_2, λ_3 are the eigenvalues of A in increasing order. Since λ_3 is the only eigenvalue above 1 in absolute value, we have:

$$\begin{aligned} &D^k = \mathrm{diag}(\lambda_1^k, \lambda_2^k, \lambda_3^k) \sim \mathrm{diag}(0, 0, \lambda_3^k) \quad \text{when} \quad k \longrightarrow +\infty \\ &\text{hence} \quad e^T A^k e \approx e^T V \,\mathrm{diag}(0, 0, \lambda_3^k) V^{-1} e = \lambda_3^k (V^T e)_3 (V^{-1} e)_3 \\ &\approx 3.1778 \cdot (2.2470)^k \end{aligned}$$

□

Proposition 9.4. *The polyhedra* $\mathrm{Dom}(\boldsymbol{m})$ *associated with accepted mode vectors* $\boldsymbol{m}$ *form a partition of* $[0, \rho_{\mathrm{jam}}]^{n+2}$.

Proof. Let $\boldsymbol{m}$ and $\boldsymbol{m}'$ be two distinct accepted mode vectors and $\boldsymbol{s}$, $\boldsymbol{s}'$ the associated strings. We pick $i \in \{0, \cdots, n\}$ such that $s(i) \neq s'(i)$. Then $\mathrm{Dom}(\boldsymbol{m}) \subset \mathbf{P}_{i+1/2}$ and $\mathrm{Dom}(\boldsymbol{m}') \subset \mathbf{P}'_{i+1/2}$, where $\mathbf{P}_{i+1/2}$ and $\mathbf{P}'_{i+1/2}$ are two distinct polyhedra among $\mathbf{W}_{i+1/2}$, $\mathbf{L}_{i+1/2}$, $\mathbf{D}_{i+1/2}$. Hence $\mathrm{Dom}(\boldsymbol{m})$ and $\mathrm{Dom}(\boldsymbol{m}')$ are disjoint. And for any $\boldsymbol{\rho} \in [0, \rho_{\mathrm{jam}}]^{n+2}$, we can find its associated accepted mode vector $\boldsymbol{m}$ such that $\boldsymbol{\rho} \in \mathrm{Dom}(\boldsymbol{m})$, hence the different $\mathrm{Dom}(\boldsymbol{m})$ span the whole state space. □

Algorithm 9.2 mode vector $\boldsymbol{m}$ to mode string: **m2s($\boldsymbol{m}$)**

Require: accepted mode vector $\boldsymbol{m}$
1. **if** $m_1 \in \{1,2\}$ **then** $s(0) = W$ $\quad \backslash\backslash \mathbf{P}_{1/2} = \mathbf{W}_{1/2}$ in (9.16)
2. **if** $m_1 \in \{3,4\}$ **then** $s(0) = L$ $\quad \backslash\backslash \mathbf{P}_{1/2} = \mathbf{L}_{1/2}$ in (9.16)
3. **if** $m_1 \in \{5,6,7\}$ **then** $s(0) = D$ $\quad \backslash\backslash \mathbf{P}_{1/2} = \mathbf{D}_{1/2}$ in (9.16)
4. **for** $i \in \{1, \cdots, n\}$:
5. $\quad m_i \in \{1,3,5\}$ **then** $s(i) = W$ $\quad \backslash\backslash \mathbf{P}_{i+1/2} = \mathbf{W}_{i+1/2}$ in (9.16)
6. $\quad m_i \in \{2,6\}$ **then** $s(i) = L$ $\quad \backslash\backslash \mathbf{P}_{i+1/2} = \mathbf{L}_{i+1/2}$ in (9.16)
7. $\quad m_i \in \{4,7\}$ **then** $s(i) = D$ $\quad \backslash\backslash \mathbf{P}_{i+1/2} = \mathbf{D}_{i+1/2}$ in (9.16)
8. **return** the mode string $s(0)s(1)\cdots s(n)$

Algorithm 9.3 mode string to mode vector: **s2m(**$s(0)\cdots s(n)$**)**

Require: accepted mode string $s(0)\cdots s(n)$
1. apply lines 8 to 16 of Algorithm 9.2
2. **return** the mode vector $\boldsymbol{m}$

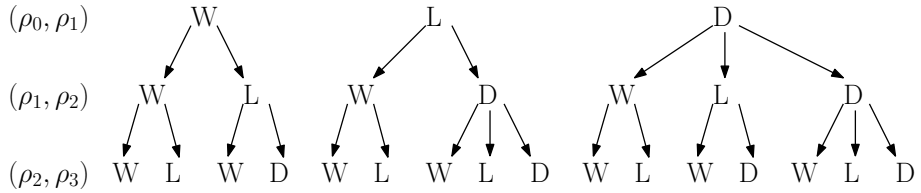

Figure 9.4: The sixteen *accepted mode strings* for the first three pairs (ρ_0, ρ_1), (ρ_1, ρ_2), and (ρ_2, ρ_3). For more details, see Propositions 9.2 and 9.3.

Minimal representation

We now introduce the concepts of *minimal representation* and *adjacent polyhedra*.

Definition 9.3 (Faces of a polyhedron)**.** *A supportive hyperplane of a closed convex set $\boldsymbol{C}$ is a hyperplane $\partial\boldsymbol{H}$ such that $\boldsymbol{C} \cap \partial\boldsymbol{H} \neq \emptyset$ and $\boldsymbol{C} \subseteq \boldsymbol{H}$, where $\boldsymbol{H}$ is one of the two closed half-spaces (associated with the hyperplane). Given a (closed) polyhedron $\boldsymbol{P}$, the intersection with any supportive hyperplane is a face of $\boldsymbol{P}$. Moreover, a vertex is a zero-dimension face, an edge a one-dimension face, and a facet is a face of dimension $d-1$ if $\boldsymbol{P}$ is of dimension d. For a full-dimensional polyhedron, a facet is of dimension $n+1$ (recall that the space $[0, \rho_{\text{jam}}]^{n+2}$ is of dimension $n+2$).*

Definition 9.4 (Minimal H-representation)**.** *There exist infinitely many H-descriptions of a (closed) convex polytope. For a full-dimensional convex polytope, the minimal H-description is unique and is given by the set of the facet-defining half-spaces [75].*

We now want to find the minimal representation of $\text{Dom}(\boldsymbol{m}) = \bigcap_{i=0}^{n} \mathbf{P}_{i+1/2}$ for all accepted modes $\boldsymbol{m}$. Each one of the $3(n+1)$ polyhedra $\mathbf{W}_{i+1/2}$, $\mathbf{L}_{i+1/2}$, $\mathbf{D}_{i+1/2}$, $i = 0, \cdots, n$ defined

in (9.8) is intersection of two half-spaces:

$$\begin{array}{l} \mathbf{W}_{i+1/2} = \mathbf{H}_{i+1/2} \cap \mathbf{H}_{i+1} \\ \mathbf{L}_{i+1/2} = \mathbf{H}_i \cap \mathbf{H}_{i+1}^c \\ \mathbf{D}_{i+1/2} = \mathbf{H}_i^c \cap \mathbf{H}_{i+1/2}^c \end{array} \tag{9.18}$$

where

$$\begin{array}{l} \mathbf{H}_i = \{\boldsymbol{\rho} \in [0,\, \rho_{\text{jam}}]^{n+2} \mid \rho_i > \rho_c\},\ i = 0, \cdots, n+1 \\ \mathbf{H}_{i+1/2} = \{\boldsymbol{\rho} \in [0,\, \rho_{\text{jam}}]^{n+2} \mid \rho_{i+1} + \frac{v_f}{w_f}\rho_i > \rho_{\text{jam}}\},\ i = 0, \cdots, n \end{array} \tag{9.19}$$

and $\mathbf{H}_i^c$, $\mathbf{H}_{i+1/2}^c$ are the complementary of $\mathbf{H}_i$ and $\mathbf{H}_{i+1/2}$ respectively. The projections of these half-spaces on $Vect(e_i,\, e_{i+1})$ are illustrated in Figure 9.5.

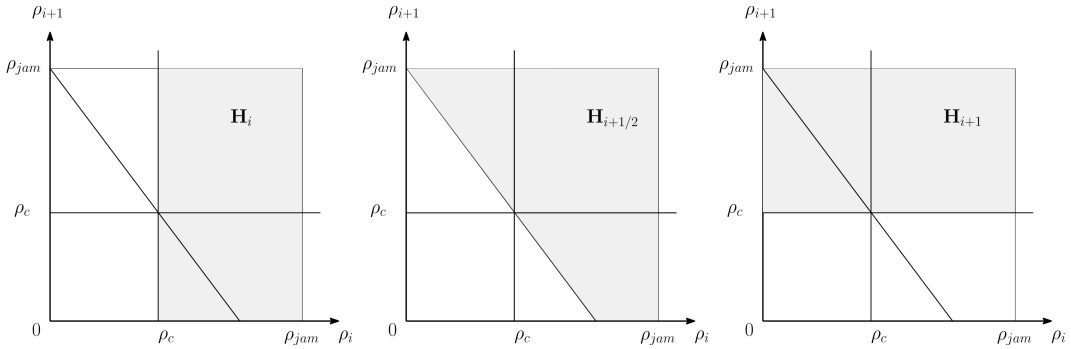

Figure 9.5: Projection of the half-spaces $\mathbf{H}_i$, $\mathbf{H}_{i+1/2}$, $\mathbf{H}_{i+1}$ on the plane $Vect(e_i,\, e_{i+1})$.

In example (9.12), we have:

$$\begin{array}{l} \text{Dom}(\{\boldsymbol{m} = \{2,3\}\}) \\ = \mathbf{W}_{1/2} \cap \mathbf{L}_{1+1/2} \cap \mathbf{W}_{2+1/2} \\ = (\mathbf{H}_{1/2} \cap \mathbf{H}_1) \cap (\mathbf{H}_1 \cap \mathbf{H}_2^c) \cap (\mathbf{H}_{2+1/2} \cap \mathbf{H}_3) \\ = \mathbf{H}_{1/2} \cap \mathbf{H}_1 \cap \mathbf{H}_2^c \cap \mathbf{H}_{2+1/2} \cap \mathbf{H}_3 \\ = \mathbf{H}_{1/2} \cap \mathbf{H}_1 \cap \mathbf{H}_2^c \cap \mathbf{H}_{2+1/2} \end{array}$$

Since $\mathbf{H}_2^c \cap \mathbf{H}_{2+1/2} \subset \mathbf{H}_3$, we can remove $\mathbf{H}_3$ from the intersection. After removing this redundant constraint, the last equality gives the *minimal representation* of $\text{Dom}(\{\boldsymbol{m} = \{2,3\}\})$.

While finding the minimal representation of a nonempty polyhedron can be difficult in general, it is easy for the polyhedra $\text{Dom}(\boldsymbol{m})$ associated with accepted mode vectors $\boldsymbol{m}$. In the form $\text{Dom}(\boldsymbol{m}) = \bigcap_{i=0}^{n} \mathbf{P}_{i+1/2}$, we sequentially derive the minimal representation of each polyhedron of the decreasing sequence $\{\bigcap_{i=0}^{k} \mathbf{P}_{i+1/2}\}_{k\geq 0}$ by successively adding the non-redundant constraints in $\mathbf{P}_{k+1/2} \in \{\mathbf{W}_{k+1/2},\, \mathbf{L}_{k+1/2},\, \mathbf{D}_{k+1/2}\}$ to the minimal representation of $\bigcap_{i=0}^{k-1} \mathbf{P}_{i+1/2}$. The minimal representation is given by Algorithm 9.4.

Algorithm 9.4 Minimum representation of Dom($\boldsymbol{m}$): **minRep($\boldsymbol{m}$)**

Require: accepted mode vector $\boldsymbol{m}$
1. $\mathcal{H} = \{\}$
2. **if** $m_1 \in \{1,2\}$ **then** $\mathcal{H} = \mathcal{H} \cup \{\mathbf{H}_{1/2}, \mathbf{H}_1\}$
3. **if** $m_1 \in \{3,4\}$ **then** $\mathcal{H} = \mathcal{H} \cup \{\mathbf{H}_0, \mathbf{H}_1^c\}$
4. **if** $m_1 \in \{5,6,7\}$ **then** $\mathcal{H} = \mathcal{H} \cup \{\mathbf{H}_0^c, \mathbf{H}_{1/2}^c\}$
5. **for** $k \in \{1, \cdots, n\}$:
6. **if** $m_k = 1$ **then** $\mathcal{H} = \mathcal{H} \cup \{\mathbf{H}_{k+1}\}$
7. **if** $m_k = 2$ **then** $\mathcal{H} = \mathcal{H} \cup \{\mathbf{H}_{k+1}^c\}$
8. **if** $m_k = 3$ **then** $\mathcal{H} = \mathcal{H} \cup \{\mathbf{H}_{k+1/2}\}$
9. **if** $m_k = 4$ **then** $\mathcal{H} = \mathcal{H} \cup \{\mathbf{H}_{k+1/2}^c\}$
10. **if** $m_k = 5$ **then** $\mathcal{H} = \mathcal{H} \cup \{\mathbf{H}_{k+1/2}, \mathbf{H}_{k+1}\}$
11. **if** $m_k = 6$ **then** $\mathcal{H} = \mathcal{H} \backslash \{\mathbf{H}_{k-1}^c\} \cup \{\mathbf{H}_k, \mathbf{H}_{k+1}^c\}$
12. **if** $m_k = 7$ **then** $\mathcal{H} = \mathcal{H} \backslash \{\mathbf{H}_{k-1/2}^c\} \cup \{\mathbf{H}_{k+1/2}^c, \mathbf{H}_k^c\}$
13. **return** the minimal representation $\mathcal{H}$

Lemma 9.3. *The following inclusions for the half-spaces $\boldsymbol{H}_k$, $\boldsymbol{H}_{k+1/2}$, $\boldsymbol{H}_{k+1}$ for $k \in \{0, \cdots, n\}$ hold:*

$$\begin{array}{ll} \boldsymbol{H}_k \cap \boldsymbol{H}_{k+1} & \subset \boldsymbol{H}_{k+1/2} \\ \boldsymbol{H}_k^c \cap \boldsymbol{H}_{k+1/2} & \subset \boldsymbol{H}_{k+1} \\ \boldsymbol{H}_{k+1/2}^c \cap \boldsymbol{H}_{k+1} & \subset \boldsymbol{H}_k^c \\ \boldsymbol{H}_k^c \cap \boldsymbol{H}_{k+1}^c & \subset \boldsymbol{H}_{k+1/2}^c \end{array} \tag{9.20}$$

Proof. The proof is left to the reader. See Figure 9.5 for an illustration of these inclusions. □

Proposition 9.5. *For every accepted mode vector $\boldsymbol{m}$, Algorithm 9.4 returns the minimal representation of the closure of* Dom($\boldsymbol{m}$).

Proof. We fix an accepted mode vector $\boldsymbol{m}$ and we fix the associated decomposition (9.16), which gives the sequence $\mathbf{P}_{k+1/2}$, $k \in \{0, \cdots, n\}$. Let $\mathcal{H}$ be the set of half-spaces (or linear inequalities) in $[0, \rho_{\text{jam}}]^{n+2}$ defined in algorithm 9.4. First, we express Dom($\boldsymbol{m}$) in the form Dom($\boldsymbol{m}$) $= \bigcap_{i=0}^{n} \mathbf{P}_{i+1/2}$, then we prove by induction that at the k-th iteration of the for loop in algorithm 9.4, the intersection of all the half-spaces in the current $\mathcal{H}$ is the minimal representation of $\bigcap_{i=0}^{k} \mathbf{P}_{i+1/2}$.

Initialization $k = 1$: if $m_1 \in \{1,2\}$, then we have $\mathcal{H} = \{\mathbf{H}_{1/2}, \mathbf{H}_1\}$ from the algorithm and $\mathbf{P}_{1/2} = \mathbf{W}_{1/2}$ from Table 9.1. The expression $\cap_{\mathbf{H} \in \mathcal{H}} \mathbf{H} = \mathbf{H}_{1/2} \cap \mathbf{H}_1$ is clearly the minimal representation of $\mathbf{W}_{1/2}$ from (9.18). The cases $m_1 \in \{3,4\}$ and $m_1 \in \{5,6,7\}$ follow similarly.

Step k: The algorithm provides $\mathcal{H}^-$ and $\mathcal{H}$ which are the minimal representations of $\bigcap_{i=0}^{k-2} \mathbf{P}_{i+1/2}$ and $\bigcap_{i=0}^{k-1} \mathbf{P}_{i+1/2}$ respectively. We want to show that the algorithm updates $\mathcal{H}$ to $\mathcal{H}^+$, such that $\mathcal{H}^+$ is the minimal representation of $\bigcap_{i=0}^{k} \mathbf{P}_{i+1/2}$. We have 7 cases:

a) If $m_k = 1$, then from Table 9.1 and (9.18),

$$\begin{aligned}\mathbf{P}_{k-1/2} &= \mathbf{W}_{k-1/2} = \mathbf{H}_{k-1/2} \cap \mathbf{H}_k \\ \mathbf{P}_{k+1/2} &= \mathbf{W}_{k+1/2} = \mathbf{H}_{k+1/2} \cap \mathbf{H}_{k+1}\end{aligned}$$

in the expression (9.16), hence $\mathcal{H} \subset \mathcal{H}^- \cup \{\mathbf{H}_{k-1/2}, \mathbf{H}_k\}$. In this case, algorithm 9.4 adds constraint $\mathbf{H}_{k+1}$ to $\mathcal{H}$, so $\mathcal{H}^+ \subset \mathcal{H}^- \cup \{\mathbf{H}_{k-1/2}, \mathbf{H}_k, \mathbf{H}_{k+1}\}$. Then

$$\begin{aligned}\mathbf{H}_{k-1/2} \cap \mathbf{H}_k \cap \mathbf{H}_{k+1} &= (\mathbf{H}_{k-1/2} \cap \mathbf{H}_k) \cap (\mathbf{H}_{k+1} \cap \mathbf{H}_{k+1/2}) \\ &= \mathbf{W}_{k-1/2} \cap \mathbf{W}_{k+1/2}\end{aligned}$$

where the third equality is from $\mathbf{H}_k \cap \mathbf{H}_{k+1} \subset \mathbf{H}_{k+1/2}$ in (9.20). Hence $\mathcal{H}^+$ is a representation of $\cap_{i=0}^{k} \mathbf{P}_{i+1/2}$. Finally, $\mathcal{H}^+$ is minimal because the added constraint $\mathbf{H}_{k+1}$ is the only constraint on ρ_{k+1}, so it is not redundant with the constraints in $\mathcal{H}$.

b) If $m_k = 2$, then $\mathbf{P}_{k-1/2}$ and $\mathbf{P}_{k+1/2}$ in the expression (9.16) are:

$$\begin{aligned}\mathbf{P}_{k-1/2} &= \mathbf{W}_{k-1/2} = \mathbf{H}_{k-1/2} \cap \mathbf{H}_k \\ \mathbf{P}_{k+1/2} &= \mathbf{L}_{k+1/2} = \mathbf{H}_k \cap \mathbf{H}_{k+1}^c\end{aligned}$$

and constraint $\mathbf{H}_{k+1}^c$ is added to $\mathcal{H}$ in algorithm 9.4, so $\mathcal{H}^+ \subset \mathcal{H}^- \cup \{\mathbf{H}_{k-1/2}, \mathbf{H}_k, \mathbf{H}_{k+1}^c\}$, and

$$\begin{aligned}\mathbf{H}_{k-1/2} \cap \mathbf{H}_k \cap \mathbf{H}_{k+1}^c &= (\mathbf{H}_{k-1/2} \cap \mathbf{H}_k) \cap (\mathbf{H}_k \cap \mathbf{H}_{k+1}^c) \\ &= \mathbf{P}_{k-1/2} \cap \mathbf{P}_{k+1/2}\end{aligned}$$

so $\cap_{\mathbf{H} \in \mathcal{H}^+} \mathbf{H} = \cap_{i=0}^{k} \mathbf{P}_{i+1/2}$, *i.e.* $\mathcal{H}^+$ is a representation of $\cap_{i=0}^{k} \mathbf{P}_{i+1/2}$. This is the minimal representation because the added constraint $\mathbf{H}_{k+1}^c$ is the only constraint on ρ_{k+1}.

c) If $m_k = 3$, the analysis is similar to case $m_k = 1$.

d) If $m_k = 4$, the analysis is similar to case $m_k = 2$.

e) If $m_k = 5$, then $\mathbf{P}_{k-1/2} = \mathbf{D}_{k-1/2} = \mathbf{H}_{k-1}^c \cap \mathbf{H}_{k-1/2}^c$ and $\mathbf{P}_{k+1/2} = \mathbf{W}_{k+1/2} = \mathbf{H}_{k+1/2} \cap \mathbf{H}_{k+1}$ in expression (9.16). Algorithm 9.4 adds constraints $\mathbf{H}_{k+1/2}$, $\mathbf{H}_{k+1}$ to $\mathcal{H}$, hence $\mathcal{H}^+$ is a representation of $\cap_{i=0}^{k} \mathbf{P}_{i+1/2}$. It is easy to see that the constraints $\mathbf{H}_{k+1/2} \cap \mathbf{H}_{k+1}$ are not redundant, hence $\mathcal{H}^+$ is the minimal representation of $\cap_{i=0}^{k} \mathbf{P}_{i+1/2}$.

f) If $m_k = 6$, then $\mathbf{P}_{k-1/2} = \mathbf{D}_{k-1/2} = \mathbf{H}_{k-1}^c \cap \mathbf{H}_{k-1/2}^c$ and $\mathbf{P}_{k+1/2} = \mathbf{L}_{k+1/2} = \mathbf{H}_k \cap \mathbf{H}_{k+1}^c$ in expression (9.16). We have $\mathcal{H} \subset \mathcal{H}^- \cup \{\mathbf{H}_{k-1}^c, \mathbf{H}_{k-1/2}^c\}$. Algorithm 9.4 removes constraint $\mathbf{H}_{k-1}^c$ from $\mathcal{H}$ (if $\mathcal{H}$ contains it) and adds constraints $\mathbf{H}_k$, $\mathbf{H}_{k+1}^c$, hence $\mathcal{H}^+ \subset \mathcal{H}^- \cup \{\mathbf{H}_{k-1/2}^c, \mathbf{H}_k, \mathbf{H}_{k+1}^c\}$. The only potential redundancies in $\mathcal{H}^+$ would be between $\mathbf{H}_{k-1/2}^c$ and the newly added constraints $\mathbf{H}_k, \mathbf{H}_{k+1}^c$. It is easy to verify that there is no redundant constraint in $\mathcal{H}^+$. Finally, since we have the inclusion $\mathbf{H}_{k-1/2}^c \cap \mathbf{H}_k \subset \mathbf{H}_{k-1}^c$ from (9.20)

$$\begin{aligned}\mathbf{H}_{k-1/2}^c \cap \mathbf{H}_k \cap \mathbf{H}_{k+1}^c &= (\mathbf{H}_{k-1}^c \cap \mathbf{H}_{k-1/2}^c) \cap (\mathbf{H}_k \cap \mathbf{H}_{k+1}^c) \\ &= \mathbf{D}_{k-1/2} \cap \mathbf{L}_{k+1/2}\end{aligned}$$

hence $\mathcal{H}^+$ is the minimal representation of $\cap_{i=0}^{k} \mathbf{P}_{i+1/2}$.

If $m_k = 7$, the analysis is similar to case $m_k = 6$.

$$\begin{aligned}\mathbf{H}^c_{k-1} \cap \mathbf{H}^c_k \cap \mathbf{H}^c_{k+1/2} &= (\mathbf{H}^c_{k-1} \cap \mathbf{H}^c_{k-1/2}) \cap (\mathbf{H}^c_k \cap \mathbf{H}^c_{k+1/2}) \\ &= \mathbf{D}_{k-1/2} \cap \mathbf{D}_{k+1/2}\end{aligned}$$

hence $\mathcal{H}^+$ is the minimal representation of $\cap_{i=0}^{k} \mathbf{P}_{i+1/2}$. This finishes the proof. □

Adjacent polyhedra

Definition 9.5 (Adjacent polyhedra)**.** *Two polyhedra $\boldsymbol{P}$ and $\boldsymbol{P}'$ in a polyhedral partition of the space are said to be k-adjacent if they have a face of dimension k in common,* i.e. *there exists a supportive hyperplane $\partial\boldsymbol{H}$ for both $\boldsymbol{P}$ and $\boldsymbol{P}'$ and the intersection $\boldsymbol{P} \cap \boldsymbol{P}' \cap \partial\boldsymbol{H}$ is of dimension k. Then $\boldsymbol{P}$ and $\boldsymbol{P}'$ are said to be $\partial\boldsymbol{H}$-adjacent.*

For an accepted mode vector $\boldsymbol{m}$ and its associated polyhedron $\text{Dom}(\boldsymbol{m})$, it is of interest to find the polyhedra of the partition adjacent to it. Algorithm 9.5 returns all the polyhedra of the partition (n+1)-adjacent to $\text{Dom}(\boldsymbol{m})$. First, the mode string $s(0)\cdots s(n)$ and the minimal representation of $\text{Dom}(\boldsymbol{m})$ are computed with Algorithms 9.2 and 9.4. Then for all $\mathbf{H} \in \mathcal{H}$, the algorithm computes the mode string of the polyhedron of the partition $\partial\mathbf{H}$-adjacent to $\text{Dom}(\boldsymbol{m})$, and finds the associated mode vector $\boldsymbol{m}_\mathbf{H}$ with Algorithm 9.3 (see Figure 9.6 for an illustration of the algorithm).

