Title
A hybrid approach of physical laws and data-driven modeling for estimation: the example of queuing networks

Author
Hofleitner, Aude

Publication Date
2013

Peer reviewed|Thesis/dissertation

A hybrid approach of physical laws and data-driven modeling for estimation: the example of queuing networks

by

Aude Hofleitner

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Alexandre Bayen, Chair
Professor Pieter Abbeel
Professor Laurent El Ghaoui
Professor Alexandre Skabardonis

Spring 2013

A hybrid approach of physical laws and data-driven modeling for estimation: the example of queuing networks

Abstract

A hybrid approach of physical laws and data-driven modeling for estimation: the example of queuing networks

by

Aude Hofleitner

Doctor of Philosophy in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Alexandre Bayen, Chair

Mathematical models are a mathematical abstraction of the physical reality which is of great importance to understand the behavior of a system, make estimations and predictions and so on. They range from models based on physical laws to models learned empirically, as measurements are collected, and referred to as data-driven models. A model is based on a series of choices which influence its complexity and realism. These choices represent trade-offs between different competing objectives including interpretability, scalability, accuracy, adequation to the available data, robustness or computational complexity. The thesis investigates the advantages and disadvantages of models based on physical laws versus data-driven models through the example of signalized queuing networks such as urban transportation networks.

The dynamics of conservation flow networks are accurately represented by a first order partial differential equation. Using Hamilton-Jacobi theory, the thesis underlines the importance to leverage physical laws to reconstruct missing information (*e.g.* signal or bottleneck characteristics) and estimate the state of the network at any time and location. Noise and uncertainty in the measurements can be integrated in the model. When measurements are sparse, the state of the network cannot be estimated at every time and location on the network. Instead, the thesis shows how to leverage other characteristics, such as periodicity. From deterministic dynamics, the thesis derives the probability distribution functions of physical entities (*e.g.* waiting time, density) by marginalizing the periodic variable. Using a Dynamic Bayesian Network formulation and exploiting the convexity structure of the system, the thesis shows how this modeling leads to realistic estimations and predictions, even when little measurements are available. Finally, the thesis investigates how sparse modeling and dimensionality reduction can provide insights on the large scale behavior of the network. Large scale dynamics and patterns are hard to model accurately based on physical laws. They can be discovered through data mining algorithms and integrated into physical models.

This dissertation is dedicated to my wonderful family.
To my mother, Anne. I wish I could share this achievement with you. You taught me how to embrace life and I am thankful for that everyday.
To my father, Patrick, and my sister, Céline who have always believed in me and encouraged me.
To my boyfriend, Ryan. Your love has supported me throughout this incredible journey. I feel privileged to have you by my side everyday.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to take this very special opportunity to acknowledge the importance of my family in the completion of this dissertation. My family has always supported my choices and encouraged me to seek interesting challenges and opportunities. In particular, I am grateful to my mom, Anne. She is sadly no longer among us to share this moment but I would like to emphasize her essential role in my life. She had a passion for travel, for working in international environment and socializing with people from various cultures which has strongly shaped my personality and my interests, both personally and professionally. She has always encouraged me to seek international opportunities and to benefit from the amazing cultural experiences that they provide. One of these opportunities was to go to Berkeley to join the *Mobile Millennium* team for a one year internship, starting in July 2008. I seized the opportunity and immersed myself in an amazing project, working with incredibly interesting and welcoming people. The end of 2008 brought dramatic emotions as she unexpectedly and so suddenly passed away. I am grateful for the support I received from my dad, Patrick and my sister, Céline as well as my family and friends following this dramatic event. When the time came to decide between starting the PhD program at UC Berkeley or coming back to France and be close to them, both my dad and sister encouraged me to continue my journey and helped me overcome this tragic life event. Despite the distance, our relationship has grown closer, stronger, and so much deeper.