Algorithm 9.5 Find all the polyhedra adjacent to $\text{Dom}(\boldsymbol{m})$: **adj($\boldsymbol{m}$)**

Require: accepted mode vector $\boldsymbol{m}$
1. $s(0)\cdots s(n) = \mathbf{m2s}(\boldsymbol{m})$
2. $\mathcal{H} = \mathbf{minRep}(\boldsymbol{m})$
3. **for** $\mathbf{H} \in \mathcal{H}$:
4. $\quad s'(0)\cdots s'(n) = s(0)\cdots s(n)$
5. $\quad$ **for** $i \in \{0, \cdots, n\}$:
6. $\quad\quad$ **if** $\mathbf{H} = \mathbf{H}_i$ **then** $s'(i) = D$
7. $\quad\quad$ **if** $\mathbf{H} = \mathbf{H}^c_i$ **then** $s'(i) = W$
8. $\quad\quad$ **if** $\mathbf{H} = \mathbf{H}_{i+1}$ **then** $s'(i) = L$
9. $\quad\quad$ **if** $\mathbf{H} = \mathbf{H}^c_{i+1}$ **then** $s'(i) = W$
10. $\quad\quad$ **if** $\mathbf{H} = \mathbf{H}_{i+1/2}$ **then** $s'(i) = D$
11. $\quad\quad$ **if** $\mathbf{H} = \mathbf{H}^c_{i+1/2}$ **then** $s'(i) = L$
12. $\quad \boldsymbol{m}_\mathbf{H} = \mathbf{s2m}(s'(0)\cdots s'(n))$
13. **return** adjacent polyhedra $\{\boldsymbol{m}_\mathbf{H}\}_{\mathbf{H}\in\mathcal{H}}$

Definition 9.6. *Two accepted mode vectors $\boldsymbol{m}$ and $\boldsymbol{m}'$ are adjacent if the closures of their respective domain* $\text{Dom}(\boldsymbol{m})$ *and* $\text{Dom}(\boldsymbol{m}')$ *are (n+1)-adjacent.*

Proposition 9.6. *For every accepted mode vector $\boldsymbol{m}$, Algorithm 9.5 returns all the accepted mode vectors adjacent to $\boldsymbol{m}$. (Formal proof given in the appendix.)*

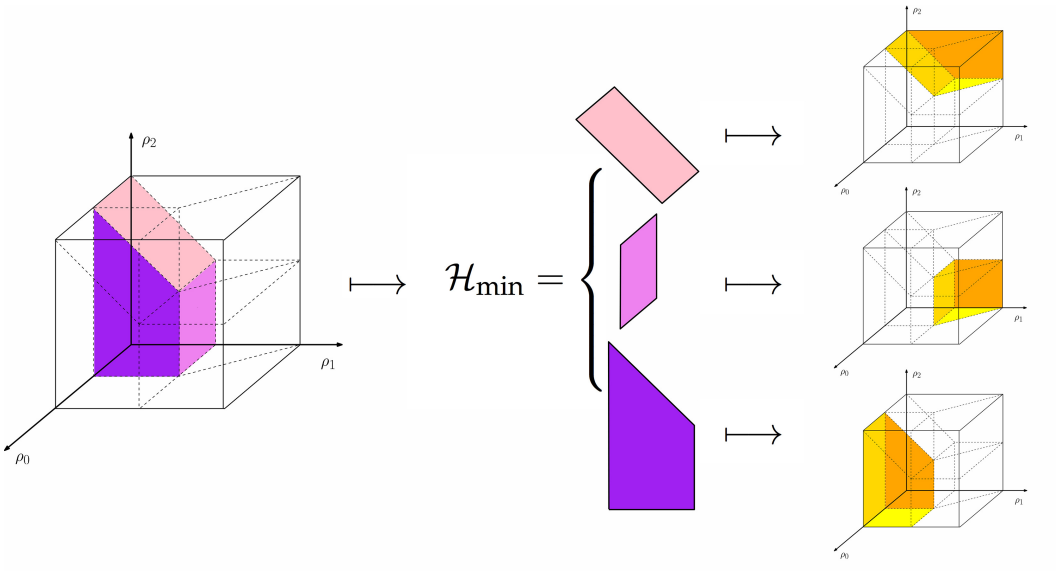

Figure 9.6: We want all the polyhedra of the partition adjacent to a fixed polyhedron. First, we find all the K facet-defining hyperplanes of purple, *i.e.* minimal representation. Then for each facet, we find the only polyhedron that shares this facet with purple. Hence K is also the number of polyhedra of the partition adjacent to purple.

Since Algorithm 9.4 adds at most 2 constraints per iteration, minRep($\boldsymbol{m}$) has at most $2(n+1)$ constraints, hence at most $2(n+1)$ accepted mode vectors are adjacent to $\boldsymbol{m}$.

9.4 Hybrid estimation algorithms

Kalman filtering algorithm for each mode vector

In discrete time and space, the dynamics of the traffic flow along a homogeneous section of highway is well described by the Godunov scheme applied to the LWR equation with triangular flux function (see Prop. 9.1). The small uncertainties on the parameters of the model $A_{\boldsymbol{m}}$ and $b_{\boldsymbol{m}}$ can be reasonably covered by a zero-mean Gaussian noise $\boldsymbol{\eta}^t \sim \mathcal{N}(0, Q^t)$ with covariance Q^t. The discrete dynamics in mode vector $\boldsymbol{m}$ become:

$$\boldsymbol{\rho}^{t+1} = A_{\boldsymbol{m}}\boldsymbol{\rho}^t + b_{\boldsymbol{m}} + c^t + \boldsymbol{\eta}^t \tag{9.21}$$

The mode vector $\boldsymbol{m}$ is no longer fixed by $\boldsymbol{\rho}^t$, but a probability distribution over all accepted mode vectors is maintained to take into account the uncertainty in mode estimation; that is, at each time step t, the model is in several different mode vectors with positive probabilities. We add an *observation model*:

$$\boldsymbol{z}^t = H^t\boldsymbol{\rho}^t + \boldsymbol{\chi}^t \tag{9.22}$$

where $\boldsymbol{\chi}^t \sim \mathcal{N}(0, R^t)$ is the zero-mean observation noise with covariance matrix R^t, and H^t is the $d_t \times (n+2)$-dimensional linear observation matrix which encodes the d_t observations (each one of them being at a discrete cell on the discretization domain) for which the density is observed during discrete time step t, and n is the dimensionality of the system. In the traffic case, sensing devices (such as loop detectors) are placed at several locations along a section of highway, and their positions are encoded in the matrix H^t. For example, in the discrete case for $n = 3$, if one sensor is in cell 1 and another in cell 3, then both sensors provide observations $z_1^t = \rho_1^t + \chi_1^t$ and $z_2^t = \rho_3^t + \chi_2^t$, which is in matrix form:

$$\begin{pmatrix} z_1^t \\ z_2^t \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \boldsymbol{\rho}^t + \begin{pmatrix} \chi_1^t \\ \chi_2^t \end{pmatrix} \tag{9.23}$$

where the state is $\boldsymbol{\rho}^t = (\rho_0^t, \rho_1^t, \cdots, \rho_5^t)^T$. In this small example, the observation matrix is $H^t = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$ and the number of observations is $d_t = 2$.

In the rest of the section, we use the standard notations $\boldsymbol{m}_j$ for the different mode vectors, and subscript j denotes quantities that are pertaining to mode $\boldsymbol{m}_j$. Note that $\boldsymbol{m}_j$ refers to the whole *mode vector* $\boldsymbol{m}$ and not the entries of $\boldsymbol{m}$.

Let $\hat{\boldsymbol{\rho}}^{t:t}$ and $P^{t:t}$ be the *a posteriori* state estimate and error covariance matrix at time t. The *predicted* state estimate $\hat{\boldsymbol{\rho}}_j^{t+1:t}$ and covariance estimate $P_j^{t+1:t}$ of the *prediction step* in *mode* $\boldsymbol{m}_j$ are:

$$\begin{aligned} \textbf{Prediction:} \quad & \hat{\boldsymbol{\rho}}_j^{t:t+1} = A_j \hat{\boldsymbol{\rho}}^{t:t} + b_j + c^t \\ & P_j^{t:t+1} = A_j P^{t:t} (A_j)^T + Q^t \end{aligned} \tag{9.24}$$

The *measurement residual* $\boldsymbol{r}_j^{t+1}$, *residual covariance* S_j^{t+1}, *Kalman gain* K_j^{t+1}, *updated state estimate* $\hat{\boldsymbol{\rho}}_j^{t+1:t+1}$, and *updated estimate covariance* $P_j^{t+1:t+1}$ of the *update step* in *mode* j are:

$$\begin{aligned} \textbf{Residuals:} \quad & \boldsymbol{r}_j^{t+1} = \boldsymbol{z}^{t+1} - H^{t+1} \hat{\boldsymbol{\rho}}_j^{t:t+1} \\ & S_j^{t+1} = H^{t+1} P_j^{t:t+1} (H^{t+1})^T + R^{t+1} \\ \textbf{Kalman gain:} \quad & K_j^{t+1} = P_j^{t:t+1} (H^{t+1})^T (S_j^{t+1})^{-1} \\ \textbf{Updates:} \quad & \hat{\boldsymbol{\rho}}_j^{t+1:t+1} = \hat{\boldsymbol{\rho}}_j^{t:t+1} + K_j^{t+1} \boldsymbol{r}_j^{t+1} \\ & P_j^{t+1:t+1} = (I - K_j^{t+1} H^{t+1}) P_j^{t+1:t} \end{aligned} \tag{9.25}$$

In [107], a measure of the likelihood of the Kalman filter in mode j is given by the *mode likelihood function* Λ_j^{t+1}, where $\mathcal{N}(x; a, b)$ is the probability density function of the normal distribution with mean a and variance b:

$$\Lambda_j^{t+1} = \mathcal{N}(\boldsymbol{r}_j^{t+1};\ 0,\ S_j^{t+1}) \tag{9.26}$$

The noise might result in densities outside bounds. We project onto $[0, \rho_{\text{jam}}]^{n+2}$, *i.e.* the equation is implicitly $\hat{\boldsymbol{\rho}}_j^{t+1:t+1} = \Pi(\hat{\boldsymbol{\rho}}_j^{t:t+1} + K_j^{t+1} \boldsymbol{r}_j^{t+1})$ where $\Pi(\cdot)$ is the projection operator. This is a legitimate because densities cannot be negative nor exceed a maximum value ρ_{jam}.

Interactive multiple model KF

Let us denote by $\{m(t) = \boldsymbol{m}_j\}$ the event that the system is in the mode $\boldsymbol{m}_j$ at time t. We then assume that the model is a discrete-time stochastic linear hybrid system in which the mode evolution is governed by the finite state Markov chain

$$\mu^{t+1} = \Pi \mu^t \tag{9.27}$$

where $\pi_{ij} = P(m(t+1) = \boldsymbol{m}_j \,|\, m(t) = \boldsymbol{m}_i)$ for all $\boldsymbol{m}_i, \boldsymbol{m}_j \in \mathcal{M}$ is the mode transition matrix, $\mu_j^t = P(m(t) = \boldsymbol{m}_j)$ for all $m_j \in \mathcal{M}$ is the mode probability at time t; and the set of accepted modes is $\mathcal{M}$.

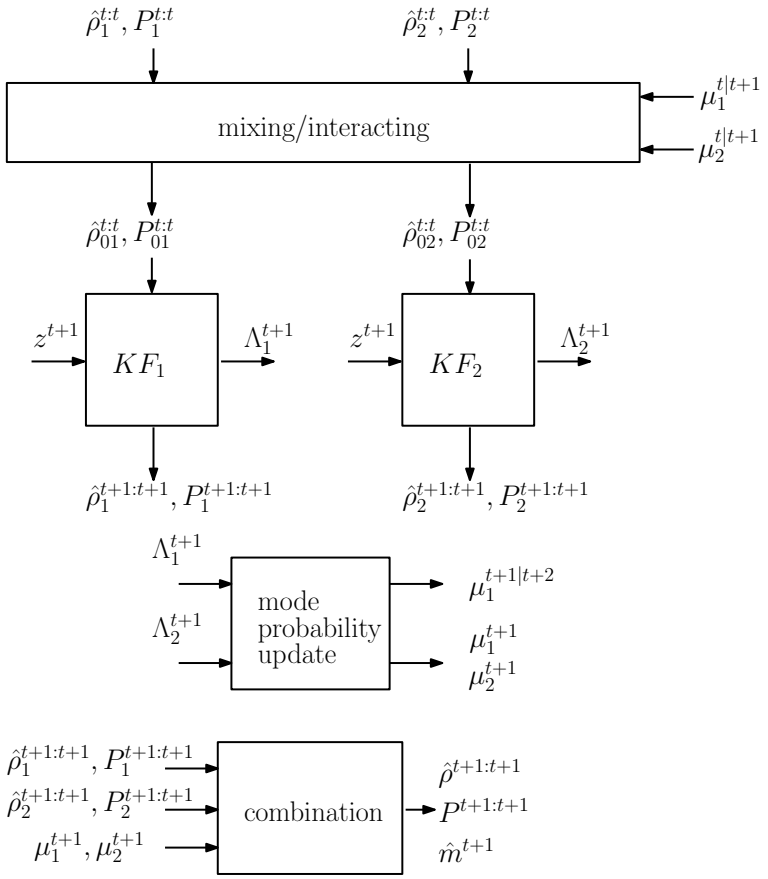

Figure 9.7: Illustration of the structure of IMM algorithm for a two-mode system from [107].

Effective estimation techniques for stochastic hybrid systems are based in multiple models since it is natural to apply a statistical filter for each of the modes. The *Interactive Multiple Model* (IMM) algorithm [9, 24, 108] is a cost-effective (in terms of performance versus complexity) estimation scheme in which there is a *mixing/interacting* step at the beginning of the estimation process, which computes new initial conditions for the Kalman filters matched to the individual modes at each time step as illustrated in Figure 9.7.

We consider the IMM algorithm in which $\mathcal{M}^t$ is the set of modes for which the Kalman filter is applied at time step t. The set $\mathcal{M}^t$ is the set of modes $\boldsymbol{m}_j$ with positive mode probabilities $\mathcal{M}^t = \{\boldsymbol{m}_j \,|\, \mu_j^t > 0\}$. In the standard IMM, a filter is applied to every mode. The components of the *mixing* step are the *mixing probability* $\mu_{ij}^{t|t+1}$ of being in mode i at time t given that the mode at time $t+1$ is j, the *mixed condition* $\hat{\boldsymbol{\rho}}_{0j}^{t:t}$ and $P_{0j}^{t:t}$ for the state estimate and covariance of mode j at time t, and the "spread-of-the-means" X_j in the expression of $P_{0j}^{t:t}$. They are computed for $j \in \mathcal{M}^{t+1}$ w.r.t. $\hat{\boldsymbol{\rho}}_i^{t:t}$ and $P_i^{t:t}$, the state estimate and its covariance of Kalman filter i at time t:

$$\begin{array}{rl} \mu_{ij}^{t|t+1} & = \frac{1}{Z_j}\pi_{ij}\mu_i^t \text{ for } i \in \mathcal{M}^t \quad \text{with} \quad Z_j = \sum_{i\in\mathcal{M}^t} \pi_{ij}\mu_i^t \\ \hat{\boldsymbol{\rho}}_{0j}^{t:t} & = \sum_{i\in\mathcal{M}^t} \hat{\boldsymbol{\rho}}_i^{t:t}\mu_{ij}^{t|t+1} \\ P_{0j}^{t:t} & = \sum_{i\in\mathcal{M}^t} P_i^{t:t}\mu_{ij}^{t|t+1} + X_j \\ X_j & := \sum_{i\in\mathcal{M}^t} (\hat{\boldsymbol{\rho}}_i^{t:t} - \hat{\boldsymbol{\rho}}_{0j}^{t:t})(\hat{\boldsymbol{\rho}}_i^{t:t} - \hat{\boldsymbol{\rho}}_{0j}^{t:t})^T \mu_{ij}^{t|t+1} \end{array} \tag{9.28}$$

We apply the Kalman filter in each mode $j \in \mathcal{M}^{t+1}$ (KF_j) as described with equations (9.24,9.25) and the resulting mode likelihood functions Λ_j^{t+1} are obtained from $\hat{\boldsymbol{\rho}}_j^{t+1:t+1}$ and $P_j^{t+1:t+1}$ with equation (9.26). The mode probability $\mu^t = \{\mu_j^t\}$ is then updated through:

$$\mu_j^{t+1} = \frac{1}{Z}\Lambda_j^{t+1} \sum_{i\in\mathcal{M}^t} \pi_{ij}\mu_i^t \text{ for } j \in \mathcal{M}^{t+1} \tag{9.29}$$

where Z is a normalization constant and Λ_j^{t+1} is the mode likelihood function defined in (9.26). The output of the IMM algorithm are the state estimate $\hat{\boldsymbol{\rho}}^{t+1:t+1}$ which is a weighted sum of the estimates from the Kalman filters in each mode and its covariance $P^{t+1:t+1}$, and the mode estimate $\hat{\boldsymbol{m}}^{t+1}$ is the mode which has the highest mode probability. They are given by the *combination* step:

$$\begin{array}{l} \hat{\boldsymbol{\rho}}^{t+1:t+1} = \sum_{j\in\mathcal{M}^{t+1}} \hat{\boldsymbol{\rho}}_j^{t+1:t+1}\mu_j^{t+1} \\ P^{t+1:t+1} = \sum_{j\in\mathcal{M}^{t+1}} P_j^{t+1:t+1}\mu_j^{t+1} + X \\ X := \sum_{j\in\mathcal{M}^{t+1}} (\hat{\boldsymbol{\rho}}_j^{t+1:t+1} - \hat{\boldsymbol{\rho}}^{t+1:t+1})(\hat{\boldsymbol{\rho}}_j^{t+1:t+1} - \hat{\boldsymbol{\rho}}^{t+1:t+1})^T \mu_j^{t+1} \\ \hat{\boldsymbol{m}}^{t+1} := \mathrm{argmax}_{j\in\mathcal{M}^{t+1}}\, \mu_j^{t+1} \end{array} \tag{9.30}$$

In [107, 83], the IMM algorithm is used as a hybrid estimator for Air Traffic Control (ATC) tracking. The models used include one for the uniform motion and one (or more) for the maneuver. However, the discretized PDE model described in Section 9.2 has an exponential number of modes, which induces an exponential time complexity of the IMM.

Extended Kalman filter

In the simplest case, we assume that the only possible mode at the next time is the mode $\boldsymbol{m}_j$ of the estimate, *i.e.* $\mathcal{M}^{t+1} = \{\boldsymbol{m}_j\}$ and $\mu_j^{t+1} = 1$ with $\hat{\boldsymbol{\rho}}^{t:t} \in \mathrm{Dom}(\boldsymbol{m}_j)$. We apply the Kalman filter only to this mode. With $\mathcal{M}^t = \{\boldsymbol{m}_i\}$, equations (9.28) become:

$$\hat{\boldsymbol{\rho}}_{0j}^{t:t} = \hat{\boldsymbol{\rho}}_i^{t:t}, \quad P_{0j}^{t:t} = P_i \tag{9.31}$$

We apply the Kalman filter only to mode $\boldsymbol{m}_j$ to obtain $\hat{\boldsymbol{\rho}}_j^{t+1:t+1}$ and $P_j^{t+1:t+1}$. Finally, the outputs of the *combination step* given by equations (9.30) are simply $\hat{\boldsymbol{\rho}}^{t+1:t+1} = \hat{\boldsymbol{\rho}}_j^{t+1:t+1}$, $P^{t+1:t+1} = P_j^{t+1:t+1}$, and $\hat{\boldsymbol{m}}^{t+1} = \boldsymbol{m}_j$.

In this model, the IMM algorithm is exactly an *Extended Kalman filter* (EKF) applied to our discretized system presented in Proposition 9.1. The linear model in mode $\boldsymbol{m}$ such that $\hat{\boldsymbol{\rho}}^{t:t} \in \mathrm{Dom}(\boldsymbol{m})$ coincides exactly with the linearization of the discrete dynamics around $\hat{\boldsymbol{\rho}}^{t:t}$.

Despite the exponential number of modes, we can compute the *predicted state estimate* $\hat{\boldsymbol{\rho}}_j^{t+1:t}$ and the *predicted covariance estimate* $P_j^{t+1:t}$ in mode $\boldsymbol{m}_j$ (see (9.24)) in linear time and quadratic time respectively, without generating any dense matrix because A_j is completely defined by mode vector $\boldsymbol{m}_j$ and A_j is tridiagonal (see Algorithm 9.4). Hence the time complexity of the prediction step is $O(n^2)$, with constant space complexity. With d the number of observations (or number of sensors), the time complexity of the *update step* of the Kalman filter given by (9.25) is $O(dn^2 + d^3 + nd^2)$, and so as the two steps combined of the KF.

In comparison, the *Ensemble Kalman Filter* (EnKF) is a popular estimation algorithm for non-linear dynamical systems. It is commonly used in the traffic monitoring community [170]. The EnKF is based on a *Monte Carlo approximation* of the Kalman filter which approximates the covariance matrix of the state vector with the sample covariance of the ensemble. The prediction step consists in applying the system's dynamics to each sample, which has complexity $O(Nn^2)$, where N is the number of samples (ensemble members). Mandel's report [113] shows that the computational complexity of the update step of the EnKF algorithm is $O(d^3 + d^2N + dN^2 + nN^2)$. So the total complexity of the EnKF is $O(d^3 + d^2N + dN^2 + nN^2 + Nn^2)$.

Algorithm 9.4 describes the EKF. The parameters $\mathbf{L}(1), \cdots, \mathbf{L}(7) \in \mathbb{R}^3$, $w(1), \cdots, w(7) \in \mathbb{R}$, given in Table 9.1 describe the linear modes of our hybrid system.

Algorithm 9.6 (Explicit) Extended Kalman filter

Require: initial state $\boldsymbol{\rho}_0 \in [0, \rho_{\text{jam}}]^{n+2}$, boundary conditions $(u(t),\, d(t))_{t\geq 0}$, state covariance $\{Q^t\}_{t\geq 0}$, observations $\{\boldsymbol{z}^t\}_{t\geq 0}$, observation matrix $\{H^t\}_{t\geq 0}$, observation covariance $\{R^t\}_{t\geq 0}$

1. **for** $t \in \{0,\, 1,\, 2,\, \cdots\}$:
2. $\quad \boldsymbol{m} = \mathbf{rho2m}(\hat{\boldsymbol{\rho}}^t)$ \\\ mode estimate, see algorithm 9.1
3. $\quad (\hat{\boldsymbol{\rho}}^{t+1},\, P^{t+1}) = \mathbf{KF}(\boldsymbol{m}, \hat{\boldsymbol{\rho}}^t, P^t, \cdots)$ \\\KF, see algorithm 9.8
4. **return** $(\hat{\boldsymbol{\rho}}^t,\, P^t)_{t\geq 0}$

Extended Kalman filter: numerical results

In traffic estimation, the density measurements along the highway are usually sparse. For example, in the 18-mile stretch of I-880 Northbound in the Bay Area, CA (see Figure 9.9.a), the Mobile Millennium traffic monitoring system receives measurements from 29 loop detectors (PeMS) every 30s on March 5th, 2012 between 7am and 8am. This section of highway is

Algorithm 9.7 Prediction step of Kalman filter in mode $\boldsymbol{m}$: **predict**$(\boldsymbol{m}, \boldsymbol{\rho}, P, u^+, d^+, Q)$

Require: mode vector $\boldsymbol{m} = [m_1, \cdots, m_n] \in \{1, \cdots, 7\}^n$, current state $\boldsymbol{\rho} = [\rho_0, \cdots, \rho_{n+1}] \in [0, \rho_{\text{jam}}]^{n+2}$, current state estimate covariance P, next boundary conditions u^+, $d^+ \in \mathbb{R}$, current state noise covariance Q.

1. $\rho_0^+ = u^+$
2. $\rho_{n+1}^+ = d^+$
3. **for** $i \in \{1, \cdots, n\}$:
4. $\quad \rho_i^+ = \mathbf{L}(m_i) \times [\rho_{i-1},\, \rho_i,\, \rho_{i+1}]^T + w(m_i)$
5. $M := \text{zeros}(n+2,\, n+2)$ \\ create temporary matrix M
6. **for** $(i, j) \in \{1, \cdots, n\}^2$:
7. $\quad M_{ij} = \mathbf{L}(m_i) \times [P_{i-1,j},\, P_{i,j},\, P_{i+1,j}]^T$ \\ do $A \times P$
8. **for** $(i, j) \in \{1, \cdots, n\}^2$:
9. $\quad P_{ij}^+ = [M_{i,j-1},\, M_{i,j},\, M_{i,j+1}] \times \mathbf{L}(m_j)^T$ \\do $(AP)A^T$
10. $P^+ = P^+ + Q$ \\predict state covariance
11. **return** $\boldsymbol{\rho}^+$, P^+

Algorithm 9.8 Kalman filter in mode $\boldsymbol{m}$: **KF**$(\boldsymbol{m}, \hat{\boldsymbol{\rho}}^t, P^t, u(t+1), d(t+1), Q^t, \boldsymbol{z}^{t+1}, H^{t+1}, R^{t+1})$

Require: mode vector $\boldsymbol{m}$, current state $\hat{\boldsymbol{\rho}}^t$, current state estimate covariance P^t, next boundary conditions $u(t+1)$, $d(t+1)$, current state noise covariance Q^t, next measurement $\boldsymbol{z}^{t+1}$, next observation matrix H^{t+1}, next observation covariance R^{t+1}.

1. $(\hat{\boldsymbol{\rho}}^{t:t+1}, P^{t:t+1}) =$ **predict**$(\hat{\boldsymbol{\rho}}^t, P^t, \{\cdots\})$ \\see algorithm 9.4
2. $(\hat{\boldsymbol{\rho}}^{t+1}, P^{t+1}, \Lambda^{t+1}) =$ **update**$(\hat{\boldsymbol{\rho}}^{t:t+1}, \{\cdots\})$ \\see (9.25)
3. **return** $\hat{\boldsymbol{\rho}}^{t+1}$, P^{t+1}, Λ^{t+1}

discretized into cells of length 198m, hence $n = 148$ and $m = 29$, and the EnKF with 100 ensembles is currently used for traffic estimation, so $N = 100$ and $m \leq \min(n, N)$. Hence the time complexities of the KF (or EKF) and EnKF are $O(mn^2)$ and $O(n^2N + nN^2)$ respectively. With N large (>50), the complexity analysis predicts that the EKF should be faster than the EnKF.

The running times of the implementation of both the EKF and the EnKF estimators on an Intel® Core™ i5 480M 2.67GHz are shown in Figure 9.8.a), for increasing portions of the I-880 starting from East Industrial in Fremont, CA. For example, 60 cells (∼7.5miles) span from East Industrial to Dumbarton Bridge, and 113 cells (∼14miles) reaches San Mateo Bridge. The EKF is significantly faster than the EnKF with 100 samples, which is implemented in the Mobile Millennium. This confirms our complexity analysis of both algorithms.

Figure 9.9.c,d) shows the contour plots of the output of the EnKF and the EKF estimators, which consists in the density in the time-space domain. The regions with high densities are represented in red and the regions with low densities in blue. Both estimators give very similar higher resolution scalar fields of the density (1440 time steps by 141 cells) by assimilating sparse density measurements (240 time steps by 29 PeMS stations, see Figure

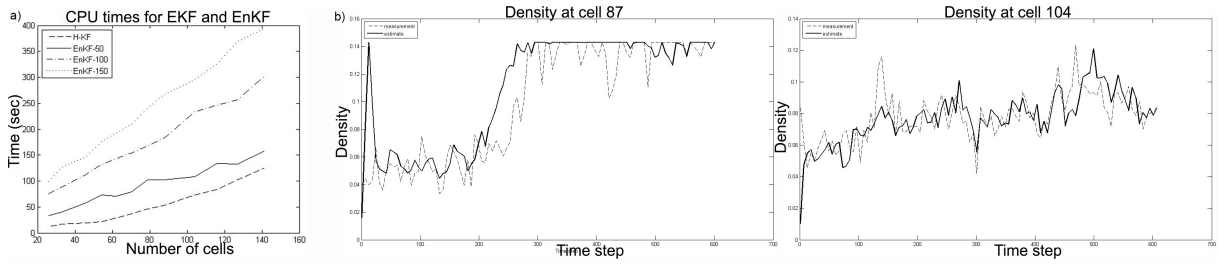

Figure 9.8: a) Computational time for an increasing section of the I-880 (measured in the number of cells) for the EKF (dashed line), the EnKF with 50 ensembles (continuous line), the EnKF with 100 ensembles (dashed-dotted line), and the EnKF with 150 ensembles (dotted line). b) Comparison between the density measurements (dashed line) and estimates (bold line) at cell 87 and cell 104.

9.9.b). Moreover, by removing measurements at an arbitrary cell, Figure 9.8.b) shows that the estimation algorithm performs well since the density estimate is close to the actual measurement.

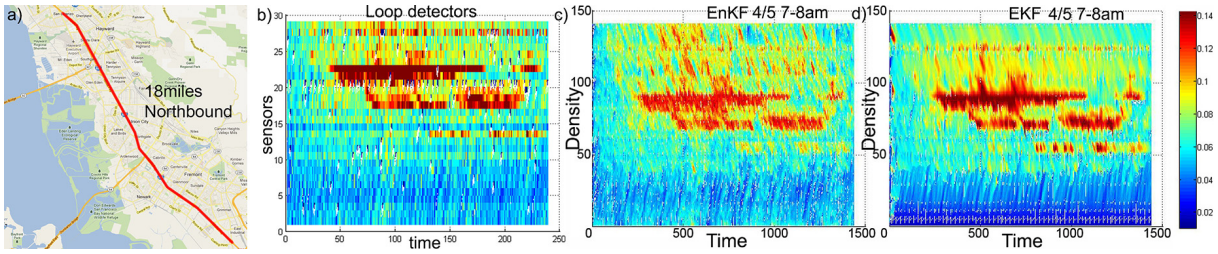

Figure 9.9: a) Experimental data location: 18-mile long stretch of I-880 in the Bay Area on the *Mobile Century* site. b) Contour plot of the density from the 29 PeMS stations every 30s on March 5th, 7-8am. Each vertical line in the contour plot reports the measurements from the 29 sensors along the highway at a specific time. c) Output of the EnKF d) Output of the EKF. The time step is on the X-axis and the number of cells is on the Y-axis. Each vertical line of the diagram is a snapshot of the state estimate of the highway at a specific time.

In summary, the explicit representation as a switched hybrid system gives a powerful framework for tracking the mode evolution and preforming hybrid estimation. For instance, the EKF can be implemented easily by applying the KF in the mode vector of the state estimate. However, straight application of the IMM algorithm [107] is not tractable because the complexity is $O(\tau_n(2.247)^n)$ where τ_n is the complexity of the KF and $(2.247)^n$ is the asymptotic number of modes.

9.5 Reduced IMM

Reduction to adjacent modes

We presented an algorithm to construct the minimal representation of $\mathrm{Dom}(\boldsymbol{m})$, which enables to find the adjacent modes. Moreover, two adjacent modes only differ by at most two entries. Hence, when the discretized model is in quasi-steady state, and n is relatively small, only one cell switches mode at the next time step, so the state is most likely to jump to an adjacent mode vector. This suggests to consider only the mode of the state estimate and its adjacent modes. Hence, the number of modes considered is less than $2(n+1)$.

We can further reduce the number of modes by taking into account the state covariance P and the distance between the state estimate and the facets of the polyhedron. Let $\mathcal{H}$ be the minimal representation of the mode vector $\hat{\boldsymbol{m}}$ of the state estimate (*i.e.* $\hat{\boldsymbol{\rho}} \in \mathrm{Dom}(\hat{\boldsymbol{m}})$), and let $\mathbf{H} \in \mathcal{H}$ with equation $\mathbf{H} = \{\boldsymbol{\rho} \,|\, \mathbf{a} \cdot \boldsymbol{\rho} - b \leq 0\}$ and $\|a\|_2 = 1$. Then the distance from the supportive hyperplane $\partial\mathbf{H}$ is: $d(\hat{\boldsymbol{\rho}}, \partial\mathbf{H}) = \min \|\hat{\boldsymbol{\rho}} - \partial\mathbf{H}\|_2 = |b - \mathbf{a} \cdot \hat{\boldsymbol{\rho}}|$.

The probability distribution of the state along the normal $\mathbf{a}$ to $\partial\mathbf{H}$ is $Ke^{-\frac{(\mathbf{a}\cdot(\boldsymbol{\rho}-\hat{\boldsymbol{\rho}}))^2}{2\mathbf{a}^T P \mathbf{a}}}$, so the probability that the state is inside of half-space $\mathbf{H}$ along the normal $\mathbf{a}$ is

$$K \int_{-\infty}^{|b-\mathbf{a}\cdot\hat{\boldsymbol{\rho}}|} e^{-\frac{t^2}{2\mathbf{a}^T P \mathbf{a}}} dt = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{|b - \mathbf{a}\cdot\hat{\boldsymbol{\rho}}|}{\sqrt{2\mathbf{a}^T P \mathbf{a}}}\right)\right)$$

where *erf* is the *error function*. Since *erf* is an increasing function, we keep only the $\partial\mathbf{H}$-adjacent modes for which the following quantity is small (see Algorithm 9.9)

$$r(\hat{\boldsymbol{\rho}}, \mathbf{H}) = |b - \mathbf{a}\cdot\hat{\boldsymbol{\rho}}| / \sqrt{2\mathbf{a}^T P \mathbf{a}}, \quad \mathbf{H} \in \mathcal{H} \tag{9.32}$$

Algorithm 9.9 Find all adjacent polyhedra close to $\hat{\boldsymbol{\rho}}$: **adj2**$(\boldsymbol{m}, \hat{\boldsymbol{\rho}}, P, \beta)$

Require: mode estimate $\hat{\boldsymbol{m}}$, state estimate $\hat{\boldsymbol{\rho}}$, state estimate covariance P, tolerance β
1. $s(0)\cdots s(n) =$ **m2s**$(\hat{\boldsymbol{m}})$
2. $\mathcal{H} =$ **minRep**$(\hat{\boldsymbol{m}})$
3. **for** $\mathbf{H} \in \mathcal{H}$:
4. **if** $\mathbf{H} = \mathbf{H}_i$ **then** $r = |\rho_i - \rho_c| / \sqrt{2P_{ii}}$
5. **if** $\mathbf{H} = \mathbf{H}_{i+1/2}$ **then** $r = |\rho_{i+1} + \frac{v_f}{w_f}\rho_i - \rho_{\mathrm{jam}}| / \sqrt{2(\frac{v_f}{w_f})^2 P_{ii} + 4\frac{v_f}{w_f}P_{i,i+1} + 2P_{i+1,i+1}}$
6. **if** $r > \beta$ **then** remove $\mathbf{H}$ from $\mathcal{H}$
7. execute lines 3 to 14 of Algorithm 9.5
8. **return** adjacent polyhedra close to state estimate $\{\boldsymbol{m}_{\mathbf{H}}\}_{\mathbf{H}\in\mathcal{H}}$

This is a refinement of the EKF. Instead of relying on one possible mode, we consider a set of possible adjacent modes at time t and apply the KF to each one of them. However, the adjacent modes differ by only one or two entries, so they only represent a restricted set of close possibilities centered around the mode estimate. Hence, the reduced IMM based on adjacent modes is still very similar to the EKF.

Representative mode vectors with clustering algorithm

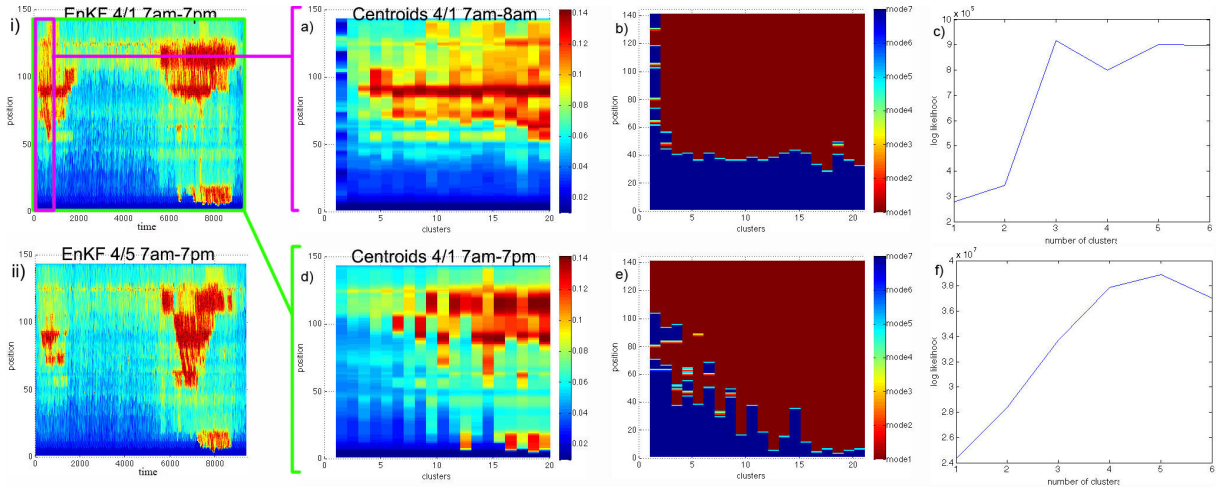

Figure 9.10: i) Traffic density estimate on March 1st, 2012 from 7am to 7pm. ii) Traffic density estimate on March 5th, 2012 from 7am to 7pm. a,b,c) 20 clusters of the density space using *k-means* on March 1st, 2012 from 7 to 8am, the corresponding modes, and the log likelihood. d,e,f) 20 clusters of the density space using *k-means* on March 1st, 2012 from 7am to 7pm.

An intuitive method consists in using a clustering algorithm to reduce the space of modes to a representative set $\mathcal{M}^K$. Historical data of traffic density estimate on March 1st, 2012 (see Figure 9.10.i)) provides $T = 9355$ observations or samples of the state vector, where T is the number of time steps in the observed data. We partition these T samples into K clusters using the popular *k-means algorithm*. The centroid of each cluster, which may not necessarily be a member of the data set, are density vectors that represent particular states of the highway which are representative of its evolution. They are shown in Figures 9.10.a,d). We have the index of the cluster on the X-axis and the position along the highway on the Y-axis. For instance, the first cluster represents a density vector of the highway mostly in free flow whereas the last cluster represents the density vector of the highway mostly in congestion in the top part.

Then, we derive the modes of these K centroids, and we assume that our system can only be in these K modes. They are illustrated in Figure 9.10.e). We have the index of the modes on the X-axis, and the position on the highway along the Y-axis. Each column represents a modal regime of the highway. For example, in the first mode vector (in the first column), the cells are in mode 7 in the upstream part, and the cells in the downstream are in mode 1. When ρ_{i-1}, ρ_i, $\rho_{i+1} > \rho_c$ for a particular cell i, the Godunov flux (9.7) is in congestion regime at both interfaces $i-1|i$ and $i|i+1$ and cell i is in mode 1 (see Table 9.1). Conversely, $m_i = 7$ when the Godunov flux is in free flow regime at both interfaces: $\rho_{i-1}, \rho_i, \rho_{i+1} < \rho_c$. Hence the regions in which $m_i = 1$ (resp. $m_i = 7$), colored in red (resp. blue), represent cells

that are in congestion (resp. free flow) regime. The cells are in the other modes $\{2, 3, \cdots, 6\}$ correspond to a transition regime between free flow and congestion. We apply the IMM with this reduced set of modes to estimate the traffic on March 5th. This is a valid approach since the traffic conditions are similar during weekdays (see Figure 9.10.i,ii)).

Algorithm 9.10 Clustering historical data: **cluster**$(\{\boldsymbol{\rho}^t\}_{t\in\{1,\cdots,T\}})$

Require observed data set $\{\boldsymbol{\rho}^t\}_{t\in\{1,\cdots,T\}}$
1. partition $\{\boldsymbol{\rho}^t\}_{t\in\{1,\cdots,T\}}$ into K clusters and get centroids $\{\bar{\boldsymbol{\rho}}^k\}_{k\in\{1,\cdots,K\}}$
2. **for** $k \in \{1, \cdots, K\}$ **do** $\bar{\boldsymbol{m}}^k = \mathbf{rho2m}(\bar{\boldsymbol{\rho}}^k)$; **end for**
3. **return** set of K representative modes $\mathcal{M}^K = \{\bar{\boldsymbol{m}}^k\}_{k\in\{1,\cdots,K\}}$

To determine the optimal number of clusters, we have applied the above procedure to one hour of observed data, on March 1st from 7am to 8am. The density centroids and their mode are shown in Figure 9.10.a,b). Then we applied the IMM algorithm on March 5th from 7am to 8am and compared it against the state estimate given by the EnKF for different numbers of clusters. We have calculated the log-likelihood which is a measure of the performance of the estimation scheme. We see that the optimal number of clusters is 3, because adding more clusters won't increase the performance of the estimation algorithm (see Figure 9.10.c)). We have also applied the procedure to 12 hours of observed data. In this case, the optimal number of clusters increases to 5. This is expected because we have a greater variety of regimes in 12 hours. This proves the efficiency of the IMM algorithm applied with this representative modes, because the complexity is a small factor of the EKF.

Implementation and numerical results

Algorithm 9.11 presents the four variants of the IMM algorithm discussed above. With only the mode of the state estimate (variant='EKF'), the IMM is reduced to the EKF algorithm. If we add the adjacent modes (RIMM1), we obtain an improvement on the EKF with at most $2(n + 1)$ modes. When we only consider the adjacent modes close to the state estimate (RIMM2), then the number of modes depends on the tolerance β in Algorithm 9.9. In the last variant (RIMM3), discussed in 9.5, we suppose that the system can only switch between K representative mode vectors.

We implement our algorithms on the same experimental data location as in 9.4. As mentioned in [107], the choice of the transition probabilities only affects slightly the performance of the IMM algorithm. The guideline for a proper choice is to match roughly the transition probabilities with the actual mean sojourn time of each mode. In RIMM1 and RIMM2, it is difficult to estimate the transition probabilities because of the exponential number of modes, so we suppose that the system is equally likely to transition to all the modes. In RIMM3, we take sample transition probabilities from the observed data:

$$\tilde{\pi}_{ij} = \frac{\gamma + \sum_{t=1}^{T} \mathbb{I}(\boldsymbol{\rho}^t \in \mathcal{C}_i,\, \boldsymbol{\rho}^{t+1} \in \mathcal{C}_j)}{\gamma K + \sum_{t=1}^{T-1} \mathbb{I}(\boldsymbol{\rho}^t \in \mathcal{C}_i)} \tag{9.33}$$

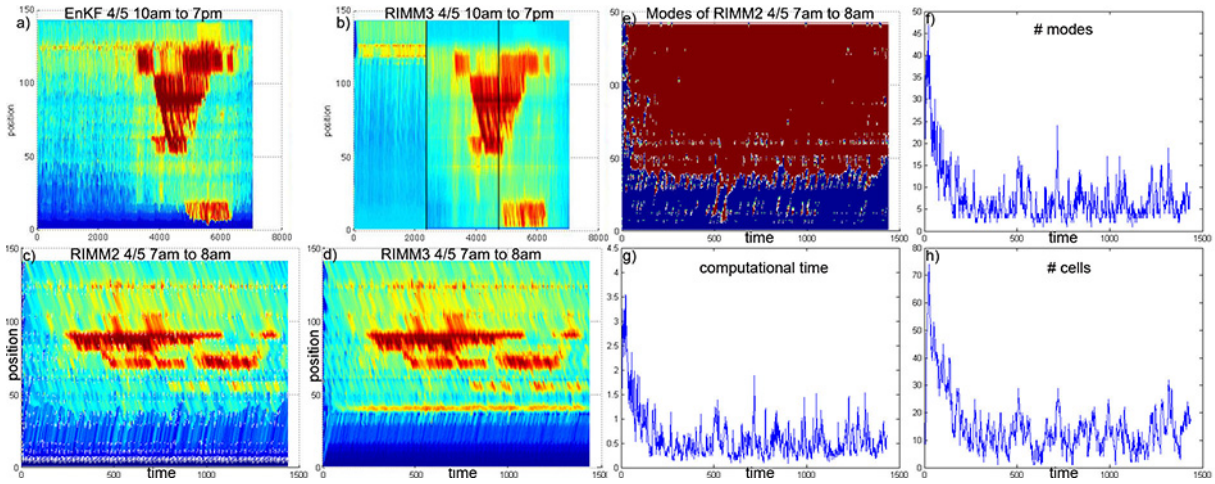

Figure 9.11: Contour plot of the density given by (a) the EnKF with 100 ensembles on May 5th, 10am-7pm, (b) the RIMM3 with 5 clusters on May 5th, at 10am-1pm, 1-4pm, 4-7pm, (c) the RIMM2 with $\beta = 1$ on May 5th, 7-8am, (d) the RIMM3 with 20 clusters using the k-means algorithm on May 5th, 7-8am. Analysis of each time step of the RIMM2 with $\beta = 1$: (e) plot of the mode estimate, (f) number of modes selected by RIMM2, (g) computational time, (h) number of cells with density close to ρ_c.

where the sets $\{\mathcal{C}_k\}_k$ are the Voronoi cells centered on centroids $\{\tilde{\boldsymbol{\rho}}^k\}$ computed in Algorithm 9.10, $\mathbb{I}$ is the indicator function, and γ controls the smoothing from the uniform transitions.

The EnKF is a popular estimation algorithm based on *Monte-Carlo approximation* of the Kalman filter. The results are compared with the EnKF estimate presented in 9.4. Figures 9.9.c,d) 9.11.c,d) present the four estimates which consist in the density in the time-space domain. The regions with high (resp. low) density are represented in red (resp. blue). The estimators give similar higher resolution scalar fields of the density (1440 time steps by 141 cells) by assimilating sparse density measurements (240 time steps by 29 PeMS stations. The shock wave propagation is more noticeable in the output of RIMM estimators in the congested regions.

The density centroids have also been computed to get a set of 5 representative mode vectors for each of the 10am-1pm, 1-4pm, 4-7pm time periods on March 1st, and we applied RIMM3 to estimate the density on March 5th at the same time periods. The estimates are very similar (see Fig 9.11.a,b)).

Figure 9.11.e) shows the mode estimate computed in the *combination step* of the IMM. Each column represents a modal regime of the highway at a specific time. Finally, Figures 9.11.f,g,h) show that the number of modes selected, the computational times, and the number of cells with density close to ρ_c at each time step are proportional.

Algorithm 9.11 IMM with reduced number of modes: **IMM(**algorithm**)**

Require: initial state $\boldsymbol{\rho}_0$, boundary conditions $\{u(t),\, d(t)\}_{t\geq 0}$, state covariance $\{Q^t\}_{t\geq 0}$, observations $\{\boldsymbol{z}^t\}_{t\geq 0}$, observation matrix $\{H^t\}_{t\geq 0}$, observation covariance $\{R^t\}_{t\geq 0}$.

1. $\mathcal{M}^0 = \{\mathbf{rho2m}(\boldsymbol{\rho}_0)\}$ \\initial set of modes is the mode of $\boldsymbol{\rho}_0$
2. **for** $t \in \{0,\, 1,\, 2,\, \cdots\}$:
3. $\quad \boldsymbol{m} = \mathbf{rho2m}(\hat{\boldsymbol{\rho}}^t)$ \\Algo 9.2
4. **if** 'EKF' **then** $\mathcal{M}^{t+1} = \{\boldsymbol{m}\}$
5. **if** 'RIMM1' **then** $\mathcal{M}^{t+1} = \{\boldsymbol{m}\} \cup \mathbf{adj}(\boldsymbol{m})$ \\Algorithm 9.5
6. **if** 'RIMM2' **then** $\mathcal{M}^{t+1} = \{\boldsymbol{m}\} \cup \mathbf{adj2}(\boldsymbol{m}, \hat{\boldsymbol{\rho}}^t, P^t, \beta)$ \\Algorithm 9.9
7. **if** 'RIMM3' **then** $\mathcal{M}^{t+1} = \mathcal{M}^K$ \\Algorithm 9.10
8. **for** $\boldsymbol{m}_j \in \mathcal{M}^{t+1}$
9. $(\hat{\boldsymbol{\rho}}^t_{0j},\, P^t_{0j}) = \mathbf{mixing}((\hat{\boldsymbol{\rho}}^t_i,\, P^t_i,\, \mu^t_i)_{i\in\mathcal{M}^t})$ \\see (9.28)
10. $(\hat{\boldsymbol{\rho}}^{t+1}_j,\, P^{t+1}_j,\, \Lambda^{t+1}_j) = \mathbf{KF}(\boldsymbol{m}_j, \hat{\boldsymbol{\rho}}^t_{0j}, P^t_{0j}, \cdots)$ \\Algorithm 9.8
11. $\mu^{t+1}_j = \mathbf{modeProbUpdate}(\Lambda^{t+1}_j)$ \\see (9.29)
12. $(\hat{\boldsymbol{\rho}}^{t+1}, P^{t+1}, \hat{\boldsymbol{m}}^{t+1}) =$ $\mathbf{combination}((\hat{\boldsymbol{\rho}}^{t+1}_j, P^{t+1}_j, \mu^{t+1}_j)_{j\in\mathcal{M}^{t+1}})$ \\see (9.30)
13. **return** $(\hat{\boldsymbol{\rho}}^t,\, P^t)_{t\geq 0}$

Chapter 10

Fusion of cellular and traffic sensor data for route flow estimation via convex optimization

10.1 Introduction

A new convex optimization framework is developed for the route flow estimation problem from the fusion of vehicle count and cellular network data. The issue of high underdetermined-ness of link flow based methods in transportation networks is investigated, then solved using the proposed concept of *cellpaths* for cellular traces. With this data-driven approach, our proposed approach is versatile: it is model agnostic and thus compatible with user equilibrium, system-optimum, Stackelberg concepts, and other models. Using a dimensionality reduction scheme relying on a particular choice of the nullspace, we design a projected gradient algorithm suitable for the proposed route flow estimation problem for traffic assignment. The algorithm solves a block isotonic regression problem in the projection step in linear time. This chapter has been written following to the best of our abilities practices of *reproducible research*, and we have accordingly posted our code online. The accuracy, computational efficiency, and versatility of the proposed approach are validated on the I-210 corridor near Los Angeles, where we achieve 92% route flow accuracy with 1033 traffic sensors and 950 cellular towers covering a large network of highways and arterials with more than 20,000 links.

While there is a wealth of literature in transportation science aiming at modeling, computing, and estimating the movement of traffic in terms of *link* flows, there is less work focused on *route* flow estimation. The route flow estimation problem is particularly important because route flows estimates can capture phenomena in traffic behavior that link flows cannot. For example, route flows would enable analysis of which commuters a link closure would affect most. Accurate route flow estimates are increasingly critical for a more effective use of existent traffic infrastructure as population density and the need for enhancing mobility in cities grow.

Simultaneously accurate and efficient methods for estimating route flows are crucial to modern traffic engineering, as large scale urban network analysis and planning demands scalable solutions for these problems that can be implemented on sizeable road networks. However, the first step for many approaches to estimating route flow requires enumerating all feasible routes, which is an unreasonable task for many urban road networks because it may require exponential time to compute [67, §1.2].

At the cost of restrictive assumptions, such as *deterministic user equilibrium* (UE) in which each driver is assumed to be rational and have perfect knowledge of the traffic conditions [165], a route (or path) flow can be estimated without requiring route enumeration. Under UE, all routes used to connect an origin-destination (OD) pair have the same cost, hence the distribution of flows across theses routes may not be determined [143, §3.3], [15, §5.2]). The *stochastic user equilibrium* (SUE) (probit-based [52, 111] and logit-based [64, 15]) addresses some of the shortcomings of the UE by modeling imperfect knowledge of the network and variation in drivers' preferences, which makes the estimation of route flows possible. Since there is little evidence of the validity of such models in practice, and real-life transportation networks may not be in equilibrium (or only approximately so) [79], we develop a data-driven framework that focuses on the large amount of traffic data available.

Traffic data sources

Traditional traffic sensing systems such as loop detectors embedded in the pavement and cameras provide accurate volume and speed estimates, but their placements are typically sparse and their information content is too coarse. Most importantly, they measure total counts of vehicles passing through a road segment without distinguishing between vehicles following different routes. In order to partially address the shortage of information on the routes followed by vehicles, other types of static sensors have been deployed on the road network: cameras that measure split ratios at different intersections [159] and plate scanning systems [38, 39]. However these systems require costly infrastructure and only provide highly localized traffic information. Meanwhile, given the large penetration of mobile phones among the driving population and the ubiquitous coverage of service providers in urban areas, mobile phones have become an increasingly popular source of location data for the transportation community. For example, dynamic probing by means of in-car GPS traces [169, 81, 82] is a promising technology for trajectory recovery and travel time estimation. However, the penetration of GPS-enabled devices running a dedicated sensing application currently limits the ability to accurately estimate traffic volumes and it is unlikely that such data would become available to public agencies [129].

In addition to GPS traces, location data are available directly from cellular communication network operators. A variety of phone to cell communication events such as *handovers* (HO), *location updates* (LU) and *call detail records* (CDR) [160, 161] are being recorded by cellular network infrastructures, and this data has already been shown to be effective in studying urban environments [36, 90, 154]. Since typical cellular networks in urban agglomerations include thousands of cells, HO/LU/CDR events can be used effectively to estimate the route

choice and route flow of vehicles without requiring any additional infrastructure. When the user is moving, an HO transfers an ongoing call or data session from one cell to another without disconnecting the session, and an LU allows a mobile device to inform the cellular network when it moves from one location area (or cell) to the next. CDR (mainly used by service providers for billing purposes) contain a timestamped summary of which cell each data transmission came through and therefore contain abundant mobility traces for a majority of the population. Due to the granularity of sensing, these records alone are not sufficient for recovering user routes directly, thereby motivating an inference procedure. The spatial resolution of CDR, HO, and LU data varies with the density of antennas and is roughly proportional to the daytime population density. A standard localization approach when dealing with cellular data is based on Voronoi tessellation, a simple model solely based on the locations of the cell towers [6, 35].

Related work

Several problems within traffic estimation have already benefited from incorporating data from cellular networks: OD matrix computation using cell phone location data [33, 34] such as CDRs [168], link flows estimation [171], and travel time and type of road congestion [88]. These studies vary in scale and assumptions, but they indicate the promise of non-pervasive sensing. In particular, mobile phone data has been used for OD matrix computation. The problem of OD estimation is one of the most well-studied problems in the transportation literature. It historically originates from the first two stages of the four-stage model in traffic planning [130, 127]. For practical purposes, like in the work presented later in the chapter studying the I-210 corridor near Los Angeles, we will use an OD model available from local transportation agencies (in the present case, the SCAG model). There are many surveys on the subject in the past decades [15, 2, 125], and the accuracy of OD estimates will continue to improve.

Extracting the set of potential route choices between all OD pairs is also a well-studied problem. Traditionally, the set of potential routes can be extracted from the induced equilibrium flows in equilibrium-based models. In recent years, the growing number of mobile sensors in urban areas enables the use of probe vehicles for route inference from GPS traces [82, 134]. There are also early studies on the use of cellular network data for traffic assignment: [148] estimates the route choice for each user in the cellular network using a distance measure to determine the best matching route and also incorporate additional constraints from travel time and user equilibrium. Their small experiment (2-4 routes) performed via a macro-simulator indicates the potential for cellular network data to add valuable information for solving this problem. However, a recent survey on the use of wireless signals for road traffic detection [117] concluded that there is thus far no existing system that can estimate traffic densities in a practical sense, that is, in terms of scalability, coverage, cost, and reliability, thus motivating our work on estimating route flows.

Contributions of this chapter

One of the key innovations of the present work is generalizing the common notion of an OD matrix to a general form of coarse route flow measurements (here collected from cellular network data). As mentioned above, the problem of traffic assignment is historically highly underdetermined because the OD matrix and link flows (even when all the links are observed) contains relatively little information as compared to the number of routes people can take. To address this fundamental problem, we introduce the framework of *cellpaths*, which generalizes 2-point network flow, which we call *OD flow*, to n-point network flow, which we call *cellpath flow*. OD flow, which is the number of vehicles that originates at some origin and terminates at some destination, can be characterized by 2 region centroids (illustrated in Figure 10.3). Similarly, cellpath flow can be characterized by n region centroids. In this chapter, the centroids for cellpath flow correspond to cellular base stations, and the centroids for OD flows correspond to centroids of Traffic Analysis Zones (TAZ). Since our approach includes a "strict" generalization of ODs to cellpaths, the methodology presented in this chapter can be applied to a variety of traffic assignment problems.

Now, we define our problem as follows: given a large-scale road network in the quasi-static regime, a set of OD demands, a set of admissible routes between each OD pairs, cellpath flow estimates along the network, and link flow measurements on a subset of links in the network, our goal is to develop a method to estimate the distribution of flow over the set of routes. We pose this problem as a convex optimization program in which the cellular network traces are assigned to the constraints, and the objective encodes link sensor data and the OD matrix. Convex optimization techniques have been used quite frequently by the transportation community for diverse purposes. For example, the classical Wardrop equilibrium approach to the traffic assignment problem can be formulated as a convex optimization program given some typical assumptions on the link performance (or delay) functions [143]. Recent works often combine convex optimization with machine learning techniques [22, 144, 115].

Another key component of our approach is the analysis of the constraints of our convex optimization program. We reformulate them as block-simplex constraints and we apply a standard equality constraint elimination technique [26, §4.2.4] with a particular change of variable which converts the non-negativity constraints on the variables to ordering constraints. In the new space induced by the change of variables, we show that the projection on the feasible set (characterized by the ordering constraints) can be performed in linear time via bounded isotonic regression (see [153] for a short survey on isotonic regression). Then we solve our convex optimization program with an accelerated first order or second order projected descent algorithm. The change of variables presents two main advantages: the dimensionality is reduced (sometimes by a factor 2) which is critical for large-scale problems, and we can perform the projection in $\mathrm{O}(n)$, an improvement over $\mathrm{O}(n \log n)$ required by the projection onto the simplex [86, 162], where n is the number of routes per OD pair. In addition, it is worth noting that a wide variety of problems can benefit from this methodology. First, the use of algorithms that feature a projection step, e.g. projected descent methods and alternating direction methods, is very popular since they often provide a simple and efficient

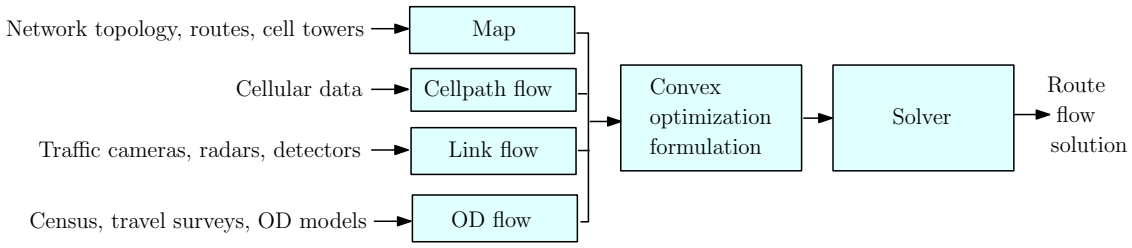

Figure 10.1: Future route flow estimation pipeline, from raw data to route flows, including: 1) aggregation of the link flow obtained from traffic sensors will be over a sizable duration (*e.g.* 1 hour) suitable to address the static estimation problem; 2) a state-of-the-art OD matrix estimation method; 3) a trip analysis step to filter driver cellular traces from other traces, group them by their cellpaths, and map them to cellpath flow values; 4) a route inference method using cellular/GPS traces; 5) data fusion from census, link sensors, GPS/cellular traces, travel surveys etc. 6) an improved solver that benefits from the multi-sensor data.

way to solve constrained convex optimization problem as opposed to more specialized active set methods. There is also a great deal of applications that feature simplex constraints, such as the aforementioned traffic assignment problem and games in general for the computation of strategy distributions, and ℓ_1-based approach in machine learning [86].

Throughout our analysis, a particular emphasis is placed on a data-driven approach that benefits from the sheer amount of cell data without relying on equilibrium-based models, since in practice traffic flow in large urban areas may not be (or just approximately) in equilibrium and there is no sufficient data to access one way or another. Aiming at a real-world production system pipeline summarized in Figure 10.1, we prove the versatility and data-driven nature of the proposed approach via validation on three datasets produced by two simulators of vehicle traffic link flows, route flows, and cellpaths based on the positions of cell towers in the region of interest. The positions of the cell towers are sampled randomly on the urban network to have full flexibility on the parameters of the simulators. We develop an equilibrium-based model[1] that generates user equilibrium (UE) and system-optimal (SO) flows on the I-210 corridor near Los Angeles, CA (containing 44 nodes, 122 links). The first simulator serves two purposes: it highlights the accurate recovery of route flows, even for quite sparse cellular networks and it provides empirical explanations for the efficiency of our method. We also use MATSim agent-based transport simulator[2] on a large-scale urban road network near Los Angeles, CA (with more than 20K links and 290K routes) to showcase the performance of our methodology on large datasets. We demonstrate that our full pipeline, from the simulators to the procedures to estimate static route flows on small and large-scale urban network, can be extended easily to incorporate other types of data such as link capacities, split ratios etc. Hence we hope that our framework will be a benchmark for many future studies in estimation

[1]The code is available on Github: `https://github.com/jeromethai/traffic-estimation-wardrop`.

[2]MATSim is an open source project (http://www.matsim.org), and related publications are available here: http://www.matsim.org/publications.

problems in transportation science.

We summarize the contributions of the presented work:

- We propose a convex optimization formulation for the route flow estimation problem which uses a new data fusion approach for loop detectors counts and cellular signal traces (ubiquitous among the driving population).
- We demonstrate that our formulation is also compatible with several other approaches to this problem, including equilibrium concepts, which may be used in conjunction for improved estimation.
- We introduce the concept of *cellpaths* and demonstrate its application to traffic estimation problems. We address the issue of highly underdetermined-ness of link flow based methods (which was already raised in the traffic assignment literature) by formalizing cellular data as cellpaths and incorporating them as constraints. Though we focus on the route flow estimation problem, many traffic problems may benefit from such an approach.
- Using a reduction scheme, we design an algorithm to solve the route flow estimation problem and large-scale traffic assignment problems in general. In the resulting formulation, the projection step can be performed in $\mathrm{O}(n)$ via isotonic regression, an improvement over $\mathrm{O}(n \log n)$, where n is the number of routes per OD pair.
- We present a full system pipeline from cellular network and link flow data to estimate the static route flow (and as a by-product, link flow) on a large-scale urban network. We demonstrate the first system to our knowledge that can produce route-level flow estimates suitable for short time horizon prediction and control applications in traffic management from the fusion of cellular network data and data from static sensors along roads.
- We present numerical results from different sets of small and large-scale datasets for Los Angeles. In particular, the emphasis is placed on a data-driven approach: it is versatile to different types of vehicular behavior.

The remainder of the chapter is organized as follows: In Section 10.2, we present the setup and assumptions of our work, then formulate our route estimation problem as a convex optimization program. We also provide a re-formulation necessary for the algorithmic approach described in Section 10.3. Further in Section 10.3, we develop a specialized projected gradient method to solve convex optimization programs with simplex constraints. Section 10.4 is dedicated to the setting of our experiments. Section 10.5 presents our numerical results. Section 6 concludes the paper by placing the presented method within a general data-driven traffic estimation framework and identifying directions for future work.

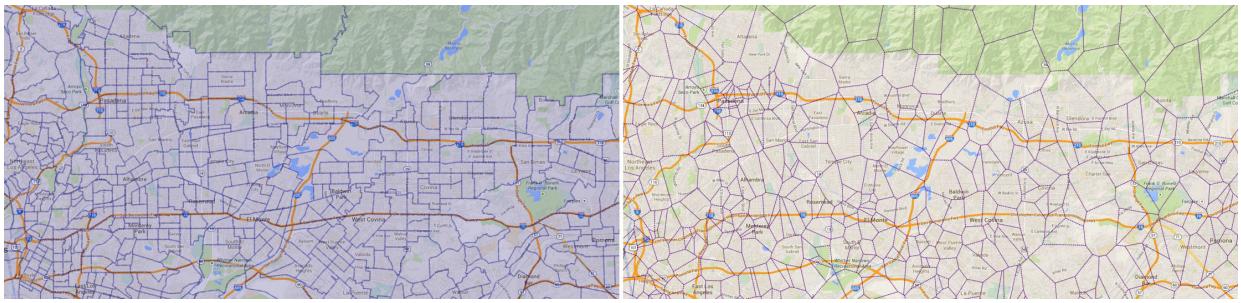

Figure 10.2: I-210 corridor in Los Angeles county used for the numerical work presented in §10.5. Left subfigure: The 700 regions are origin/destinations areas called Traffic Analysis Zones (TAZ) used for the numerical experiments. Right subfigure: Corresponding Voronoi partition of the cellular network based on the cell tower locations. [Note: figure best viewed in color.]

10.2 Problem formulation

Problem setup and assumptions

We define the terminology used in the chapter, and the notations are presented in Table 10.1. It is important to distinguish between four types of flows: cellpath flow, link flow, route flow, and OD flow.

- **Origins**: traffic regions with its associated centroid, defined by a partitioning of the *road network*. Each region is both an *origin* (its centroid is a source from which trips emanate) and a *destination* (its centroid is a sink at which trips terminate). A possible implementation of the method proposed can be done by taking the origins/destinations to be the Traffic Analysis Zones (TAZ) (see Figure 10.2) as done in the numerical work late in the chapter. We define *OD flow* to be the flow (vehicle count) that originates and terminates with an OD pair.

- **Cells**: regions defined by the Voronoi partition of the *cellular network*, they are generally different from ODs.

- **Cellpath**: a sequence of cells, and we define *cellpath flow* to be the flow (vehicle count) along a cellpath.

- **Link**: a segment of road in the network, and the *link flow* is the flow (vehicle count) through a link.

- **Route**: a sequence of links from an origin to a destination. Each route also has an associated unique cellpath. The *route flow* is the flow (vehicle count) on the route.

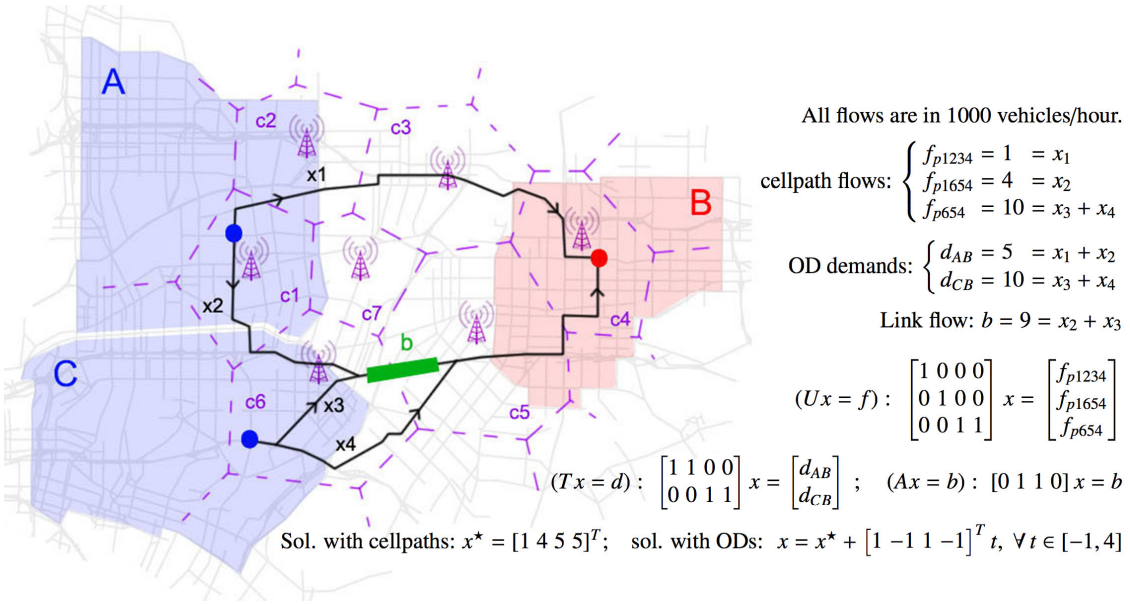

Figure 10.3: In this illustration of the cellular and loop data fusion, we have two origins A and C (the blue traffic regions and their centroid as blue dots) and one destination B (the red traffic region). We have routes r_1, r_2, r_3, r_4 with flows $\mathbf{x} = (x_1, x_2, x_3, x_4)$ such that r_1, r_2 go from A to B and r_3, r_4 go from C to B. Cells $c_1, \cdots, c_7$ are shown in purple dashed regions. Since route r_1 goes through cells c_1, c_2, c_3, c_4, its associated cellpath is p_{1234}. Similarly, routes r_2, r_3, r_4 have cellpaths $p_{1654}, p_{654}, p_{654}$ respectively. Let f_{p1234}, f_{p1654}, f_{p654} be the cellpath flows (obtained from cellular network data), *i.e.* there are f_{p1234}=1000 veh/h going through c_1, c_2, c_3, c_4. Let d_{AB} and d_{CB} be the OD demands. Cellpaths p_{1234} and p_{1654} disambiguate routes between AB: $f_{p1234} = x_1$, $f_{p1654} = x_2$, contrary to the ODs: $d_{AB} = x_1 + x_2$. However, cell towers are not dense along r_3, r_4, hence $d_{CB} = f_{p654} = x_3 + x_4$. The cellpath-route incidence matrix generalizes OD matrices since we consider the sequence of intermediate regions (cells here) that intersect with trips. We also have $x_2 + x_3 = b$, with b the flow on the green link (from loop detectors). There is a unique route flow inducing flows b, f_{p1234}, f_{p1654}, f_{p654} that is $\mathbf{x}^\star = [1\ 4\ 5\ 5]$, while there are infinitely many flows inducing b, d_{AB}, d_{CB}: $\mathbf{x} = \mathbf{x}^\star + [1\ {-1}\ 1\ {-1}]^T t$, $\forall\, t \in [-1, 4]$, so the problem has *one degree of freedom* and is *underdetermined* with only the OD demands as data.

The link-route incidence matrix $\mathbf{A}$ encodes the network topology (which routes $r \in \mathcal{R}$ contains which links $l \in \mathcal{L}$); the cellpath-route incidence matrix $\mathbf{U}$ encodes the collection of routes with the same cellpaths (which routes r is associated to which cellpath p); and the

Notation	Description
$\mathcal{O}$, $\mathcal{D}$	Set of origins/destinations $\mathcal{D} = \mathcal{O}$
$\mathcal{L}$	$\operatorname{card}\mathcal{L} = m$, links with observed flow
$\mathcal{P}$	$\operatorname{card}\mathcal{P} = q$, observed cellpaths
$\mathcal{R}$	$\operatorname{card}\mathcal{R} = n$, set of routes
$\mathcal{E}$	Set of all links in the network
$\mathbf{A} \in \{0,1\}^{\lvert\mathcal{L}\rvert\times\lvert\mathcal{R}\rvert}$	Link-route incidence matrix
$\mathbf{A}^{\text{full}} \in \{0,1\}^{\lvert\mathcal{E}\rvert\times\lvert\mathcal{R}\rvert}$	Full link-route incidence matrix
$\mathbf{U} \in \{0,1\}^{\lvert\mathcal{P}\rvert\times\lvert\mathcal{R}\rvert}$	cellpath-route incidence matrix
$\mathbf{T} \in \{0,1\}^{\lvert\mathcal{O}\rvert^2\times\lvert\mathcal{R}\rvert}$	OD-route incidence matrix
$\mathbf{d} \in \mathbb{R}^{\lvert\mathcal{O}\rvert^2}$	Vector of OD flows, $\mathbf{d} = (d_k)_{k\in\mathcal{O}^2}$
$\mathbf{b} \in \mathbb{R}^{\lvert\mathcal{L}\rvert}$	Observed link flow vector, $\mathbf{b} = (b_l)_{l\in\mathcal{L}}$
$\mathbf{f} \in \mathbb{R}^{\lvert\mathcal{P}\rvert}$	Cellpath flows vector, $\mathbf{f} = (f_p)_{p\in\mathcal{P}}$
$\mathbf{x} \in \mathbb{R}^{\lvert\mathcal{R}\rvert}$	Vector of route flows $\mathbf{x} = (x_r)_{r\in\mathcal{R}}$
$\mathbf{v} \in \mathbb{R}_+^{\lvert\mathcal{E}\rvert}$	Full link flow vector, $\mathbf{v} = (v_e)_{e\in\mathcal{E}}$
Subset $\mathcal{R}^p$	Subset of $n_p := \operatorname{card}\mathcal{R}^p$ routes with cellpath p
$\tilde{\mathbf{x}}^p \in [0,1]^{\lvert\mathcal{R}^p\rvert}$	Ratios of flows across routes $r \in \mathcal{R}^p$
$\mathbf{x}^p \in \mathbb{R}_+^{n_p}$	$\mathbf{x}_r^p$ is the flow of route $r \in \mathcal{R}^p$
$\mathcal{R}^k \subset \mathcal{R}$	Subset of n_k routes between OD pair k

Table 10.1: Notation for route estimation problem. We have m observed links, q cellpaths, n routes.

OD-route incidence matrix $\mathbf{T}$ encodes which route r is between OD pair k.[3]

$$\text{link-route: } \mathbf{A}_{lr} = \begin{cases} 1 & \text{if } l \in r \\ 0 & \text{else} \end{cases}; \quad \text{cellpath-route: } \mathbf{U}_{pr} = \begin{cases} 1 & \text{if } r \in \mathcal{R}^p \\ 0 & \text{else} \end{cases};$$

$$\text{OD-route: } \mathbf{T}_{kr} = \begin{cases} 1 & \text{if } r \in \mathcal{R}^k \\ 0 & \text{else} \end{cases}$$

The model assumptions are as follows:

- We consider a quasi-static setting, where traffic demands (flows) remain constant over time, and we focus on the noiseless case, with a short commentary on the noisy case in Section 10.5.

[3]The lowercase letters l, r, p, k written as subscripts refer to the indices associated to links, routes, cellpaths, and ODs respectively.

- Since enumerating all routes is not tractable, we consider the top routes between each OD pair following different criteria depending on the setting of the numerical experiment (see Section 10.4).
- We know the cellpath flow μ_p (vehicle count) from the cellular traces (phone count) along each cellpath p.
- All cellpaths $p \in \mathcal{P}$ are *contiguous*: each pair of consecutive cells in p shares a boundary.
- The set of cellpaths $\mathcal{P}$ is *well-posed*: there exists a unique cellpath $p \in \mathcal{P}$ for each route $r \in \mathcal{R}$, and we have a cellpath flow measurement μ_p for each $w \in \mathcal{P}$.

Formulation and analysis of the model

The fusion of cellular and loop data for route flow estimation is one of the key contributions of this chapter. We wish to find an assignment of route flow $\mathbf{x}$ that agrees with the cellpath flow $\mathbf{f}$ distributed across the routes $\mathcal{R}$ such that the measurement residual with the link flow $\mathbf{b}$ is minimized. We formulate the problem in the framework of convex optimization as a minimization of a quadratic program:

$$\begin{array}{rl} \min & \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \\ \text{s.t.} & \mathbf{U}\mathbf{x} = \mathbf{f},\, \mathbf{x} \succeq 0 \end{array} \tag{10.1}$$

The problem is a constrained linear inverse problem in which we want to estimate a signal of length n (the route flows) given that we have m measurements (the observable link flows). We additionally have q cellpath flow constraints: for each cellpath $p \in \mathcal{P}$, there are n_p routes corresponding to p, such that their flow must sum up to the cellpath flow f_p:

$$\mathbf{U}\mathbf{x} = \mathbf{f}: \quad \sum_{r \in \mathcal{R}^p} x_r^p = f_p \quad \forall\, p \in \mathcal{P} \tag{10.2}$$

In general, $m \ll n$ and $q \leq n$, thus typically the Hessian $\mathbf{A}^T\mathbf{A}$ of our convex quadratic objective is singular ($\mathbf{A}^T\mathbf{A} \in \mathbb{R}^n$ but $\operatorname{rank}(\mathbf{A}^T\mathbf{A}) \leq m \ll n$). Thus the problem might have multiple optimal solutions (underdetermined) or might have more observations than unknowns (overdetermined), depending on the number of cellpath flow constraints. In contrast with methods that consider less detailed flow measurements (e.g. OD flow) instead of cellpath flow, however, our formulation encodes more constraints than past methods, thereby constraining the solution space. Moreover, when there are uncorrelated measurement errors on the vector flow $\mathbf{b}$ (absence of interactions between the detection process of static sensors), the ordinary least squares is the best unbiased estimator of the route flows $\mathbf{x}$.[4]

Our model is related to the so-called *route assignment problem* used to solve traffic equilibrium problems [165], [143, §3], [15, §5], where $\mathcal{E}$ is the set of all links (edges) in the

[4]The errors must also have zero-mean and constant variance, then the result holds as link flows linearly depend on route flows: $\hat{\mathbf{b}} = \mathbf{A}\mathbf{x} + \boldsymbol{\epsilon}$, from the Gauss-Markov theorem.

network, $\mathbf{A}^{\text{full}} \in [0,1]^{|\mathcal{E}|\times|\mathcal{R}|}$ is the full link-route incidence matrix, and ϕ is the Beckmann objective function [13]:

$$\min \phi(\mathbf{A}^{\text{full}}\mathbf{x}) \quad \text{s.t.} \quad \mathbf{T}\mathbf{x} = \mathbf{d},\, \mathbf{x} \succeq 0 \tag{10.3}$$

This is a standard formulation in traffic assignment in which a local minimum of (10.3) is a Wardrop equilibrium of a congestion game [120]. If the cellpath-route incidence matrix U is reduced to an OD-route incidence matrix (see Fig. 10.3), both (10.1) and (10.3) share the same constraints. Reversely, the constraints $Ux = f$ can be added to (10.3) to restrict its solution space. The main difference lies in the objective being minimized: in (10.1) it is the link flows measurement residual while in (10.3) the potential ϕ expresses the incentives of all vehicles (or players) to take the shortest route.

Proposition 10.1. *Problem (10.1) can be reduced to a least-squares problem with (separable) simplex constraints:*

$$\min \frac{1}{2}\|\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} - \boldsymbol{b}\|_2^2 \quad s.t. \quad \boldsymbol{1}^T\tilde{\boldsymbol{x}}^p = 1,\, \tilde{\boldsymbol{x}}^p \succeq 0, \quad \forall\, p \in \mathcal{P} \tag{10.4}$$

$$where \quad \tilde{\boldsymbol{A}} \in \mathbb{R}_+^{|\mathcal{L}|\times|\mathcal{R}|} \;:\; \tilde{\boldsymbol{A}}_{lr} = \begin{cases} f_p & if\ l \in r \in \mathcal{R}^p \\ 0 & else \end{cases} \tag{10.5}$$

where $\boldsymbol{1} = [1,\cdots,1]^T \in \mathbb{R}^{n_p}$ and $\tilde{\boldsymbol{A}}$ is a modified link-route incidence matrix containing the cellpath flows f_p.

Proof. The constraints $\mathbf{U}\mathbf{x} = \mathbf{f}$ in (10.1) can be written explicitly: $\sum_{r\in\mathcal{R}^p} x_r^p = f_p,\ \forall\, p \in \mathcal{P}$. With the change of variables $\tilde{\mathbf{x}}^p := \mathbf{x}^p/f_p$ for all p, the constraints become $\sum_{r\in\mathcal{R}^p} \tilde{x}_r^p = 1,\ \forall\, p \in \mathcal{P}$, or in matrix form: $\mathbf{1}^T\tilde{\mathbf{x}}^p = 1,\ \forall\, p \in \mathcal{P}$. Since $f_p > 0$ for all p, then the inequalities $\mathbf{x}^p \succeq 0$ are equivalent to $\tilde{\mathbf{x}}^p = \mathbf{x}^p/f_p \succeq 0$. Finally, the vector $\mathbf{A}\mathbf{x}$ has entries $v_l = \sum_{r\,:\,l\in r} x_r$ for $l \in \mathcal{L}$, where v_l is the flow on link l. The sum can be decomposed between the different cellpaths p: $v_l = \sum\limits_{r\,:\,l\in r} x_r = \sum\limits_p \left\{ \sum\limits_{r\,:\,l\in r\in\mathcal{R}^p} x_r^p \right\} = \sum\limits_p \left\{ \sum\limits_{r\,:\,l\in r\in\mathcal{R}^p} f_p\tilde{x}_r^p \right\} = (\tilde{\mathbf{A}}\tilde{\mathbf{x}})_l$, hence the objectives are the same. □

In the context of game theory, each cellpath can be seen as a player who choses a strategy or a probability distribution with weights $(\tilde{x}_r^p)_{r\in\mathcal{R}^p}$ over the n_p routes, and a set defined by $S^p := \{\tilde{\mathbf{x}}^p \in [0,1]^{n^p} \mid \sum_{r\in\mathcal{R}^p} \tilde{x}_r^p = 1\}$ is a *strategy set* or a *probability simplex* over the routes $r \in \mathcal{R}^p$. We note that the sets S^p are disjoint (each route has at most one cellpath associated to it), hence (10.4) is a least-squares problem with (separable) simplex constraints. In the presence of noisy cellpath flow data, (10.4) is more adequate than (10.1) since we do not restrict the sum of route flows to be equal to the cellpath flows (10.2). Intead, matrix $\tilde{\mathbf{A}}$ is noisy in (10.4). Note that the traffic assignment problem (10.3) can also be reduced in a

similar fashion, where $\mathcal{O}^2$ is the set of all OD pairs, which results in the approach proportions of Bar-Gera [8]:

$$\begin{array}{ll}\min & \phi(\tilde{\mathbf{A}}^{\text{full}}\tilde{\mathbf{x}}) \\ \text{s.t.} & \mathbf{1}^T\tilde{\mathbf{x}} = 1,\, \tilde{\mathbf{x}}^k \succeq 0,\, \forall\, k \in \mathcal{O}^2\end{array} \qquad \text{where} \quad \tilde{\mathbf{A}}^{\text{full}} = \begin{cases} d_k & \text{if } l \in r \in \mathcal{R}^k \\ 0 & \text{else} \end{cases} \tag{10.6}$$

Before we conclude the section, we note that our model (10.1) is also compatible with several other types of data, *e.g.* turning ratios and links' capacities. If, at some intersection $j \in \mathcal{V}$, we know the flow of vehicles coming from link $e = (i, j) \in \mathcal{E}$ and turning into link $e' = (j, k)$, and we denote the pair of successive links by $t = (e, e')$, then letting $\mathcal{T}$ be the set of monitored traffic turns (intersections) t, $G \in \{0, 1\}^{|\mathcal{T}|\times|\mathcal{R}|}$ the turn-route incidence matrix, and h the vector of flow that passes through each monitored intersection the objective of (10.1) can be generalized to include split ratios:

$$\min \frac{1}{2}\|\mathbf{A}'\mathbf{x} - \mathbf{b}'\|_2^2 \quad \text{s.t.} \quad \mathbf{U}\mathbf{x} = \mathbf{f},\ \mathbf{x} \succeq 0 \tag{10.7}$$

$$\text{where} \quad \mathbf{A}' = \begin{bmatrix}\mathbf{A} \\ \mathbf{G}\end{bmatrix}, \quad \mathbf{b}' = \begin{bmatrix}\mathbf{b} \\ \mathbf{h}\end{bmatrix} \quad \text{and} \quad \mathbf{G}_{tr} = \begin{cases} 1 & \text{if} \quad t = (a, a') \,:\, a, a' \in r \\ 0 & \text{otherwise} \end{cases} \tag{10.8}$$

Suppose we know the link capacities $\tilde{m}_e$, then the constraints $\mathbf{A}^{\text{full}}\mathbf{x} \preceq \tilde{\mathbf{m}}$, where $\tilde{\mathbf{m}} := (\tilde{m}_e)_{e\in\mathcal{E}}$ is the link capacities vector, can be added to program (10.1). To approximate the new problem as a program with simplex constraints, we can make the added constraints implicit in the objective:

$$\min \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \sum_{e\in\mathcal{E}} \phi(L_e^T\mathbf{x} - \tilde{m}_e) \quad \text{s.t.} \quad \mathbf{U}\mathbf{x} = \mathbf{f},\ \mathbf{x} \succeq 0 \tag{10.9}$$

where the barrier ϕ is an approximation of the indicator function $I_- : \mathbb{R} \to \mathbb{R}$ for the nonpositive reals, and the vectors L_e^T, $e \in \mathcal{E}$ are the rows of $\mathbf{A}^{\text{full}}$. A common choice for ϕ is the logarithm barrier $\phi(u) = -\alpha \log(-u)$ where α is a parameter that sets the accuracy of the approximation [26, §11.2.1].

To summarize, we choose with model (10.1) a data-driven approach in which we want to find the best statistical estimator of the linear route-link flow model given the observed cellpath flows, which contrasts from equilibrium-based route flow assignment models. Both models are still very similar since they have the same simplex constraints. Our formulation can also include other types of data.

10.3 Dimensionality reduction and projection via isotonic regression

In this section, we present an efficient constraint elimination technique relying on the choice of a particular nullspace, which is suitable for both the proposed route flow estimation problem

(10.1) and the traffic assignment problem (10.3). The projection on the inequality constraints is performed in linear time via isotonic regression.

Exploiting the structure of the equality constraints

We consider the reduced route flow estimation problem (10.4) and the reduced traffic assignment problem (10.3):

$$\begin{aligned} &\text{route flow estimation problem:} && \min_{\mathbf{x}} \tfrac{1}{2}\|\tilde{\mathbf{A}}\tilde{\mathbf{x}} - \mathbf{b}\|_2^2 \quad \text{s.t.} \quad \mathbf{1}^T\tilde{\mathbf{x}}^p = 1,\ \tilde{\mathbf{x}}^p \succeq 0, \quad \forall\, p \in \mathcal{P} \\ &\text{traffic assignment problem:} && \min_{\mathbf{x}} \phi(\tilde{\mathbf{A}}^{\text{full}}\tilde{\mathbf{x}}) \quad \text{s.t.} \quad \mathbf{1}^T\tilde{\mathbf{x}}^k = 1,\ \tilde{\mathbf{x}}^k \succeq 0, \quad \forall\, k \in \mathcal{O}^2 \end{aligned} \tag{10.10}$$

We consider a general objective function f and the simplexes $S^p = \{\tilde{\mathbf{x}}^p \in [0,1]^{n^p} \mid \sum_{r\in\mathcal{R}^p} \tilde{x}_r^p = 1\}$ as constraints, but the following analysis applies for both problems. We use standard linear algebra operations to eliminate the equality constraints [26, §4.2.4]. Since the constraints have disjoint support, we treat each one of them separately. For all $p \in \mathcal{P}$, we find a direction e^p which is a particular solution of $\mathbf{1}^T\tilde{\mathbf{x}}^p = 1$, and a matrix $\mathbf{N}^p$ whose range is the *orthogonal complement* of the vector $\mathbf{1} \in \mathbb{R}^{n_p}$, denoted $\{t\,\mathbf{1} \mid t \in \mathbb{R}\}^\perp$. With the vectors $\{\mathbf{e}^p\}_{p\in\mathcal{P}}$ stacked into an overall vector $\tilde{\mathbf{x}}_0 := (\mathbf{e}^p)_{p\in\mathcal{P}}$, and the matrices $\{N^p\}_{p\in\mathcal{P}}$ encoded in an overall block-diagonal matrix $\mathbf{N} := \operatorname{diag}((\mathbf{N}^p)_{p\in\mathcal{P}})$, the resulting problem is:

$$\min_{\mathbf{z}} \frac{1}{2} f(\tilde{\mathbf{x}}_0 + \mathbf{N}\mathbf{z}) \quad \text{s.t.} \quad \tilde{\mathbf{x}}_0 + \mathbf{N}\mathbf{z} \succeq 0 \tag{10.11}$$

$$\text{or with the blocks made explicit:} \quad \begin{array}{ll} \min\limits_{\mathbf{z}} & f((\mathbf{e}^p + \mathbf{N}^p\mathbf{z}^p)_{p\in\mathcal{P}}) \\ \text{s.t.} & \mathbf{e}^p + \mathbf{N}^p\mathbf{z}^p \succeq 0, \quad \forall\, p \in \mathcal{P} \end{array} \tag{10.12}$$

Vectors of the form $[\cdots, 1, -1, \cdots]^T$ are orthogonal to $\mathbf{1} \in \mathbb{R}^{n_p}$. We also choose a simple $\mathbf{e}^p$ solution of $\mathbf{1}^T\mathbf{x}^p = 1$:

$$\mathbf{e}^p := [0, \cdots, 0, 1]^T \in \mathbb{R}^{n_p}; \quad \mathbf{N}^p = \begin{bmatrix} 1 & & \\ -1 & 1 & \\ & -1 & \ddots \\ & & \ddots \end{bmatrix} \in \mathbb{R}^{n_p \times (n_p - 1)} \quad \forall\, p \in \mathcal{P} \tag{10.13}$$

where the columns of $\mathbf{N}^p$ form a basis of $\{t\,\mathbf{1} \mid t \in \mathbb{R}\}^\perp$. These choices result in a simplification of the constraints in (10.12), and we can interchangeably operate on variables $\mathbf{x}^p$ in (10.1) and variables $\mathbf{z}^p$ in (10.4) since they are simply related:

$$\begin{aligned} \tilde{\mathbf{x}}^p &= \mathbf{e}^p + \mathbf{N}^p z^p = [\mathbf{z}_1^p,\ \mathbf{z}_2^p - \mathbf{z}_1^p,\ \cdots,\ \mathbf{z}_n^p - \mathbf{z}_{n-1}^p,\ 1 - \mathbf{z}_{n_p}^p]^T,\ \forall\, p \in \mathcal{P} \\ \mathbf{z}^p &= [\tilde{x}_1^p,\ \tilde{x}_1^p + \tilde{x}_2^p, \cdots,\ \textstyle\sum_{i=1}^{n-2} \tilde{x}_i^p,\ \sum_{i=1}^{n-1} \tilde{x}_i^p,]^T,\ \forall\, p \in \mathcal{P} \end{aligned} \tag{10.14}$$

The constraint $\mathbf{e}^p + \mathbf{N}^p\mathbf{z}^p \succeq 0$ becomes an ordering constraint $0 \le z_1^p \le \cdots \le z_{n_p-1}^p \le 1$. The program (10.12) is now:

$$\min_{\mathbf{z}} f((e^p + N^p z^p)_{p\in\mathcal{P}}) \quad \text{s.t.} \quad 0 \le z_1^p \le \cdots \le z_{n_p-1}^p \le 1, \quad \forall\, p \in \mathcal{P} \tag{10.15}$$

The main advantage of this constraint elimination is the reduction of the dimension from n to $n-q$, where n is the number of routes and q the number of cellpaths (see Table 10.1). If each cellpath has maximum k routes, then we have $n \leq kq$, hence $n - q \leq n(1 - 1/k)$. For our target problem, we generally have $k = 3$ hence the dimension is reduced by at least a factor 1/3.

The problem (10.15) can be solved quite efficiently with a simple (accelerated) first order or second order projection algorithm, or an Augmented Lagrangian method. In particular, the basic descent projection algorithm (see Algorithm 7.2) iteratively takes a step in a descent direction $\Delta\mathbf{z}$ (line 2) from the current point $\mathbf{z}$, projects the new point $\mathbf{z} + \Delta\mathbf{z}$ onto the constraint set $\mathbf{z}^+ := \Pi(\mathbf{z} + \Delta\mathbf{z})$ (line 3), and performs a line search (line 4). The projection step is performed with q Euclidean projections of $\mathbf{z}^p + \Delta\mathbf{z}^p$ onto ordering constraints:

$$\Pi^p(\mathbf{y}^p): \quad \min_{\mathbf{u}^p} \|\mathbf{u}^p - \mathbf{y}^p\|_2^2 \quad \text{s.t.} \quad 0 \leq u_1^p \leq u_2^p \leq \cdots \leq u_{n_p-1}^p \leq 1 \quad \forall\, p \in \mathcal{P} \tag{10.16}$$

Algorithm 10.1 `Proj-descent(·)` General projected descent method

Require: initial point $\mathbf{z} = (\mathbf{z}^p)_{p\in\mathcal{P}}$ in the feasible set $\mathcal{X}$.
1. **while** stopping criteria not met **do**
2. Determine a descent direction $\Delta\mathbf{z} = (\Delta\mathbf{z}^p)_{p\in\mathcal{P}}$
3. Projection: $(\mathbf{z}^p)^+ := \operatorname{argmin}_{\mathbf{u}^p} \{\|\mathbf{z}^p + \Delta\mathbf{z}^p - \mathbf{u}^p\|^2 : 0 \leq u_1^p \leq, \cdots \leq u_{n_p-1}^p \leq 1\}, \forall\, p \in \mathcal{P}$
4. Line search on the projected arc: $\gamma :\approx \operatorname{argmin} \{f(\mathbf{z} + t(\mathbf{z}^+ - \mathbf{z})) : t \in [0,1]\}$
5. $\mathbf{z} := \mathbf{z} + \gamma(\mathbf{z}^+ - \mathbf{z})$
6. **return z**

In line 4 of Algorithm 7.2, we perform a backtracking line search [26, §9.2]. This is an Armijo-rule based step size selection that ensures sufficient descent, it approximately minimizes the objective along the projected arc $\{\mathbf{z} + t(\mathbf{z}^+ - \mathbf{z}) \,|\, t \in [0,1]\}$. Since the feasible set is convex, the projected arc is feasible, hence the method also ensures feasibility of the next iterate. We apply backtracking with objective $f(\mathbf{z}) = \|\mathbf{A}(\tilde{\mathbf{x}}_0 + \mathbf{N}\mathbf{z})\|_2^2$ and descent direction $\mathbf{d} = \mathbf{z}^+ - \mathbf{z}$.

A simple projection using isotonic regression

The projections (10.16) have general form (10.17), given data points $\mathbf{y} := [y_1, \cdots, y_n] \in \mathbb{R}^n$, weights $\mathbf{w} := [w_1, \cdots, w_n] \succ 0$, and bounds $L < U$.[5] Without bounds, we have an isotonic regression problem (10.18) (see [153] and references therein).

$$\text{ISO}_{1\to n}^{[L,U]}(\mathbf{y}, \mathbf{w}): \quad \min_{\mathbf{u}} \textstyle\sum_{i=1}^n w_i(y_i - u_i)^2 \quad \text{s.t.} \quad L \leq u_1 \leq u_2 \leq \cdots \leq u_n \leq U \tag{10.17}$$

$$\text{ISO}_{1\to n}^{\mathbb{R}}(\mathbf{y}, \mathbf{w}): \quad \min_{\mathbf{u}} \textstyle\sum_{i=1}^n w_i(y_i - u_i)^2 \quad \text{s.t.} \quad u_1 \leq u_2 \leq \cdots \leq u_n \tag{10.18}$$

[5] For subsection 10.3 only, $U \in \mathbb{R}$ is the upper bound in problem (10.17). In the rest of the chapter, U is the cellpath-route incidence matrix.

where we use the notation $\mathrm{ISO}^I_{s\to t}(\mathbf{y},\mathbf{w})$ such that subscript $s\to t$ means we only consider data points with indices from s to t, and superscript I is the interval in which the variables $u_s, u_{s+1}, \cdots, u_t$ lie. Since both problems are strongly convex, they both have a unique solution. The solution to (10.18), denoted $\mathbf{u}^{\mathrm{iso}}$, can be computed in linear time using the *Pool Adjacent Violators* (PAV) algorithm [21, §3], so one hopes that the solution to (10.17), denoted $\mathbf{u}^\star$, derives easily from $\mathbf{u}^{\mathrm{iso}}$. We first give the following lemma:

Lemma 10.1. *Given $\boldsymbol{u}^{iso}$ the solution to (10.18), if there exists k such that $u^{iso}_k < u^{iso}_{k+1}$ then (10.18) reduces to two subproblems:*

$$\begin{array}{ll} ISO^{\mathbb{R}}_{1\to k}(\boldsymbol{y},\boldsymbol{w}): & \min\limits_{\boldsymbol{u}} \sum_{i=1}^k w_i(y_i-u_i)^2 \quad s.t. \quad u_1\le\cdots\le u_k \\ ISO^{\mathbb{R}}_{k+1\to n}(\boldsymbol{y},\boldsymbol{w}): & \min\limits_{\boldsymbol{u}} \sum_{i=k+1}^n w_i(y_i-u_i)^2 \quad s.t. \quad u_{k+1}\le\cdots\le u_n \end{array} \tag{10.19}$$

such that $[u^{iso}_1,\cdots,u^{iso}_k]$ is the solution to the first one and $[u^{iso}_{k+1},\cdots,u^{iso}_n]$ is the solution to the second one. The same result holds for (10.17) and $\boldsymbol{u}^\star$, with resulting subproblems $ISO^{[L,+\infty)}_{1\to k}(\boldsymbol{y},\boldsymbol{w})$ and $ISO^{(-\infty,U]}_{k+1\to n}(\boldsymbol{y},\boldsymbol{w})$.

Proof. Since the constraint $u_k\le u_{k+1}$ is not active at $\mathbf{u}^{\mathrm{iso}}$, it may be removed from (10.18) without altering the solution. Then the resulting program can be separated into the two programs in (10.19) with respective solutions $[u^{\mathrm{iso}}_1,\cdots,u^{\mathrm{iso}}_k]$ and $[u^{\mathrm{iso}}_{k+1},\cdots,u^{\mathrm{iso}}_n]$. □

We now prove the following result:

Proposition 10.2. *The solution $\boldsymbol{u}^\star$ to (10.17) is the Euclidean projection of the solution $\boldsymbol{u}^{iso}$ to (10.18) onto $[L,U]^n$.*

Proof. We start with two simple cases.

Case 1: $[u^{\mathrm{iso}}_i\le L,\,\forall\, i]$. Suppose $\exists\, k$, $u^\star_k > L$. We choose k the smallest of such indices, then either $k=1$ or $L=u_{k-1}<u_k$. In both cases, $[u^\star_k,\cdots,u^\star_n]$ is the unique solution to $\mathrm{ISO}^{(-\infty,U]}_{k\to n}(\mathbf{y},\mathbf{w})$ from Lemma 1. Since $[u^{\mathrm{iso}}_k,\cdots,u^{\mathrm{iso}}_n]$ is also feasible for $\mathrm{ISO}^{(-\infty,U]}_{k\to n}(\mathbf{y},\mathbf{w})$, we have $\sum_{i=k}^n w_i(y_i-u^{\mathrm{iso}}_i)^2 > \sum_{i=k}^n w_i(y_i-u^\star_i)^2$, and adding $\sum_{i=1}^{k-1} w_i(y_i-u^{\mathrm{iso}}_i)^2$ on both sides yields $\sum_{i=1}^n w_i(y_i-u^{\mathrm{iso}}_i)^2 > \sum_{i=1}^{k-1} w_i(y_i-u^{\mathrm{iso}}_i)^2+\sum_{i=k}^n w_i(y_i-u^\star_i)^2$. Since $[u^{\mathrm{iso}}_1,\cdots,u^{\mathrm{iso}}_{k-1},u^\star_k,\cdots,u^\star_n]$ is also feasible for (10.18) ($u^{\mathrm{iso}}_{k-1}\le l<u^\star_k$), this contradicts the optimality of $\mathbf{u}^{\mathrm{iso}}$. Hence $u^\star_k=L,\,\forall\, k$, *i.e.* $\mathbf{u}^\star=\Pi_{[L,U]^n}(\mathbf{u}^{\mathrm{iso}})$.

Case 2: $[u^{\mathrm{iso}}_i\ge U,\,\forall\, i]$. The analysis is similar to case 2. We have: $u^\star_k=U,\,\forall\, k$, *i.e.* $\mathbf{x}^\star=\Pi_{[L,U]^n}(\mathbf{u}^{\mathrm{iso}})$.

General case: Without loss of generality, we suppose there exist two indices s,t such that: $u^{\mathrm{iso}}_1\le\cdots\le u^{\mathrm{iso}}_s\le L<u^{\mathrm{iso}}_{s+1}\le\cdots\le u^{\mathrm{iso}}_{t-1}<U\le x^{\mathrm{iso}}_t\le\cdots\le x^{\mathrm{iso}}_n$. From Lemma 1, $[u^{\mathrm{iso}}_1,\cdots,u^{\mathrm{iso}}_s]$, $[u^{\mathrm{iso}}_{s+1},\cdots,u^{\mathrm{iso}}_{t-1}]$, and $[u^{\mathrm{iso}}_t,\cdots,u^{\mathrm{iso}}_n]$ are then solutions to $\mathrm{ISO}^{\mathbb{R}}_{1\to s}(\mathbf{y},\mathbf{w})$, $\mathrm{ISO}^{\mathbb{R}}_{s+1\to t-1}(y,w)$, and $\mathrm{ISO}^{\mathbb{R}}_{t\to n}(\mathbf{y},\mathbf{w})$ respectively. From case 1, the vector $[L,\cdots,L]\in\mathbb{R}^s$

is solution to $\mathrm{ISO}_{1\to s}^{[L,+\infty)}(\mathbf{y},\mathbf{w})$ and from case 2, the vector $[U,\cdots,U]\in\mathbb{R}^{n-t+1}$ is solution to $\mathrm{ISO}_{t\to n}^{(-\infty,U]}(\mathbf{y},\mathbf{w})$. Then the global vector $\mathbf{x}^*:=[L,\cdots,L,\,u_{s+1}^{\mathrm{iso}},\cdots,u_{t-1}^{\mathrm{iso}},\,U,\cdots,U]$ is the solution to the global program:

$$\begin{array}{ll}\min\limits_{\mathbf{u}} & \sum_{i=1}^n w_i(y_i-u_i)^2\\ \text{s.t.} & L\le u_1\le\cdots\le u_s,\quad u_{s+1}\le\cdots\le u_{t-1},\quad u_t\le\cdots\le u_n\le U\end{array}\tag{10.20}$$

Adding the constraints $u_s\le u_{s+1}$ and $u_{t-1}\le u_t$ to (10.20) does not alter the solution since they are inactive. Hence $[L,\cdots,L,\,u_{s+1}^{\mathrm{iso}},\cdots,u_{t-1}^{\mathrm{iso}},\,U,\cdots,U]$ is the solution to (10.17), *i.e.* $\mathbf{u}^\star=\Pi_{[L,U]^n}(\mathbf{u}^{\mathrm{iso}})$. □

Although isotonic regression is generally studied in the form (10.18), the bounded version (10.17) has appeared in [74]. The simple connection presented in Proposition 10.2 is new to the best of our knowledge. This result can be written $\mathbf{u}^\star=\Pi_{[L,U]^n}(\mathbf{u}^{\mathrm{iso}})$ where $\Pi_{\mathcal{K}}$ is the Euclidean projector onto space $\mathcal{K}$. When $\mathcal{K}=[L,U]^n$, the projected vector $\mathbf{p}:=\Pi_{[L,U]^n}(\mathbf{u})$ is obtained from $\mathbf{u}\in\mathbb{R}^n$ by simply projecting each entry u_i onto $[L,U]$, *i.e.* $p_i=u_i$ if $u_i\in[L,U]$, $p_i=L$ if $x_i<L$, and $p_i=U$ if $x_i>U$. We first give a lemma.

We now give an efficient algorithm to perform the projections (10.16) in line 3 of Algorithm 7.2:

Algorithm 10.2 `PAV-proj(`$\mathbf{y}^p$`)` Projection onto ordering constraints in line 2 of Algorithm 7.2

Require: vector $\mathbf{y}^p\in\mathbb{R}^{n_p-1}$

1. compute $\mathbf{y}^{p,\mathrm{iso}}:=\operatorname{argmin}_{\mathbf{u}^p}\{\|\mathbf{u}^p-\mathbf{y}^p\|_2^2\,:\,u_1^p\le u_2^p\le\cdots\le u_{n_p-1}^p\}$ with the PAV algorithm [21]
2. project $\mathbf{y}^{p,\mathrm{iso}}$ onto $[0,1]^{n_p-1}$: $\tilde{y}_k^p=y_k^{p,\mathrm{iso}}$ if $y_k^{p,\mathrm{iso}}\in[0,1]$; $\tilde{y}_k^p=0$ if $y_k^{p,\mathrm{iso}}\le 0$; $\tilde{y}_k^p=1$ if $y_k^{p,\mathrm{iso}}\ge 1$.
3. **return** $\tilde{\mathbf{y}}^p$

We note that without the constraint elimination described earlier, a projected descent method applied to (10.4) would require q projections onto the probability simplices $\{\tilde{\mathbf{x}}^p\in\mathbb{R}^{n_p}\,|\,\mathbf{1}^T\tilde{\mathbf{x}}^p=1,\,\tilde{\mathbf{x}}^p\succeq 1\}$ at each iteration. The complexity of these projections is $\mathrm{O}\big(n_p\log n_p\big)$ [56, 162], which is less attractive than the $\mathrm{O}\big(n_p\big)$ complexity of Algorithm 10.3.

Problems (10.4) and (10.6) are both convex, and can be solved efficiently with including interior point methods, augmented Lagrangian, gradient projection, and conjugate gradient. In particular, we choose the *Barzilai and Borwein* (BB) method for the accelerated gradient method, where $\mathbf{z}$ is the current iterate and $\mathbf{z}^-$ and previous iterate:

$$\Delta\mathbf{z}=-((\mathbf{y}^T\mathbf{s})/(\mathbf{y}^T\mathbf{y})).\Delta f(\mathbf{z})\quad\text{where}\quad\mathbf{y}=\nabla f(\mathbf{z})-\nabla f(\mathbf{z}^-),\quad\mathbf{s}=\mathbf{z}-\mathbf{z}^-\tag{10.21}$$

The change of variable reduces the dimensionality, at the cost of loosing some of the structure of the traffic assignment problem. While long-standing algorithms such as the

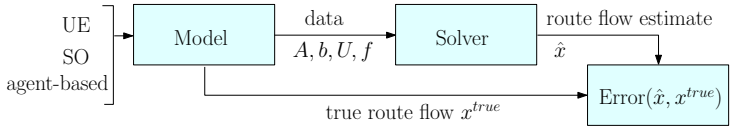

Figure 10.4: Our experiment flow block diagram, where the model is comprised of a network, traffic assignment model, and sensor configuration. The solver is presented in §10.3. The error metric represented here is a function of the estimated and actual route flow. We may compute the percent flow error or, using additional information (e.g. network topology), we may also compute the link flow GEH error.

Frank-Wolfe assignment [101] and the Origin-based assignment [8] and their modifications may have diminished efficiency since the all-or-nothing assignment step is no longer available, their slow convergence is known [125, §11.2.3.1]. We suggest that the estimation problem (10.4) and the traffic assignment problem (10.6) can be reduced to the form (10.12), and then be solved efficiently with quasi-Newton methods (*e.g.* L-BFGS [124]), accelerated gradient methods, or alternating direction methods. These algorithms are proven to have fast convergence, and the proposed projection step is efficient as discussed above. Due to space limitations, early numerical results on the speed up of the algorithms are not shown in the present chapter.

10.4 Experimental setting and validation process

We demonstrate our approach by providing numerical results on experimental networks of varying sizes, using different traffic assignment models and sensor configurations, on the I-210 highway corridor in Los Angeles. To demonstrate the versatility to the underlying experimental model, we investigate the following three scenarios (see Figure 10.4):

1. *Highway network in user equilibrium* (UE), with varying cellpath densities and static sensor coverage.

2. *Highway network in system optimum* (SO), with varying cellpath densities and varying static sensor coverage.

3. *Activity-based agent model on full network*, with varying cellpath densities, 5% static sensor coverage.

Note that we have chosen on purpose three different models (UE, SO, agent-based) to demonstrate the versatility of the method, which is model agnostic and is a major advantage of the approach. Thus, we study networks of different sizes and complexities, different driver behavior models, and trade-offs for different sensor placements. We additionally present preliminary investigation on the effect of measurement and model error on the accuracy of the approach.

Sensor configurations

We have two main types of data: static link sensors data (loop based) and cellpath sensors (cell based). We consider static link sensors on a subset of the links in the network (ranging from 5% to 100% coverage). For the Highway network with UE/SO flow, the subsets of links are chosen such that the *most congested* links are observed, *i.e.* links with highest traffic volumes or flows, whereas in the full large-scale network, we use locations of real highway (PeMS [43]) and arterial loop sensors where the coverage is 5%. For greater coverage on the full network, we randomly sample static sensors along roads in the network.

Although the use of real cellular network data from a service provider would demonstrate even greater applicability of our framework, its availability is restricted for privacy issues, and designing simulators would still be necessary for the flexibility and availability of ground truth data. Our team at the present time is not able to share findings based on collaborations with companies such as AT&T. Our model for cell placement is based on employee population density and locations of major roads. Most notably, many ordinances prohibit towers in residential areas but promote towers in industrial and commercial centers. For both networks, the locations $(X_i, Y_i) \in \mathbb{R}^2$ of the cell towers are randomly sampled on the plane such that the distribution models realistically represent the cellular network.The overall sensor configuration (10.22,10.23,10.24) consists of $N = N^B + N^S + N^L$ total cell towers, where N^B, N^S, N^L are specified by the user and the weights of the multinomial distributions are determined by demographics and geometry. Our sensor configurations are drawn from three distribution models:[6]

1. Within the whole region delimited by a *Bounding box*, N^B cell tower locations (X_1^B, Y_1^B), (X_2^B, Y_2^B), $\cdots$, $(X_{N^B}^B, Y_{N^B}^B)$ are sampled uniformly (10.22).

2. The whole region is comprised of sub-regions $\mathcal{S}$. Within each sub-region s, delimited by a rectangle $(X_{\min}^s, Y_{\min}^s)$, $(X_{\max}^s, Y_{\max}^s)$, N^s more cell tower locations $\{(X_i^s, Y_i^s)\}_{i=1,\cdots,N^s}$ are sampled (10.23). The number of base stations N^s within each sub-region $s \in \mathcal{S}$ is sampled from a multinomial distribution with N^S trials, where N^S is the total number of cell towers among all the sub-regions (excluding those sampled in the previous step from the whole region) and weights proportional to demographic information for each region (e.g. employee population).

3. The network within the region contains $\mathcal{E}$ major edges (that is, those likely to have cell towers nearby, e.g. highways). Along each edge e (also called *arcs*), N^e cell tower locations are sampled uniformly along the link with a Gaussian noise (10.24) where (X_s^e, Y_s^e) is the location of the start of link e, and (X_t^e, Y_t^e) is the location of the end of link e. The numbers of base stations N^e along links $e \in \mathcal{E}$ are sampled from a multinomial distribution with N^L trials, where N^L is the total number of cells along edges (specified by the user) and weights proportional to the length of e.

[6]Implementation is open source and available at `https://github.com/cathywu/synthetic-traffic`.

$$\text{Bounding box}: \quad X_i^B \sim U([X_{\min}^B, X_{\max}^B]),\, Y_i^B \sim U([Y_{\min}^B, Y_{\max}^B]), \quad \text{for} \quad i = 1, \cdots, N^B \tag{10.22}$$

$$\text{Sub-region } S: \quad X_i^s \sim U([X_{\min}^s, X_{\max}^s]),\, Y_i^s \sim U([Y_{\min}^s, Y_{\max}^s]), \quad \text{for} \quad i = 1, \cdots, N^s \tag{10.23}$$

$$\text{Link } a: \quad \begin{cases} X_i^a \sim X_s^a + t_i(X_t^a - X_s^a) + N(0, \sigma) \\ Y_i^a \sim Y_s^a + t_i(Y_t^a - Y_s^a) + N(0, \sigma) \end{cases} \quad \text{such that} \quad t_i \sim U([0, 1]), \tag{10.24}$$

$$\text{for} \quad i = 1, \cdots, N^a \tag{10.25}$$

Scenarios 1 and 2: UE and SO on the highway network

We consider first the *highway network* of the region.[7] The roads are extracted from OpenStreetMaps (OSM) and we only keep the ones with five lanes or more and up to 11 lanes. This results in a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = 44$ nodes and $|\mathcal{E}| = 122$ directed links. We obtain the free flow delay d_e on each link $e \in \mathcal{E}$ as the link's length divided by the link's free speed, as defined by OSM, and cross-check the values with the delays given by Google Maps. An illustration of the network is provided in Fig. 10.5.

The OD demands are based on census data and employment concentration in L.A. county, which are extracted from the Census Bureau. The OD demands model is simplified to a static morning rush hour model[8] of the region such that: i) only 21 origins have positive flows emanating from them; ii) all the trips terminate at three destinations: near Burbank at node 5, towards Santa Monica at node 20, and in Downtown L.A. at node 22; iii) we only have 42 OD pairs with positive flows ranging from 1200 veh/hour to 12,000 veh/hour.[9]

For our equilibrium-based approach, we consider the traffic assignment model presented in [143, §3.1] to generate route flows and cellpath flows. Specifically, the travel time on a given edge e is a strictly increasing function $c_e(\cdot)$ of the traffic volume (flow) v_e on that link only. We choose the congestion performance estimated by the Bureau of Public Roads, where d_e is the free flow delay and m_e the number of lanes on edge e, and provide the Beckmann objective function ϕ^{UE} associated to the overall model [13]):

$$\text{link delay:} \quad c_e(x_e) = d_e(1 + 0.15(x_e/m_e)^4),\ \forall\, e \in \mathcal{E} \tag{10.26}$$

$$\text{UE potential:} \quad \phi^{\text{UE}}(\mathbf{x}) = \sum_{e \in \mathcal{E}} \int_0^{x_e} c_e(u)du \tag{10.27}$$

[7] The region has bounding box [-118.328299, 33.984601, -117.68132, 34.255881] in latitude longitude.

[8] Based on observed flows on 2014-06-12 at 9:14 AM from Google Maps.

[9] The script and the Python's class to construct the Highway network are online: `https://github.com/jeromethai/traffic-estimation-wardrop`.

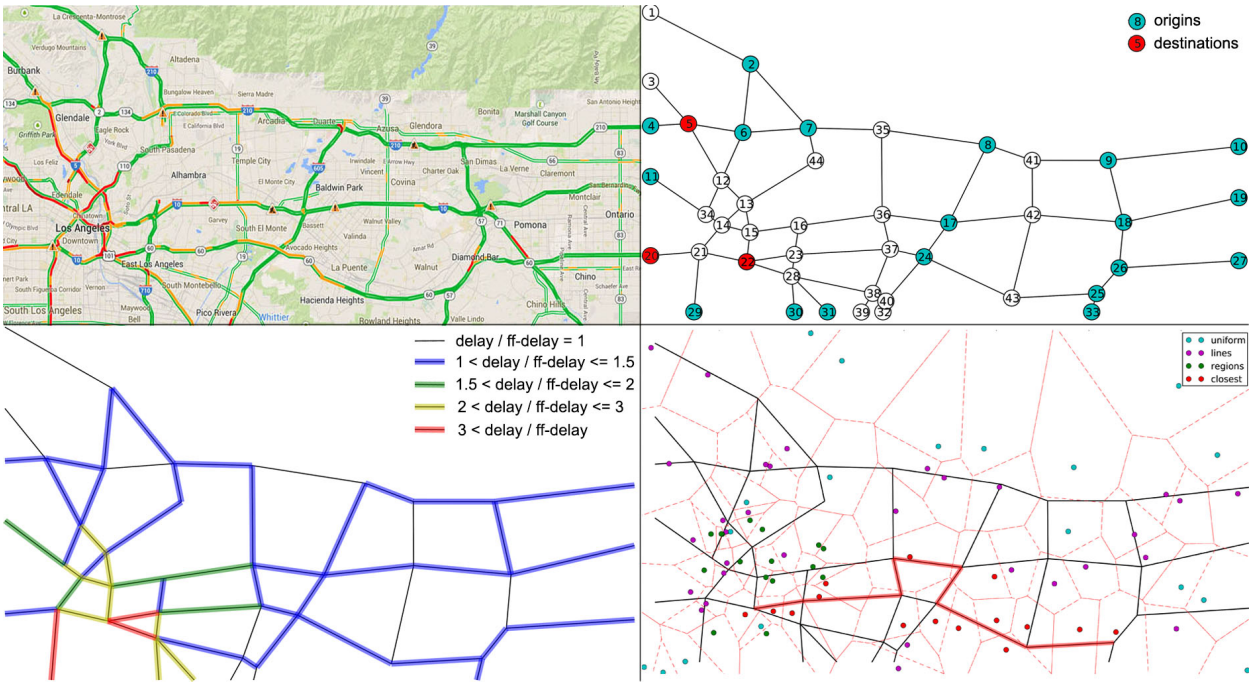

Figure 10.5: Benchmark (small-scale) example used for the first numerical run: The four subfigures present the Highway network of the I-210 highway corridor in L.A. county. Starting from the top left and in clockwise order: 1) the state of traffic on 2014-06-12 at 9:14 AM from Google Maps; 2) the nodes in blue and red are nodes from which positive flows emanate, nodes in red are nodes from which positive flows terminate; 3) network with 80 sampled cells, with a higher concentration of cells near downtown. A random path from 25 to 22 is shown in red with the closest cell towers. 4) The highway network in User Equilibrium with the resulting delays.

In our equilibrium model, the vertices are indexed by $v \in \mathcal{V}$, the 42 OD pairs are indexed by $k \in \{1, \cdots, Q\}$, $\mathbf{A}^{\text{full}} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{R}|}$ is the link-route incidence matrix, $N \in \{-1, 0, 1\}^{|\mathcal{V}|\times|\mathcal{E}|}$ is the node-link incidence matrix, and let $\mathbf{d}^k \in \mathbb{R}^{|\mathcal{V}|}$ be the vector associated to OD pair $k = (k_s, k_t)$ such that $d_i^k = -d_k$ at node $i = k_s$ (the origin), $d_i^k = d_k$ at node $i = k_t$ (the destination), and $d_i^k = 0$ otherwise. Under the assumptions of our experiment, the path-flow traffic assignment (PTA) is equivalent to the link-flow traffic assignment (LTA), *i.e.* they give the same *unique* link flow solution [67]:

$$\text{PTA}: \min \phi^{\text{UE}}(\mathbf{A}^{\text{full}}\mathbf{x}) \quad \text{s.t.} \quad \mathbf{T}\mathbf{x} = \mathbf{d},\, \mathbf{x} \succeq 0 \tag{10.28}$$

$$\text{LTA}: \min \phi^{\text{UE}}(\mathbf{v}) \quad \text{s.t.} \quad \mathbf{v} \in \mathcal{K} \tag{10.29}$$

where the feasible set for the (LTA) is:

$$\mathcal{K} := \left\{ \mathbf{v} \in \mathbb{R}_+^{|\mathcal{E}|} \mid \exists\, \mathbf{v}^k \in \mathbb{R}_+^{|\mathcal{E}|},\, \mathbf{v} = \sum_{k=1}^{Q} \mathbf{v}^k,\, \mathbf{N}\mathbf{v}^k = \mathbf{d}^k,\, \forall\, k \in \{1, \cdots, Q\} \right\} \tag{10.30}$$

Since PTA is not tractable due to the computational cost of enumerating all the possible routes, we solve LTA in the first step, then perform the following steps to generate a set of routes $\mathcal{R}$ with an associated UE route flow vector $\mathbf{x}^{\mathrm{UE}} \in \mathbb{R}_+^{|\mathcal{R}|}$, and a set $\mathcal{P}$ of cellpaths with an associated UE cellpath flow vector $\mathbf{f}^{\mathrm{UE}} \in \mathbb{R}_+^{|\mathcal{P}|}$:

1. We solve LTA and obtain the UE link flow $\mathbf{v}^{\mathrm{UE}} \in \mathbb{R}_+^{|\mathcal{E}|}$ and resulting link delays.

2. We find the K-shortest paths under the UE delays for each of the 42 OD pairs, using Yen's algorithm [173]. Note that K is chosen large enough such that *at least* all used routes are extracted, *i.e.* all the routes with the same shortest delays as characterized by Wardrop equilibrium. We choose $K = 5$ and extract 275 candidate routes.

3. We solve PTA with the 275 *candidate routes* starting from a random initial point. Let $\mathbf{x}^{\mathrm{UE}}$ be a route flow solution (the resulting link flow $\mathbf{A}^{\mathrm{full}}\mathbf{x}^{\mathrm{UE}}$ should be equal to $\mathbf{v}^{\mathrm{UE}}$ since the UE link flow is unique).

4. We sample cellpaths on the highway network following the model presented in §10.4 (see Fig. 10.5).

5. For each of the 275 routes with a positive flow on it – we found $\mathrm{card}\,\{r \mid x_r^{\mathrm{UE}} > 0\} = 91$ *used routes* – we compute the sequence of cells that intersect with it. The cellpath flows are given by: $f_p^{\mathrm{UE}} = \sum_{r \in \mathcal{R}^p} x_r^p$.

On a network with SO flow, the average delay is minimized [165, 92], hence the potential function of be minimized is ϕ^{SO} in (10.31) subject to the constraints in (10.28) for the path-flow formulation, and (10.29) for the link-flow formulation. In fact, the SO link flow corresponds to the UE link flow with the modified delay function $\tilde{c}_e(\cdot)$ in (10.31), called the marginal delay function [138] (where the prime indicates the derivative function):

$$\text{link marginal delay}: \quad \tilde{c}_e = c_e(x_e) + x_e c_e'(x_e)\ ; \qquad \text{SO potential:} \quad \phi^{\mathrm{SO}}(\mathbf{v}) = \sum_{e \in \mathcal{E}} v_e c_e(v_e) \tag{10.31}$$

Steps 1 to 5 are performed with the SO potential ϕ^{SO} to generate a SO route flow $\boldsymbol{\mu}^{\mathrm{SO}}$ and a SO cellpath flow $\mathbf{f}^{\mathrm{SO}}$ on the Highway network described above, with a few minor differences:

- In step 2, we find the K-shortest paths under the marginal delays induced by the SO link flow. We choose $K = 7$ and we extract 300 candidate routes.

- In step 5, we found 153 routes with positive flow on it.

Figure 10.6: Full-scale network including highway and arterial networks of the I-210 corridor used for MATSim data generation, and for the estimation problem. See Figure 10.2 for the Voronoi tessellation model of the cellular network and the 700 origins given by the TAZ.

Scenario 3: activity-based agent model on the large-scale full network

We additionally consider a large *full network*, comprising of both the highway network and the arterial networks in the region. We use the OpenStreetMaps network of the greater Los Angeles area, excluding residential links. Our network comprises of 20,513 edges (links) and 10,538 nodes (intersections). We take the origins to be the Traffic Analysis Zones (TAZ) given by the US census, of which there are 700 in the region (see Fig. 10.6).

On this large-scale network, we consider an *activity-based agent model*. MATSim is a well-known open-source traffic simulation framework [84], which searches for a user equilibrium in terms of utility functions defined for the agents using a co-evolutionary optimization algorithm. In our setting, we consider agent utility as a function of travel time. MATSim differs from the user equilibrium model above in that it is only quasi-static, by varying slightly the departure times for every agent. MATSim is suitable for performing large-scale agent simulations. We simulate the morning and evening rush-hours using 500,000 agents, as those are the most vital times to understand the state of traffic. The home and work locations for each agent are distributed randomly via census demographics. Since MATSim randomly selects starting and ending points (within origins and destination) as opposed to using the region centroids, typically all of the trajectories it generates are unique. Selecting all of the

trajectories to be our possible routes lends itself to be a trivial problem in our formulation. Instead, we examine trajectories between each OD pair and group them by similarity as follows: 1) Find the trajectory which matches with the most other trajectories ($\geq$80% match in length). Add this trajectory to the list of routes for the OD pair; 2) Remove all trajectories that match with this route and repeat. Stop when 50 routes are selected or when there are no more trajectories. 50 routes empirically accounts for 99.4% of the 500K trajectories. In this scenario, we consider coupled OD and cellpath flow information as provided by MATSim, which we denote as *OD + cellpath flow*, for estimating route flow. In a real setting, this information may be inferred by a trip analysis method applied to cellular network data.

Implementation

The software to run the experiments was developed mostly in `python 2.7`, using the `GEOS (v.3.4.2)` library for geometric computations. All data is managed and stored in a `PostGIS 2.1.3` database. The geometries and other data about routes, cell tower Voronoi tesselations, and the links of the road network are all stored in the database with spatial indices on all geometry columns, allowing `PostGIS` spatial queries to be performed efficiently for extracting cell path information associated with each route. The data for the I-210 corridor contains 280,691 routes, 700 origins, 1033 sensors, and was tested with numerous different numbers of cells, ranging from 200 to 4000.

The incidence matrices A and U (roughly 250K-by-300K matrices) are generated by finding the cellpath and OD pair for each route from the database by ordering the set of Voronoi cells that intersected with the respective route. The link-route incidence matrix is formed by finding all routes whose distance from the sensor locations was less than some threshold empirically selected such that the maps matched well ($\approx$ 10 meters tolerance for the PeMS geometries). All incidence matrices are saved in the `scipy.sparse` format. The convex optimization program[10] was developed in Python, using `scipy.sparse` and `numpy` for matrix computation. The PAV projection algorithm was written in `C`, and bindings were written so that it could be called from the Python optimization algorithm.

10.5 Numerical results

We validate our approach by measuring our accuracy in route flow estimates $\hat{\mathbf{x}}$, where $\hat{\mathbf{x}}$ is a solution of our model (10.1) for different scenarios. Note that the problem being solved is (10.4) following our algorithmic approach, and the solution in $\mathbf{z}$ is converted to $\tilde{\mathbf{x}}$ then $\hat{\mathbf{x}}$ following the simple relation in (10.12) and (10.14). We additionally present our accuracy in terms of link flow estimates, to serve as a comparison to classical approaches to link flow estimation:

[10]Implementation is open source and available at `https://github.com/cathywu/traffic-estimation`.

- Route flow error: $\epsilon_r = \|\mathbf{x}^{true} - \hat{\mathbf{x}}\|_1 / \|\mathbf{x}^{true}\|_1$, with $\mathbf{x}^{true}$ the true route flow and $\hat{\mathbf{x}}$ the estimated route flow. This is the percent error of flow allocation among all routes.

- Link flow error:
  1) For observed links: $\epsilon_l^{obs} = |GEH_i^{obs} < 5, \forall i \in \hat{\mathbf{b}}|/|\mathbf{b}^{true}|$, with $\mathbf{b}^{true} = \mathbf{A}\mathbf{x}^{true}$ true observed link flows, $\hat{\mathbf{b}} = \mathbf{A}\hat{\mathbf{x}}$ estimated observed link flows, and $GEH_i^{obs} = \sqrt{\frac{(b_i^{true} - \hat{b}_i)^2}{0.5(b_i^{true} + \hat{b}_i)}}$ associated GEH measure for each link.
  2) For all links: $\epsilon_l^{full} = |GEH_i^{full} < 5, \forall i \in \hat{\mathbf{v}}|/|\mathbf{v}^{true}|$, with $\mathbf{v}^{true} = \mathbf{A}^{full}\mathbf{x}^{true}$ true full link flows, $\hat{\mathbf{v}} = \mathbf{A}^{full}\hat{\mathbf{x}}$ estimated full link flows, and $GEH_i^{full} = \sqrt{\frac{(v_i^{true} - \hat{v}_i)^2}{0.5(v_i^{true} + \hat{v}_i)}}$ associated GEH measure for each link.
  This is called the GEH statistic, a heuristic formula commonly used to compare two sets of traffic volumes, e.g. for calibration of microsimulation models [55, §5.6] and for validating hourly traffic flows [155, §11-13]. For an individual link, a GEH value of less than 5.0 is considered to be a good match. For a vector of links, a fraction $\epsilon_l \leq 0.85$ of good matches is considered a good match overall between modeled and observed volumes.

Highway network

Using the highway network settings in §10.4, we start with 100% of link coverage and 80 cells such that $N^B = 20$, $N^L = 40$, and $N^S = 20$, where $\mathcal{S}$ contains only 1 region and is roughly downtown Los Angeles (see 10.4). The link coverage is then decreased from 90% to 10% such that we always observe the most congested links, and the number of cells is successively scaled down by a factor 2 such that the proportions between N^B, N^L, N^S are conserved. We analyze how the errors ϵ_r in route flows, vary when sensors are more sparse. Since we choose random initial points in PTA (10.28) and in the solver (10.1) to generate synthetic route flows and compute the estimate respectively, and since the cellular network is sampled randomly, all the results presented in this section have been averaged over 10 trials.[11]

Figure 10.7 presents the numerical results when link flows and OD demands are known, and cellular network data are assimilated into the model. The problem being solved has in fact a different objective from (10.1):[12]

$$\min \frac{1}{2}\|\mathbf{A}'\mathbf{x} - \mathbf{b}'\|_2^2 \quad \text{s.t.} \quad \mathbf{U}\mathbf{x} = \mathbf{f},\ \mathbf{x} \succeq 0 \qquad \text{where} \quad \mathbf{A}' = \begin{bmatrix}\mathbf{A}\\ \mathbf{T}\end{bmatrix} \quad \text{and} \quad \mathbf{b}' = \begin{bmatrix}\mathbf{b}\\ \mathbf{d}\end{bmatrix} \tag{10.32}$$

In UE, the presence of cell phone data in addition to OD demands reduces ϵ_r by at least a factor 10 when there are 5 to 40 cells and less than 60% of links observed (Fig. 10.7, top

[11] The code was fully implemented in Python and is available on github: `https://github.com/jeromethai/traffic-estimation-wardrop`.

[12] Since the inequalities $Ux = f$, $Tx = d$, $x \succeq 0$ might not define simplexes, we chose formulation (10.32) over: $\min \frac{1}{2}\|Ax - b\|_2^2$ s.t. $Ux = f$, $Tx = d$, $x \succeq 0$ to have the same constraints as in (10.1) for our algorithmic approach. Besides, with dense cellular networks, satisfying $Tx = d$ is redundant with the constraints $Ux = f$ because OD demands are included in cellular network data, hence both formulations reduce to (10.1).

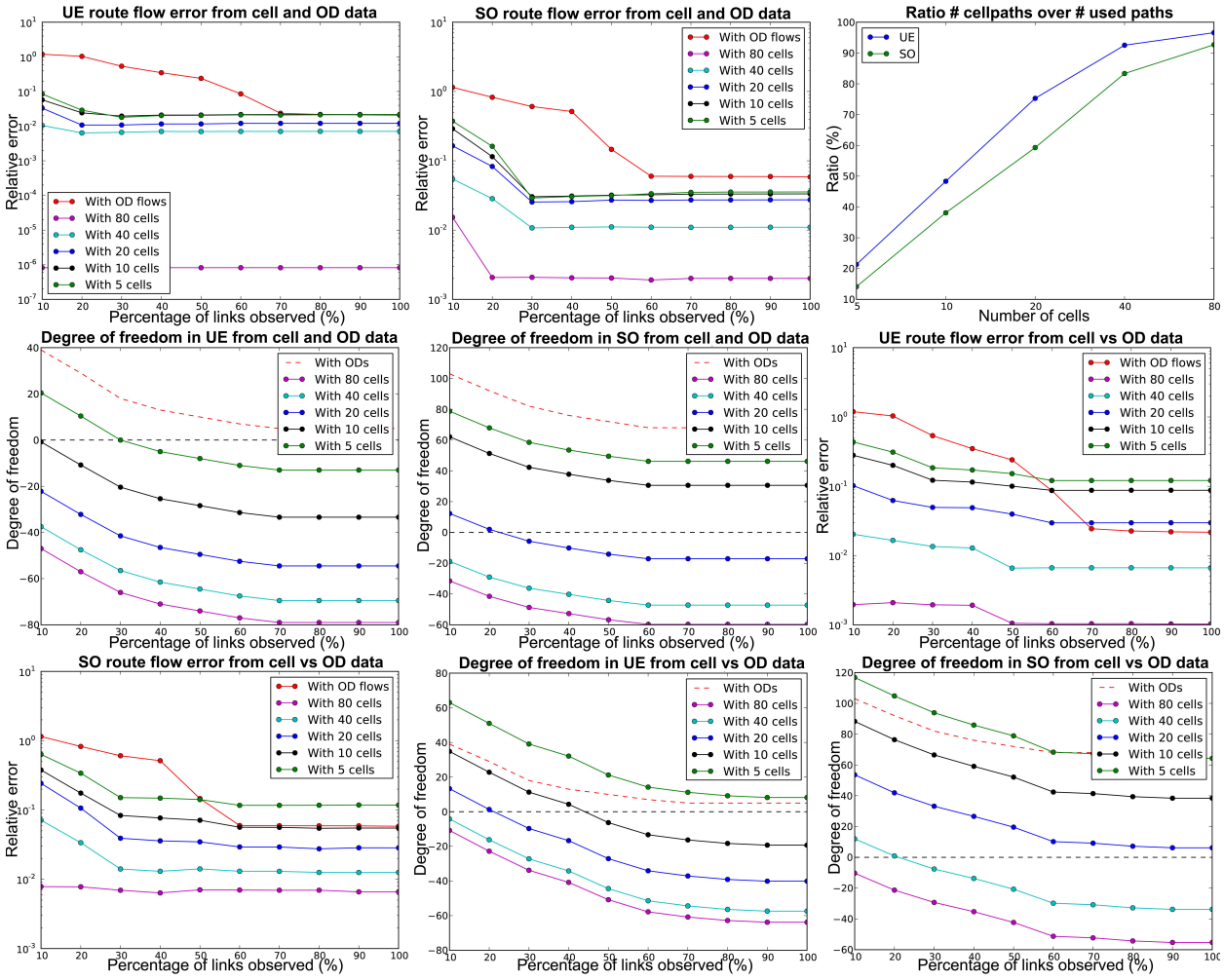

Figure 10.7: The nine subfigures present the numerical results for the highway network. From the left to right: 1) the route flow error ϵ_r from OD demands (red curve) and OD demands & cellpath flows (other curves) with different link coverage values and different numbers of cells for the network in UE; 2) ϵ_r from OD demands (red curve) and OD demands & cellpath flows (other curves) for different configurations of the network in SO; 3) ratio of the number of observed cellpaths to the number of used paths; 4) lower bound on the degree of freedom for the program with OD demands (red curve) and OD demands & cellpath flows (other curves) for the network in UE; 5) lower bound on the degree of freedom for the program with OD demands (red curve) and OD demands & cellpath flows (other curves) for the network in SO; 6) the route flow error ϵ_r from OD demands (red curve) and cellpath flows only (other curves) with different link coverage values and different numbers of cells for the network in UE; 7) ϵ_r from OD demands (red curve) and cellpath flows only (other curves) for different configurations of the network in SO; 8) lower bound on the degree of freedom for the program with OD demands (red curve) and cellpath flows only (other curves) for the network in UE; 9) lower bound on the degree of freedom for the program with OD demands (red curve) and cellpath flows only (other curves) for the network in SO. Best viewed in color.

left subfigure). In SO, the gain in accuracy becomes significant (by a factor 10) with 5 to 20 cells from 20 to 50% of link coverage (Fig. 10.7, top middle subfigure). Hence, data from relatively sparse networks compensates well for the lack of information from sparse static sensors. With 80 cells in the UE setting and at least 40 cells in the SO setting, ϵ_r from OD flows and cell data is lower than in any other cases. Hence, data from dense cellular networks provides information that even 100% of link coverage would not be able to provide: the flows on routes. This is also illustrated in the top right subfigure of Fig. 10.7: with 40 cells, the ratio of the number of cellpaths observed to the number of routes used is 80% and 90% in SO and UE respectively. This means that we observe the exact flow on at least 60% and 80% of the routes respectively. Among the remaining percentage, only the sum of flows on routes sharing the same cellpath is observed.

The accuracy in the estimates is closely related to the degree of freedom in problem (10.1). When we consider problem (10.32) without inequality constraints, the degree of freedom is given by the dimension $n - \text{rank}[\mathbf{A}^T, \mathbf{T}^T, \mathbf{U}^T]$ where n is the dimension of the problem. With $\mathbf{x} \succeq 0$, the support of the estimated distribution of flows is generally restricted to be on the *used routes* instead of the *candidate routes* (see §10.4), because observing positive flows along used routes from different sensors forces the program to allocate *positive fractions* of flows along these routes, with no quantities left for the other routes. Hence a lower bound and a good estimate of the degree of freedom is given by $|r \,|\, x_r > 0| - \text{rank}[A^T, T^T, U^T]$. When this quantity is negative, we have an overdetermined problem. In the middle left and middle center subfigures of Figure 10.7, we observe that problem (10.32) is underdetermined without the use of cellular network data in UE/SO, and with 10 cells or less in SO. Including cell phone data has also a greater effect on the lower bound on the degree of freedom than increasing the link coverage, which confirms the need of cell phone data to solve the underdetermined-ness in route flow problems.

The last four subfigures of Figure 10.7 present results when link flows are known, the route flow estimation error from cellular network data (without OD demands) is compared to the estimation error from OD demands. We observe that using cellular network data instead of OD demands is beneficial when there is 60% of link coverage or less on the network in UE, and 40% of link coverage or less in SO. For greater percentages of link coverage, cellular network data provides more information on the route flows than 100% of link coverage when there are at least 40 cells in UE, and 20 cells in SO (see Figure 10.7 middle right and bottom left subfigures). When OD demands and cellpath flows are not combined, the problem is underdetermined for 20 cells or less.

Full network, activity-based model

Using the full network setting in §10.4, we perform experiments using the actual locations of PeMS static highway count sensors on 1033 links (about 5% coverage). We use the following baseline sensor configuration model for base stations[13]: $N^B = 100, N^S = 800, N^L = 50$, where

[13]The I-210 region is 688mi^2 and, with cell towers spaced $\frac{1}{4}$ to 2 miles apart for suburban and urban areas, a reasonable range of cell towers for modern urban areas is 180 to 5500. We select 950 for our baseline model,

the sub-regions $\mathcal{S}$ is given by the bounding boxes for the TAZ within the whole region.We analyze how the errors in route flows and link flows vary when the number of base stations vary from 0.25 times to 4 times, with each of the model parameters scaled proportionally.

Figure 10.8 presents the numerical results when select link flows and all OD + cellpath flows are known. To select a particular estimate from the solution space, we add an ℓ_2 regularization term to the objective. In our dataset, selecting the top 50 routes per OD pair was sufficient to account for 99.4% of trajectories; however, in general, the corresponding number of routes needed will vary based on the network, time of day, underlying driver behavior, etc. Thus, we present trade-off curves for varying the number of routes from 3 up to 50. As expected, as more routes are considered, the route flow accuracy ϵ_r declines, since the solution space (and its corresponding nullspace) grows. Fortunately, the accuracy increases with the number of cells. Thus, Figure 10.8 (top left subfigure) shows that the same level of accuracy may be attained when considering different numbers of routes (per OD pair), by varying also the number of cells. Our method performs comparably for the morning (shown in Figure 10.8) and evening (not shown) rush hours, achieving 92.0% and 91.9% route flow accuracy respectively and well exceeding the GEH test (with 950 cells and 50 routes per OD), indicating the versatility of our method for diverse traffic settings. As a short note on link flow, our method achieves link flow $\epsilon_l^{obs} = 1, \epsilon_l^{full} = 1$ for all link volume classes, sensor configurations, and route choices, which is reasonable in the noiseless setting and explained by our objective minimizing the error to the observed link flows.

Similarly to the highway network experiment, the accuracy in the estimates is closely related to the degree of freedom in problem (10.1). For computational reasons, we compute an approximate measure of the degrees of freedom by $\text{nullity}(\mathbf{AN}) \geq |\mathbf{z}| - \text{rank}(\mathbf{A})$, using notation from (10.12). Although the problem remains underdetermined (based on equality constraints in the noiseless setting), the accuracy increases substantially as the degrees of freedom decreases (Figure 10.8, top right). In all scenarios, we note that adding the cellpath flow information (compared to using OD information only) greatly improves the estimates of route and link flows.

However, selecting the top routes between each OD pair relies on sophisticated models and techniques. Though this chapter focuses on the noiseless setting, here we present preliminary results for a noisy setting, motivated by situations where not all top routes may be curated. We call *modeling error* the flow that is not modeled by the curated routes. Figure 10.8 (bottom subfigures) shows an experiment where we consider the performance of our method where we curate the top 3-50 routes (per OD) and evaluate our method in the presence of modeling error. We see that curating 20-50 routes (per OD) is sufficient for achieving a low ($< 10\%$) route flow error. We see also that 20-50 routes is sufficient for performing well on the GEH metric on all links (including those not observed) for various link volume classes. Our results show that using too few routes is unsuitable for route flow estimation (but may still be suitable for link flow estimation) in scenarios where the driving population takes many varied routes, as in our MATSim dataset.

as a reasonable estimate of cell towers in the region, and experiment from 200 to 4000.

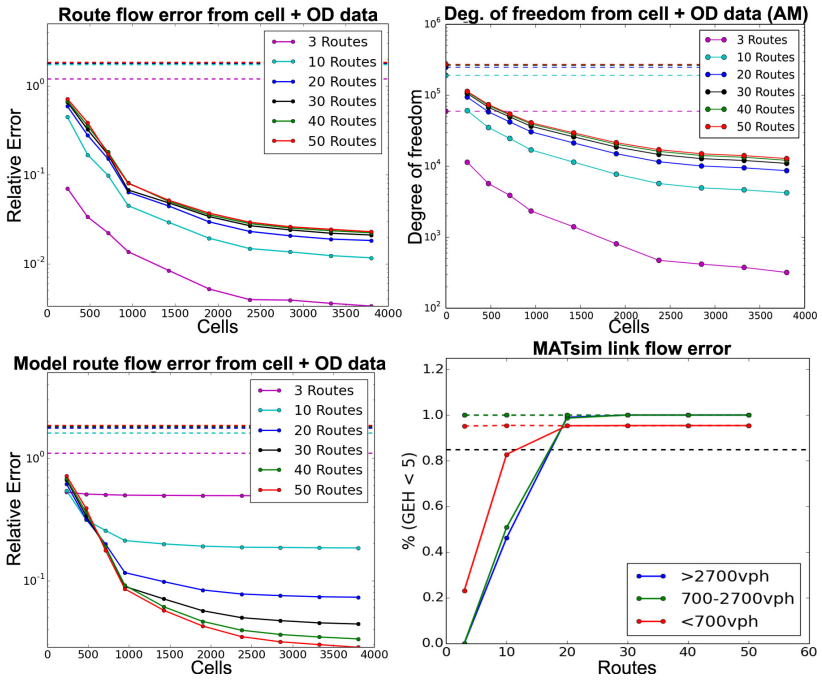

Figure 10.8: Full (highway and arterial) network experiment results, corresponding to the regularized solution for the morning commute (rush hour). The top row corresponds to the noiseless setting; the bottom row corresponds to experiments including modeling error (noise). Top left: Route flow ϵ_r from OD demands (dotted) and OD + cellpath flows (solid) for varying cell counts. The different curves indicate the number of routes (per OD) considered; Top right: Approximate degrees of freedom for the program with OD demands (dotted) and OD + cellpath flows (solid) for varying cell counts. Bottom left: Including modeling error, the route flow ϵ_r from OD demands (dotted line) and OD + cellpath flows (curves) for varying numbers of cells. Bottom right: Link flow error ϵ_l evaluated on observed links b (dotted) and all links v (solid), shown for different link flow volume classes for 950 cells.

10.6 Conclusion

Our work demonstrates a data-driven method that is capable of estimating route-level flow accurately, on a large scale, and is versatile to different vehicle behaviors. We address the traditionally highly underdetermined problem by proposing the concept of *cellpaths* for cellular traces. We design a projected gradient algorithm suitable for the route flow estimation problem, as well as the traffic assignment problem. We validate our approach on several networks of varying sizes and underlying models. Finally, our methodology is shown to be compatible with several other approaches and types of data, which may be used in conjunction for improved estimation.

As route flows contain strictly more information than link flows, which underlie many

transportation methods, the potential for accurate route flow estimates in transportation applications is vast. Additionally, whereas traffic assignment, which models rather than estimates route flows, is critical for long-term land-use planning, their strong model assumptions limit their application in short time-horizon applications. Being a data-driven approach, our method enables new short time-horizon applications for the prediction and control of transportation such as route guidance, re-routing (e.g. minimizing effects of road closures, disasters, large events, etc.), demand prediction, and anomaly detection and analysis. Our framework aims to be widely deployable (wherever there is wide-spread cellular network coverage) and extendable, thereby providing a baseline estimator of the state of our current traffic networks, against which new controls and designs for intelligent transportation systems can compare.

The directions for future work concern with the implementation of the production system for the I-210 corridor in California, US. We plan to analyze and improve the robustness of our model in the presence of measurement error. Real loop sensors are notoriously noisy and a fraction of them are offline at any given point. Since cellpath flow is not measured directly, but rather is inferred from cellular network traces, and so is prone to error from any inference procedure used. Given the achieved computational performance, we plan to extend our work to the dynamic case, where we explore time-varying traffic demands in near real-time. The full pipeline (summarized in Figure 10.1) will be implemented to perform large-scale route flow estimation using cellular network traces from AT&T and actual cell tower locations for the I-210 corridor in California, US.

Chapter 11

Conclusion

In Chapter 2 of our dissertation, we have presented the theoretical foundations of the selfish routing game, and we have shown that the equilibrium flow can be computed as a solution of a convex optimization problem, or a variational inequality problem. In contrast to the selfish routing game with homogeneous players, its heterogeneous extension, in which the cost of traveling is perceived differently among the driving population, described an equilibrium flow that cannot be formulated as a solution of a convex optimization problem. Fortunately, the equilibrium flow in the heterogeneous setting can still be computed by solving a variational inequlity problem. In particular, from classic results in variational inequality theory, see *e.g.*, [142], [61], existence and uniqueness of the equilibrium flow on the network is guaranteed if the cost functions are continuous and sctrictly increasing.

In Chapter 3, we describe the Frank-Wolfe algorithm (a.k.a. the conditional gradient algorithm), which is a popular algorithm for solving the traffic assignment problem. Specifically, the Frank-Wolfe algorithm is an iterative descent method in which each iteration finds a search direction going toward a smaller objective function value at a best step length. On the theoretical side, we provide convergence rates on the Frank-Wolfe algorithm that generalize the result of Jaggi in [87]. On the computational side, we show that the Frank-Wolfe algorithm enables to leverage the sparsity structure of the traffic assignment problem by reducing the problem of computing the search direction to determining shortest-paths between all Origin-Destiation pairs based on travel costs at the current iteration. Even though this enables to compute the search direction very efficiently using, *e.g.*, Dijkstra's algorithm, we have observed that this computation still accounts for more than 95% of the overall execution time. of the overall execution time. As an extension of our work on the Frank-Wolfe algorithm, which is available on GitHub (`github.com/megacell/python-traffic-assignment`0, we have collaborated with Juliette Ugirumurera,[1] on a High Performance Computing (HPC) that is not covered in the present dissertation. Specifically, we incorporated a parallel shortest-path algorithm into the Frank-Wolfe algorithm applied to the Traffic Assignment problem. We implemented the parallel Frank-Wolfe algorithm on the Edison supercomputer at NERSC

[1] Juliette Ugirumurera is a postdoctoral research fellow in the Scalable Solvers Group of the Computational Research Division at the Lawrence Berkeley National Lab

(nersc.gov). Our initial parallelization duplicated the network on 5 compute nodes, and equally divided the O-D pairs among 120 cores (24 cores per node). The 120 cores computed the shortest-paths for their assigned O-D pairs simultaneously. We tested this algorithm using the Los Angeles network, which had 12,982 nodes 39,018 links and 1,360,427 O-D pairs. The computation time is reduced by a factor of 25 compared to the sequential Frank-Wolfe algorithm.

In Chapter 4, we showed how the selfish routing model provides a game-theoretical framework that can be used to study the impact of the increasing penetration of routing apps on road usage. Our numerical simulations show that app-based routing can potentially increase the vehicle miles traveled (VMT) on local roads by .34 million miles per hour, which represents a three-fold increase in traffic on low-capacity links, while there is only a 10% decrease in VMT on high-capacity roads. Despite a general decrease in VMT due to more efficient routing, the relative increase on low-capacity roads is very important for each 10% increase in routed users, due to the small traffic flow on the low-capacity network. This causes residential streets to be congested, encouraging cities to spend millions in infrastructure to steer the traffic away. As an extension of this preliminary work, Théophile Cabannes led a team of researchers that seeked to empirically validate the recent rise in "cut-through" traffic due to the use of GPS-enabled routing. Specifically, they use INRIX speed data on a specific day in LA to show that travel times on arterial roads and the I-210 are equalized during peak hours between Pasadena and Azusa. They also show that arterial road detours can be as much as 20% faster than the corresponding I-210 route. In addition, they use PeMS data (2013 to 2017) and INRIX data (2014 to 2015), to show that an increasing number of drivers might be using shortcuts, leading to a three-fold flow increase on some off ramps over four years and a 14% decrease in speed on some arterial roads over one year.

In Chapters 5 and 6, we noted that the use of the selfish routing game by urban planners to evaluate projects heavily relies upon the assumption that the edge cost functions yield equilibrium flows that are representative of the actual flows on the urban network. Hence, we presented a framework that enables to study the prediction accuracy of the selfish routing model, when it is chosen to fit the empirical data. Specifically, we study the selfish routing game seen as a regression model encoding the relationship between the traffic demand (inputs of our model) and the resulting equilibrium flow (outputs). We assume that the vector of edge cost functions $F = (c_e(\cdot))_{e\in\mathcal{E}}$ belongs to an indexed-family $\{F_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$, and the empirical risk minimization principle consists in choosing the parameter that gives the lowest empirical risk $R_N(\boldsymbol{\theta})$, which is defined as the loss under the empirical measure defined by N samples of inputs and outputs. It is then critical to know if the empirical risk, which is obtained for free from the empirical risk minimization principle, is a good estimate of the population risk $R(\boldsymbol{\theta})$, where the population risk is the expected loss if we were to sample a new predictor. The population risk is thus a measure of how good, on average, our model is at predicting the output given a new input. This is in fact the ultimate measure of the quality of the model. To understand if the empirical risk is a good estimation of the population risk, we studied the behavior of the uniform deviation $\|\mathbb{P} - \mathbb{P}_N\|_{\mathcal{L}} := \sup_{\boldsymbol{\theta}\in\Theta} \|R(\boldsymbol{\theta}) - R_N(\boldsymbol{\theta})\|$ between the empirical and population

risks. If we assume that the candidate cost functions are continuous, c-strongly-monotonic, and L-Lipschitz, then we can use results in sensitivity analysis and approximation theory to derive upper bounds on $\|\mathbb{P} - \mathbb{P}_N\|_{\mathcal{L}}$ of the form $\mathbb{P}[\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}} \leq f(N, \delta, c, L)] \geq 1 - \delta$. In practice, this result enables to derive a sufficient condition on the number N of observations such that $\mathbb{P}[\|\mathbb{P} - \mathbb{P}_N\|_{\mathcal{L}} \leq \epsilon] \geq 1 - \delta$:

$$\sqrt{N} \geq \frac{\sqrt{|\mathcal{E}|}\Big(60 + (L-c)\sqrt{2\log(\frac{1}{\delta})}\Big)}{\epsilon} \frac{c\,(L-c)}{L^2\big(1 - \sqrt{1 - \frac{c^2}{L^2}}\big)}$$

By doing some asymptotic analysis, we have

$$\frac{L}{c} \longrightarrow \infty \quad \Longrightarrow \quad \sqrt{N} \gtrsim \frac{2c\sqrt{|\mathcal{E}|}\sqrt{2\log(\frac{1}{\delta})}}{\epsilon} \left(\frac{L}{c}\right)^2$$

$$1 - \frac{c}{L} \longrightarrow 0 \quad \Longrightarrow \quad \sqrt{N} \gtrsim \frac{60\sqrt{|\mathcal{E}|}}{\epsilon} \left(1 - \frac{c}{L}\right)$$

The number of samples that is needed to maintain a low uniform deviation grows with the ratio L/c raised to the fourth power. If the ratio c/L goes to 1, the number of samples needed decreases quadratically with $1 - c/L$. An extension of our work consists in deriving optimal rates on the number N of samples, *i.e.* to ensure that the above bounds are tight. A possible approach aims at deriving lower bounds on the uniform deviation of the form $\mathbb{P}[\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}} \geq g(N, \delta, c, L)] \geq 1 - \delta$. Such bounds enable to derive upper bounds on the number N of samples so that $\mathbb{P}[\|\mathbb{P}_N - \mathbb{P}\|_{\mathcal{L}} \geq \epsilon]$ with high probability.

In Chapter 7 we proposed a framework for imputing the function that describes an optimization or equilibrium process from observations of traffic flows that are approximately in equilibrium. We formulate the resulting inverse optimization and variational inequality problems as a multi-objective optimization problem in which we want to simultaneously minimize the gap function, which guarantees that the equilibrium condition is approximately satisfied, and the deviation from the model prediction and the flow measurements. We then applied a block coordinate descent algorithm to infer the edge cost functions and price tolls on the road network of Los Angeles.

In Chapter 8, we explored the statistical implications of the optimization framework proposed in Chapter 7. In particular, we used results in concentration inequalities to show how the value of the objective function concentrates in a neighborhood of the distance between the learned and the true models. To obtain such results, we assumed that the measurement noise is distributed according to a Gaussian distribution, results on the concentration of Lipschitz functions of Gaussian variables. In general, it still remains an open question whether or not a similar property holds for sub-Gaussian variables. In the case of distribution-free bounded random errors, dimensionless concentration results can still be obtained by using bounded differences inequalities.

In Chapter 9, we considered a stretch of highway, that is discretized into n cells. And we modeled the flow dynamics on the highway with a discretized hyperbolic partial differential

equation. We showed that the discretized system is a hybrid system that switches between K linear system dynamics. However, the number K of modes grows exponentially with the number n of cells. We proposed to reduce the number of modes to a tractable one by applying a clustering algorithm. Combined with an algorithm for the estimation of hybrid systems such as the interactive multiple model (IMM), we got performance improvements compared to the state-of-the-art Ensemble Kalman filter.

In Chapter 10, we partially addressed the shortage of traditional traffic monitoring sensors, such as loop detectors and video cameras, by leveraging the large penetration of mobile phones among the driving population. We proposed a framework for the fusion of cellular and loop data.

Part IV

Appendices

Appendix A

Miscellaneous

A.1 Resiliency of Mobility-as-a-Service Systems to Denial-of-Service Attacks

An additional work that is loosely related to the rest of our dissertation is the study of the resiliency of Mobility-as-a-Service (MaaS) systems such as ride sharing services (*e.g.*, Uber, Lyft) to Denial-of-Service (DOS) attacks. In our paper [152], we note that MaaS systems have expanded very quickly over the past years. However, the popularity of MaaS systems make them increasingly vulnerable to DOS attacks, in which attackers attempt to disrupt the system to make it unavailable to the customers. Expanding on an established queuing-theoretical model for MaaS systems, attacks are modeled as a malicious control of a fraction of vehicles in the network. We then formulate a stochastic control problem that maximizes the passenger loss in the network, and solve it as a sequence of linear and quadratic programs. Combined with an economic model of supply and demand for attacks, we quantify how raising the cost of attacks (via cancellation fees and higher level of security) removes economical incentives for DoS attacks. Calibrating the model on 1B taxi rides, we dynamically simulate a system under attack and estimate the passenger loss under different scenarios, such as arbitrarily depleting taxis or maximizing the passenger loss. Cost of attacks of $15 protects the MaaS system against DoS attacks. The contributions are thus a theoretical framework for the analysis of the network, and practical conclusions in terms of financial countermeasures to the attacks.

A.2 Graphic design

Beyond my scientific contributions, I also had a lot of fun designing posters and logos for the EECS department at UC Berkeley.

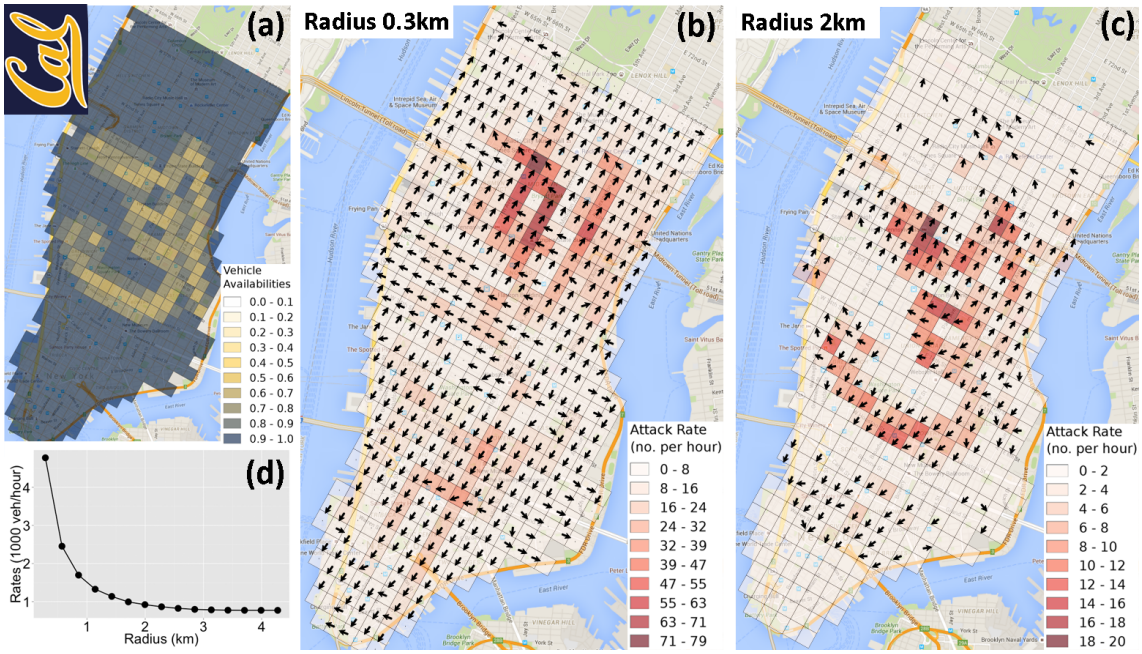

Figure A.1: Best DOS attack strategy to achieve the target following a pixelated version of the "Cal" logo.

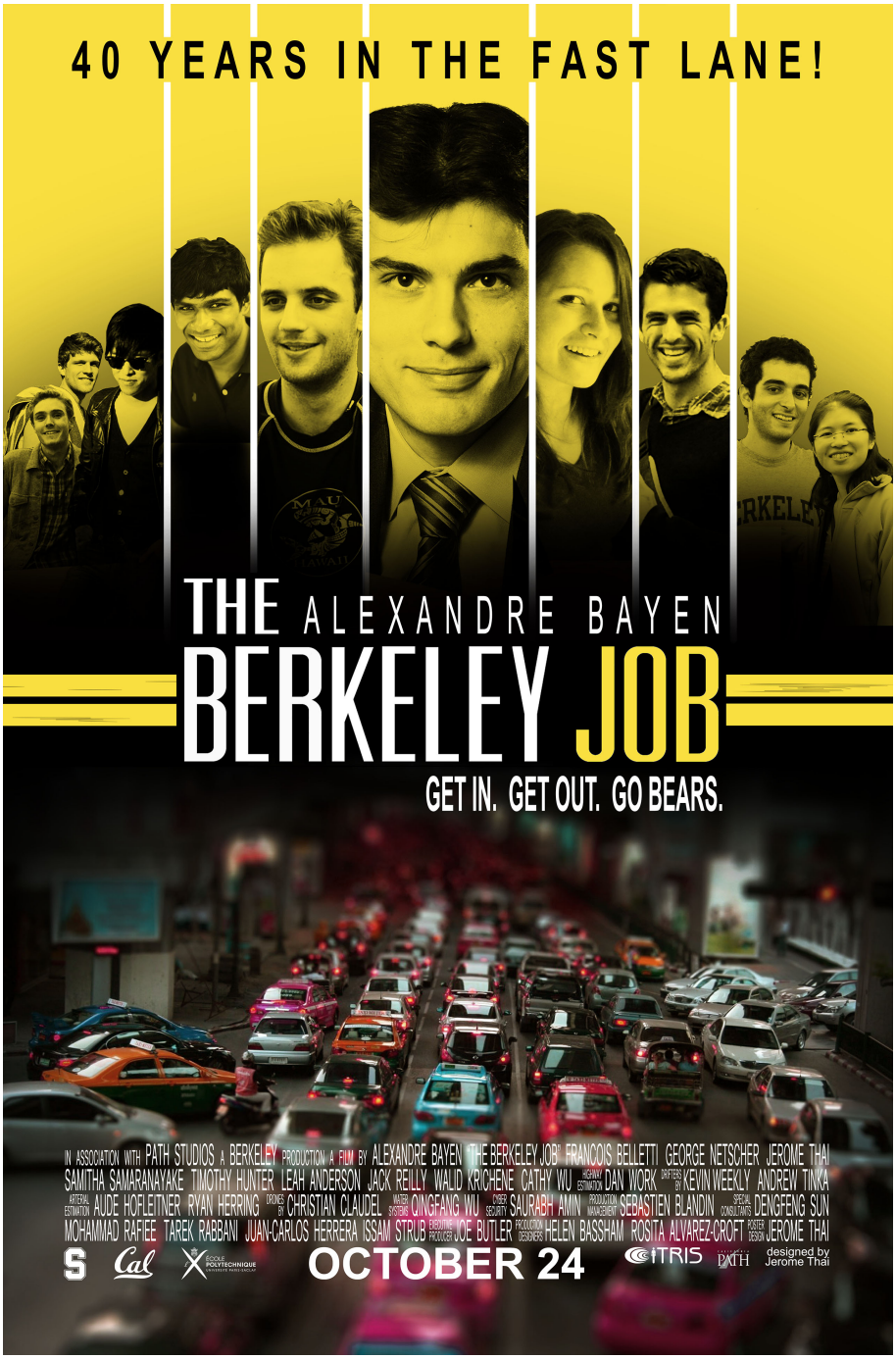

Figure A.2: Poster for Alex's 40th birthday inspired from the poster of the movie "The Italian Job".

Figure A.3: My logo design (with elements and suggestions from Grace, Ken, Betsy, Elizabeth, Joël, and others) got selected to represent our program: `http://bair.berkeley.edu/` go BAIRs!

Figure A.4: My logo design was printed on t-shirts and bags for the visit days of the EECS depatment.

Bibliography

[1] P. Abbeel and A. Ng. "Apprenticeship learning via inverse reinforcement learning". In: *Proceedings of 21st International Conference on Machine Learning*. 2004.

[2] T. Abrahamsson. "Estimation of origin-destination matrices using traffic counts - a literature survey". In: *Interim Report IR-98-021, International Institute for Applied Systems Analysis, Laxenburg, Austria* (1998).

[3] D. Ackerberg et al. "Econometric Tools for Analyzing Market Outcomes". In: *Handbook of Econometrics* 6 (2007).

[4] M. Aghassi, D. Bertsimas, and G. Perakis. "Solving asymmetric variational inequalities via convex optimization". In: *Operations Research Letters* 34.5 (2006), pp. 481–490.

[5] M. Anthony. "Uniform Glivenko-Cantelli theorems and concentration of measure in the mathematical modelling of learning". In: *Research Report LSE-CDAM-2002-07* (2002).

[6] A.-E. Baert and D. Seme. "Voronoi mobile cellular networks: topological properties". In: *In Third International Symposium on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks* (2004), pp. 29–35.

[7] P. Bajari, C. Benkard, and J. Levin. "Estimating dynamic models of imperfect competition". In: *Econometrica* 75 (2007), pp. 1331–1370.

[8] Bar-Gera. "Origin-based Algorithm for the Traffic Assignment Problem". In: *Transportation Science* (2002).

[9] Y. Bar-Shalom and X. R. Li. "Estimation and Tracking: Principles, Techniques, and Software." In: *Norwood, MA: Artech House* (1993).

[10] C. Bardos, A. Y. Leroux, and J. C. Nedelec. "First order quasilinear equations with boundary conditions". In: *Communications in partial differential equations 4* 9 (1979), pp. 1017–34.

[11] P. L. Bartlett, O. Bousquet, and S. Mendelson. "Local Rademacher complexities". In: *Annals of Statistics* 33.4 (2005), pp. 1497–1537.

[12] P.L. Bartlett and S. Mendelson. "Rademacher and gaussian complexities: risk bounds and structural results". In: *Journal of Machine Learning Research* 3 (2003), pp. 463–482.

[13] M. Beckmann, C. B. McGuire, and C. B. Winsten. *Studies in the Economics of Transportation*. Ed. by New Haven Yale University Press. Cowles Commission Monograph, 1956.

[14] Martin J Beckmann, Charles B McGuire, and Christopher B Winsten. "Studies in the Economics of Transportation". In: (1955).

[15] M. G. H. Bell and Y. Iida. *Transportation Network Analysis*. Wiley, West Sussex, United Kingdom, 1997.

[16] Moshe Ben-Akiva et al. "Modelling Inter Urban Route Choice Behavior". In: *International Symposium on Transportation and Traffic Theory* (1984), pp. 299–330.

[17] Moshe E. Ben-Akiva and Steven R. Lerman. "Discrete Choice Analysis: Theory and Application to Travel Demand". In: *MIT Press* (1985).

[18] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2005.

[19] D. Bertsimas, V. Gupta, and I. Paschalidis. "Data-driven estimation in equilibrium using inverse optimization". In: *Mathematical Programming* (2015), pp. 595–633.

[20] D. Bertsimas, V. Gupta, and I. Ch. Paschalidis. "Data-Driven Estimation in Equilibrium Using Inverse Optimization". In: *Mathematical Programming* (2014).

[21] M. J. Best and N. Chakravarti. "Active set algorithms for isotonic regression; a unifying framework". In: *Math. Programming* 47 (1990), pp. 425–439.

[22] S. Blandin, L. E. Ghaoui, and A. Bayen. "Kernel regression for travel time estimation via convex optimization". In: *IEEE Conference on Decision and Control* (2009).

[23] S. Blandin et al. "On sequential data assimilation for scalar macroscopic traffic flow models". In: *Physica D* (2012).

[24] H. A. P. Blom and Y. Bar-Shalom. "The interacting multiple model algorithm for systems with Markovian switching coefficients". In: *IEEE Trans. Autom. Control* AC-33 (1988), pp. 780–783.

[25] S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2016.

[26] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[27] S. Boyd and L. Vandenberghe. *Convex Optimization*. Ed. by Cambridge University Press. Vol. 25. Cambridge University Press, 2010.

[28] S. Boyd et al. *Linear Matrix Inequalities in Systems and Control Theory*. Philadelphia: SIAM books, 1994.

[29] D. Branston. "Link capacity functions: a review". In: *Transportation Research* 10.4 (1976), pp. 223–236.

[30] P. N. Brown and J. R. Marden. "Avoiding perverse incentives in affine congestion games". In: *55th IEEE Conference on Decision and Control*. 2016.

[31] "Bureau of Public Roads: Traffic assignment manual". In: *US Department of Commerce, Urban Planning Division* (1964).

[32] D Burton, W. Pulleyblank, and P. Toint. "The inverse shortest path problem with upper bounds on shortest path costs". In: *Network Optimization* 450 (1997), pp. 156–171.

[33] N. Caceres, J. P. Wideberg, and F. G. Benitez. "Deriving origin-destination data from a mobile phone network". In: *IET Intell. Transp. Syst* 1 (2007), pp. 15–26.

[34] F. Calabrese et al. "Estimating Origin-Destination Flows Using Mobile Phone Location Data". In: *IEEE Pervasive Computing* (2011), pp. 36–44.

[35] J. Candia et al. "Uncovering individual and collective human dynamics from mobile phone records". In: *Journal of Physics A: Mathematical and Theoretical* (2008).

[36] J. Candia et al. "Uncovering individual and collective human dynamics from mobile phone records". In: *Journal of Physics A: Mathematical and Theoretical* 41 (2008).

[37] B. Carl and I. Stephani. *Entropy, compactness and the approximation of operators*. Cambridge, UK: Cambridge Tracts in Mathematics. Cambirdge University Press, 1990.

[38] E. Castillo, J. M. Menendez, and P. Jimenez. "Trip matrix and path flow reconstruction and estimation based on plate scanning and link observations". In: *Transportation Research Part B: Methodological* 42 (2008), pp. 455–481.

[39] E. Castillo et al. "Optimal Use of Plate-Scanning Resources for Route Flow Estimation in Traffic Networks". In: *IEEE Transactions on Intelligent Transportation Systems* 11 (2010), pp. 380–391.

[40] R. Chen and J. S. Liu. "Mixture Kalman filters". In: *Royal Statistical Society* 62 (2000), pp. 493–508.

[41] Y. Chen and M. Florian. "Congested O-D trip demand adjustment problem: bilevel programming formulation and optimality conditions". In: *Kluwer Academic Publishers* (1998).

[42] Y. Chen and M. Florian. "The nonlinear bilevel programming problem: Formulations, regularity and optimality conditions". In: *Optimization* 32 (1995), 193–209.

[43] T. Choe, A. Skabardonis, and P. Varaiya. "Freeway performance measurement system (PeMS): an operation tool". In: *81st Annual Meeting Transportation Research Board, Washington, DC* (2002).

[44] Serdar Colak, Antonio Lima, and Marta C. Gonzalez. "Understanding congested travel in urban areas". In: *Nature Communications* (2016).

[45] John Conlisk. "Why Bounded Rationality?" In: *Journal of Economic Literature* 34.2 (1996), pp. 669–700.

[46] J. R. Correa, A. S. Schulz, and N. E. Stier-Moses. "On the Inefficiency of Equilibria in Congestion Games". In: *Integer Programming and Combinatorial Optimization: 11th International IPCO Conference, Berlin, Germany, June 8-10, 2005. Proceedings*. Ed. by Michael Jünger and Volker Kaibel. Springer Berlin Heidelberg, 2005, pp. 167–181.

[47] J. R. Correa and N. E. Stier-Moses. "Wardrop equilibria". In: *Wiley encyclopedia of operations research and management science* (2011).

[48] S. Dafermos. "Sensitivity analysis in variational inequalities". In: *Mathematics of Operations Research* 13 (1988), pp. 421–434.

[49] S. Dafermos. "Traffic Equilibrium and Variational Inequalities". In: *Transportation Science* 14.1 (1980), pp. 42–54.

[50] C. F. Daganzo. "The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory". In: *Transportation Research Part B 28, no. 4* 28 (1994), pp. 269–287.

[51] C. F. Daganzo. "The cell transmission model, part II: Network traffic". In: *Transportation Research Part B 29, no. 2* 29 (1995), pp. 79–93.

[52] C. F. Daganzo and Y. Sheffi. "On stochastic models of traffic assignment". In: *Transportation Science* 11 (1977), pp. 253–274.

[53] S. Dempe. *Foundations of Bilevel Programming*. Springer, 2002.

[54] R. A. DeVore and G. G. Lorentz. *Constructive approximation*. New York, NY: Springer-Verlag, 1993.

[55] R. Dowling, A. Skabardonis, and V. Alexiadis. *Traffic analysis toolbox volume III: guidelines for applying traffic microsimulation modeling software*. Tech. rep. 2004.

[56] J. Duchi, S. Gould, and D. Koller. "Projected subgradient methods for learning sparse gaussians". In: *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence* (2008).

[57] R. M. Dudley. "The sizes of compact subsets of Hilbert space and continuity of Gaussian processes". In: *J. Functional Analysis* 1 (1967), pp. 290–330.

[58] M. Ehrgott. *Multicriteria optimization*. Springer Verlag, 2005.

[59] S. Engevall, M. Gothe-Lundgren, and P. Varbrand. "The heterogeneous vehicle-routing game". In: *Transportation Science* 38 (2004), pp. 71–85.

[60] F. Facchinei, H. Jiang, and L. Qi. "A smoothing method for mathematical programs with equilibrium constraints". In: *Mathematical Programming* 85 (1999), pp. 107–134.

[61] F. Facchinei and J-S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Series in Operations Research. New York: Springer, 2003.

[62] Farhad Farokhi et al. "A Heterogeneous Routing Game". In: *51st Allerton Conference on Communication, Control and Computing (Allerton)*. 2013.

[63] Farhad Farokhi et al. "A Heterogeneous Routing Game". In: 2013.

[64] C. Fisk. "Some developments in equilibrium traffic assignment". In: *Transportation Research Part B* 14 (1980), pp. 243–255.

[65] L. Fleischer, K. Jain, and M. Mahdian. "Tolls for heterogeneous selfish users in multicommodity networks and generalized congestion games". In: *45th Annual IEEE Symposium on Foundations of Computer Science* (2004), pp. 277–285.

[66] Joe Flint. "In L.A., One Way to Beat Traffic Runs Into Backlash". In: *The Wall Street Journal* (2016).

[67] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ, 1962.

[68] M Frank and P. Wolfe. "An algorithm for quadratic programming". In: *Naval Res. Logis. Quart.* 3 (1956), pp. 95–110.

[69] Song Gao, Emma Frejinger, and Moshe Ben-Akiva. "Cognitive cost in route choice with real-time information: An exploratory analysis". In: *Transportation Research Part A* 17 (2011), pp. 136–149.

[70] S. van de Geer. *Empirical Processes in M-Estimation*. Cambridge University Press, 2000.

[71] E. Godlewski and P-A. Raviart. *Numerical approximation of hyperbolic systems of conservation laws*. Applied Mathematical Sciences, 1996.

[72] S.K. Godunov. "A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics". In: *Math. Sbornik* 47 (1959), pp. 271–306.

[73] B. D. Greenshields. "A study of traffic capacity". In: *Proceedings of the 14th annual meeting of the Highway Research Board* 14 (1934), pp. 448–477.

[74] S. J. Grotzinger and C. Witzgall. "Projection onto Order Simplexes". In: *Applied Mathematics and Optimization* 12 (1984), pp. 247–270.

[75] Branko Grünbaum. *Convex Polytopes*. Springer, 2003.

[76] J. H. Hammond. "Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms". PhD thesis. Massachusetts Institute of Technology, 1984.

[77] T.J. Hastie and R. Tibshirani. "Generalized additive models". In: *Statistical Science* 1 (), pp. 297–310.

[78] T.J. Hastie, R.J. Tibshirani, and J.H. Friedman. *The elements of statistical learning*. Series in statistics. New York: Springer, 2009.

[79] E. Hato et al. "Incorporating an information acquisition process into a route choice model with multiple information sources". In: *Transportation Research Part C* 7 (1999), pp. 109–129.

[80] R. M. Hawkes and J. B. Moore. "Performance bounds for adaptive estimation". In: *Proc. IEEE* 64 (1976), pp. 1143–1150.

[81] J.-C. Herrera et al. "Evaluation of traffic data obtained via GPS-enabled mobile phones: the Mobile Century experiment". In: *Transportation Research Part C* 18 (2009), pp. 568–583.

[82] T. Hunter et al. "Path and travel time inference from GPS probe vehicle data". In: *NIPS Analyzing Networks and Learning with Graphs* (2009).

[83] I. Hwang, H. Balakrishnan, and C. Tomlin. "State estimation for hybrid systems: applications to aircraft tracking". In: *IEE Proc. Control Theory and Applications* 153.5 (2006), pp. 556–566.

[84] G. Flötteröd Illenberger J. and K. Nagel. "Enhancing MATSim with capabilities of within-day re-planning". In: *IEEE Intelligent Transportation Systems Conference* (2007).

[85] G. Iyengar and W. Kang. "Inverse conic programming with applications". In: *Operations Research Letters* 33 (2005), pp. 319–330.

[86] Y. Singer T. Chandra J. Duchi S. Shalev-Shwartz. "Efficient Projections onto the l1-Ball for Learning in High Dimensions". In: *Proceedings of the 25th International Conference on Machine Learning* (2008).

[87] M. Jaggi. "Revisiting Frank-Wolfe: Projection-free sparse convex optimization". In: *Journal of Machine Learning Research* (2013).

[88] A. Janecek et al. "Cellular data meet vehicular traffic theory: location area updates and cell transitions for travel time estimation". In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM. 2012, pp. 361–370.

[89] H. Jiang, D. Ralph, and J. Pang. "QPECgen, a MATLAB generator for mathematical programs with quadratic objectives and affine variational inequality constraints". In: *Computational Optimization and Applications* 13 (1999), pp. 25–59.

[90] S. Jiang et al. "A review of urban computing for mobile phone traces: current methods, challenges and opportunities". In: *Proc. of the 2nd ACM SIGKDD International Workshop on Urban Computing*. ACM. 2013, p. 2.

[91] G. Karakostas and S. Kolliopoulos. "Edge pricing of multicommodity networks for heterogeneous selfish users". In: *45th IEEE Symp. on Foundations of Computer Science* (2004), pp. 268–276.

[92] F. P. Kelly. "Network routing". In: *Philosophical Transactions: Physical Sciences and Engineering* 337 (1991), pp. 343–367.

[93] A. Keshavarz, Y. Wang, and S. Boyd. "Imputing a convex opjective function". In: *IEEE International Symposium on Intelligent Control*. 2011.

[94] A. Keshavarz, Y. Wang, and S. Boyd. "Imputing a convex opjective function". In: *IEEE International Symposium on Intelligent Control*. 2011.

[95] D. Kinderlehrer and G. Stampacchia. *An Introduction to Variational Inequalities and Their Applications*. Society for Industrial and Applied Mathematics, 2000.

[96] A. N. Kolmogorov and B. Tikhomirov. "ϵ-entropy and ϵ-capacity if sets in functional spaces". In: *Uspekhi Mat. Nauk* 17 (1961), pp. 277–264.

[97] V. Koltchinskii. "Local Rademacher complexities and oracle inequalities in risk minimization". In: *Annals of Statistics* 34 (2006), pp. 2593–2656.

[98] V. Koltchinskii. "Rademacher penalties and structural rick minimization". In: *IEEE Trans. Information Theory* 47 (2001), pp. 1902–1914.

[99] V. Koltchinskii and D. Panchenko. "Rademacher processes and bounding the risk of function learning". In: *High-dimensional probability II* (2000), pp. 223–236.

[100] J. P. Lebacque. "The Godunov scheme and what it means for first order traffic flow models". In: *13th International Symposium on Transportation and Traffic Theory* (1996), pp. 647–77.

[101] L. J. LeBlanc, E. K. Morlok, and W. P. Pierskalla. "An efficient approach to solving the road network equilibrium traffic assignment problem". In: *Transportation Research* 9 (1975), pp. 309–318.

[102] M. Ledoux and M. Talagrand. *Probability in Banach spaces*. Springer-Verlag, 1991.

[103] P. LeFloch. "Explicit formula for scalar non-linear conservation laws with boundary condition". In: *Math. Meth. Appl. Sci.* 10 (1988), pp. 265–87.

[104] M. D. Lemmon, K. X. He, and I. Markovsky. "Supervisory hybrid systems". In: *IEEE Control Systems Magazine* 19 (4) (1999), pp. 42–55.

[105] B. Lennartson et al. "Hybrid systems in process control". In: *IEEE Control Systems Magazine* 16 (5) (1996), pp. 45–56.

[106] R. J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser Basel, 1992.

[107] X. R. Li and Y. Bar-Shalom. "Design of an Interacting Multiple Model Algorithm for Air Traffic Control Tracking". In: *IEEE Transactions on Control Systems Technology* 1.3 (1993).

[108] X. R. Li and Y. Bar-Shalom. "Performance prediction of the interacting multiple model algorithm". In: *IEEE Trans. Aerosp. Electron. Syst* 29 (1993), pp. 755–771.

[109] M. J. Lighthill and G. B. Whitham. "On Kinematic Waves II. A Theory of Traffic Flow on Long Crowded Roads". In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 229 (1955), pp. 317–345.

[110] Z.Q. Luo, J.S. Pang, and D. Ralph. *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, 1996.

[111] M. J. Maher and P. C. Hughes. "A probit-based stochastic user equilibrium assignment model". In: *Transportation Research* 31 (1997), pp. 341–355.

[112] Hani S. Mahmassani and Gang-Len Chang. "On Boundedly Rational User Equilibrium in Transportation Systems". In: *Transportation Science* 21 (1987).

[113] J. Mandel. "Efficient Implementation of the Ensemble Kalman Filter". In: *CCM Report No. 231* (2006).

[114] P. Marcotte and D. L. Zhu. "Existence and computation of optimal tolls in multiclass network equilibrium problems". In: *Operations Research Letters* 37 (2009), pp. 211–214.

[115] M. Mardani and G. B. Giannakis. "Robust network traffic estimation via sparsity and low rank". In: *IEEE International Conference on Acoustics, Speech and Signal Processing* (2013).

[116] R.T. Marler and J.S. Arora. "Survey of multi-objective optimization methods for engineering". In: *Structural And Multidisciplinary Optimization* (2004), pp. 369–395.

[117] J. Mathew and P.M. Xavier. "A SURVEY ON USING WIRELESS SIGNALS FOR ROAD TRAFFIC DETECTION". In: ().

[118] P. S. Maybeck. "Stochastic models, estimation, and control". In: *Academic Press* 2 (1982).

[119] L. Meier, S. van de Geer, and P. Buhlmann. "High-dimensional additive modeling". In: *Annals of Statistics* 37 (2009), pp. 3779–3821.

[120] D. Monderer and L. S. Shapley. "Potential Games". In: *Games and Economic Behavior* 14 (1996), pp. 124–143.

[121] Dov Monderer and Lloyd S Shapley. "Potential games". In: *Games and economic behavior* 14.1 (1996), pp. 124–143.

[122] Y. Nesterov. "Introductory Lectures on Convex Optimization. A Basic Course". In: *Kluwer* (2004).

[123] A. Ng and S. Russell. "Algorithms for inverse reinforcement learning". In: *Proceedings of 17th International Conference on Machine Learning*. 2000, pp. 663–670.

[124] J. Nocedal and S. Wright. *Numerical Optmization*. Springer, 2nd edition, 2006.

[125] J. de D. Ortuzar and L.G. Willumsen. *Modelling Transport*. 3rd, Edition, Wiley, West Sussex, United Kingdom, 2001.

[126] J-S. Pang. "A posteriori error bounds for the linearly-constrained variational inequality problem". In: *Mathematics of Operations Research* 12.3 (1987), pp. 4474–484.

[127] C. S. Papacostas and P. D. Prevedouros. *Transportation Engineering and Planning*. Prentice Hall, Upper Saddle River, New Jersey, 2001.

[128] M. Papageorgiou, J.-M. Blosseville, and H. Hadj-Salem. "Modelling and real-time control of traffic flow on the southern part of Boulevard Peripherique in Paris: Part I: Modelling". In: *Transportation Research* 24 (5) (1990), pp. 345–359.

[129] A. Patire et al. "How much GPS data do we need?" In: *Transportation Research Part C* (2013).

[130] M. Patriksson. *The Traffic Assignment Problem - Models and Methods*. VSP, Utrecht, 1994.

[131] M. Patriksson. *The Traffic Assignment Problem: Models and Methods*. Dover Publications, 2015.

[132] A. Pinkus. *N-Widths in Approximation Theory*. New York, NY: Springer, 1985.

[133] Y. Qiu and T. Magnanti. "Sensitivity Analysis for Variational Inequalities Defined on Polyhedral Sets". In: *Mathematics of Operations Research* 14 (1989), 410–432.

[134] M. Rahmani and H. N. Koutsopoulos. "Path inference from sparse floating car data". In: *Transportation Research Part C: Emerging Technologies* 30 (2013), pp. 41–54.

[135] N. Ratliff, J. Bagnell, and M. Zinkevich. "Maximum margin planning". In: *Proceedings of the 23rd International Conference on Machine Learning*. 2006.

[136] P. I. Richards. "Shock Waves on the Highway". In: *Operations Research* 4 (1956), pp. 42–51.

[137] T. Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, 2005.

[138] T. Roughgarden. "The price of anarchy is independent of the network topology". In: *Journal of Computer and System Sciences* 67 (2003), pp. 341–364.

[139] J. Rust. "Structural Estimation of Markov decision processes". In: *Handbook of Econometrics* 4 (1994), pp. 3081–3143.

[140] William H Sandholm. "Potential games with continuous player sets". In: *Journal of Economic Theory* 97.1 (2001), pp. 81–108.

[141] T. Schreiter et al. "Data-model synchronization in extended Kalman filters for accurate online traffic state estimation". In: *2010 Proceedings of the Traffic Flow Theory Conference, Annecy, France* (2010).

[142] Gesualdo Scutari et al. "Convex Optimization, Game Theory, and Variational Inequality Theory". In: *IEEE Signal Processing Magazine* 35 (2010).

[143] Y. Sheffi. *Urban Transportation Networks*. Prentice-Hall, Englewood Cliffs, NJ, 1985.

[144] W. Shen and L. Wynter. "A new one-level convex optimization approach for estimating origin–destination demand". In: *Transportation Research Part B: Methodological* 46 (2012), pp. 1535–1555.

[145] C.J. Stone. "Additive regression and other non-parametric models". In: *Annals of Statistics* 13 (1985), pp. 689–705.

[146] I. S. Strub and A. M. Bayen. "Weak formulation of boundary conditions for scalar conservation laws: an application to highway traffic modeling". In: *Int. J. Robust Nonlinear Control* 16 (2006), pp. 733–748.

[147] D. D. Sworder and J. E. Boyd. "Estimation problems in hybrid systems". In: *Cambridge University Press* (1999).

[148] T. Tettamanti, H. Demeter, and I. Varga. "Route Choice Estimation Based on Cellular Signaling Data". In: *Acta Polytechnica Hungarica* 9.4 (2012), pp. 207–220.

[149] J. Thai. "Negative Externalities of GPS-Enabled Routing Applications: A Game Theoretical approach". In: *19th IEEE Conference on Intelligent Transportation Systems* (2016), pp. 595–601.

[150] J. Thai and A. Bayen. "Imputing a Variational Inequality or Convex Objective function: a Robust Approach". In: *Journal of Mathematical Analysis and Applications* (2016).

[151] J. Thai, B. Prodhomme, and A. M. Bayen. "State Estimation for the discretized LWR PDE using explicit polyhedral representations of the Godunov scheme". In: *In review, American Control Conference* (2013).

[152] J. Thai, C. Yuan, and A. Bayen. "Resiliency of Mobility-as-a-Service Systems to Denial-of-Service Attacks". In: *IEEE Transactions on Control of Network Systems* (2016).

[153] R. J. Tibshirani, H. Hoefling, and R. Tibshirani. "Nearly-Isotonic Regression". In: *Technometrics* 53 (2011).

[154] JL. Toole et al. "Inferring land use from mobile phone activity". In: *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*. 2012, pp. 1–8.

[155] Wisconsin Department of Transportation (WisDOT). *Unofficial WI Traffic Analysis Guidelines, Draft*. `http://www.wisdot.info/microsimulation/index.php?title=Model_Calibration`. [Online; accessed 2014-08-30]. 2013.

[156] Howard G. Tucker. "A Generalization of the Glivenko-Cantelli Theorem". In: *The Annals of Mathematical Statistics* 30 (1959), pp. 828–830.

[157] A. W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 1998.

[158] V. N. Vapnik. *The Nature of Statistical Learning Theory*. New York, NY: Springer, 1995.

[159] H. Veeraraghavan, O. Masoud, and N. Papanikolopoulos. "Computer Vision Algorithms for Intersection Monitoring". In: *IEEE Transactions on Intelligent Transportation Systems* 4 (2003), pp. 78–89.

[160] C. Volinsky et al. "Clustering Anonymized Mobile Call Detail Records to Find Usage Groups". In: *1st Workshop on Pervasive Urban Applications (PURBA)* (2011).

[161] C. Volinsky et al. "Route Classification using Cellular Handoff Patterns". In: *13th ACM International Conference on Ubiquitous Computing* (2011).

[162] W. Wang and M. Á. Carreira-Perpiñán. "Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application". In: *CoRR* (2013).

[163] Y. Wang and S. Boyd. "Fast evaluation of quadratic control-lyapunov policy". In: *IEEE Transactions on control Systems Technology* (2010), pp. 1–8.

[164] Y. Wang and S. Boyd. "Fast model predictive control using online optimization". In: *17th IFAC world congress*. 2008.

[165] J. G. Wardrop and J. I. Whitehead. "Correspondence. Some Theoretical Aspects of Road Traffic Research". In: *ICE Proceedings: Engineering Divisions 1* (1952).

[166] John Glen Wardrop. "SOME THEORETICAL ASPECTS OF ROAD TRAFFIC RESEARCH." In: *ICE Proceedings: Engineering Divisions*. Vol. 1. 3. Thomas Telford. 1952, pp. 325–362.

[167] J. Watkins and P. Dayan. "Technical note: Q-learning". In: *Machine Learning* 8 (1992), pp. 279–292.

[168] J. White and I. Wells. "Extracting origin destination information from mobile phone data". In: *11th Int. Conf. on Road Transport Information and Control, London* (2002), pp. 30–34.

[169] D. Work et al. "An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices". In: *47th IEEE Conference on Decision and Control* (2008).

[170] D. B. Work et al. "A Traffic Model for Velocity Data Assimilation". In: *Applied Mathematics Research eXpress* (2010).

[171] S. Yadlowsky et al. "Link Density Inference from Cellular Infrastructure". In: *Submitted to Transportation Research Board (TRB) 94th Annual Meeting* (2014).

[172] J. J. Ye and D. L. Zhu. "Optimality conditions for bilevel programming problems". In: *Optimization: A Journal of Mathematical Programming and Operations Research* 33 (1995).

[173] J. Y. Yen. "Finding the k Shortest Loopless Paths in a Network". In: *Management Science* 17 (1971), pp. 712–716.

[174] N. D. Yen. "Lipschitz continuity of solutions of variational inequalities with a parametric polyhedral constraint". In: *Mathematics of Operations Research* 20.3 (1995).

[175] Dao Li Zhu and Patrice Marcotte. "Convergence Properties of Feasible Descent Methods for Solving Variational Inequalities in Banch SPaces". In: *Computational Optimization and Applications* 10 (1997), pp. 35–49.