I was also amazed by the attentive and personal support of my colleagues with whom I had only been working for a few months. In particular, Alex Bayen, my internship supervisor at the time, and my friend and colleague Saneesh Apte played an incredibly valuable role. The research environment at UC Berkeley had already exceeded my expectations in terms of dynamism, interest in the project and intellectual challenge; the realization that I could relate to my colleagues on the personal level as well convinced me that I wanted to pursue my graduate studies in this successful and fulfilling environment.

Since I first met him in a café in Paris in the winter of 2008, Alex Bayen has struck me with his constant level of energy and excitement about the work and research of his team. The dynamism and high level of expectations appealed to me from the beginning. I was not disappointed as I arrived in Berkeley and Alex entrusted me with more responsibilities than I had ever thought I would be able to manage. Alex keeps on surprising me with his trust and the confidence that he expresses for my work and career. He always encourages me to take on new and additional challenges, pushing me beyond what I think myself capable of. The completion of this PhD can, for a fair part, be attributed to his support, contagious dynamism and will to succeed. Beyond the career drive, Alex has been of incredible personal support. Throughout the years, he has combined enjoying relaxing and social times with other fellow students from the lab, working on papers in the middle of the night, congratulating achievements, discussing personal matters and career choices for long hours, tirelessly editing, reshaping and improving presentation of research ideas, giving motivation and en-

ergy during low times of the PhD.

Pieter Abbeel started following my work halfway through my internship and was offering his expertise in machine learning for the arterial estimation efforts that I was pursuing. At the beginning of my PhD, Pieter pointed out an idea which I had first developed as part of a class project: simplifying the arterial traffic flow physics to model the distribution of vehicle location on a road segment. He encouraged me to develop this idea further and generalize it to travel times, in a hybrid approach of traffic theory and statistical modeling and inference. Pieter's vision in this work turned into an essential part of my PhD work. His enthusiasm for the model, the exciting and dynamic meetings and his research directions drove my research interest far beyond my expectations.

I started collaborating with Laurent El Ghaoui through a class project which turned into a broader collaboration with one of Laurent's students, Tarek Rabbani. Since my first introduction to convex optimization at Ecole Polytechnique, I have always been interested in the field. I really appreciated the opportunity that Laurent gave me to be the Graduate Student Instructor for the class *Optimization models in engineering* and to share my enthusiasm for the field with the students of the class.

I have had the pleasure and honor to meet and discuss research ideas with Alex Skabardonis. I have learned a lot from his expertise in traffic and transportation. His feedback has been very important in the development and refinements of the queuing model and statistical model of urban traffic. I value the interest that he showed in a non-conventional approach to traffic modeling as well as his overall vision on the field and on its most important research questions.

I have found my best and dearest PhD collaborator in Ryan Herring. Ryan's view to research and scientific problems seems to perfectly complement mine. Working with him has been incredibly productive and enjoyable. Between developing models, designing code architecture, taking turns at writing papers or working on class material together, we have shared a lot of the PhD joys together. Ryan was also able to help me through the times when I was less motivated about my research or was disappointed by my results. His trust in my potential to succeed and overcome obstacles has always amazed me. I feel privileged to have the chance to have someone so dear to me to celebrate the happy times of the PhD life and give me the courage and strength to overcome the deceptive times.

I benefited from the guidance and mentoring of senior PhD students who have played an important role in the completion of this dissertation. Christian Claudel introduced me to his work on viability theory. His excitement and interest for research and mathematics were contagious and lead to fruitful collaborations throughout my PhD. Saurabh Amin struck me with his mind overflowing with novel ideas and theoretical contributions. I really enjoyed our white board discussions and brainstorming sessions on traffic modeling, estimation, statis-

tics, change detection, sampling, privacy and so much more. In many ways, these discussions have inspired a lot of the research which I have pursued throughout my PhD. In particular, Saurabh had a great vision on modeling arterial traffic, focusing on defining the appropriate level of model complexity given specific estimation, sensing or control goals, which I have followed and developed in my research.

Through his work on the *Path Inference Filter*, Timothy Hunter has made possible a lot of the numerical validation of my work. Timothy has spent countless hours developing, implementing, improving and running his algorithm on the millions of data points received in the *Mobile Millennium* system everyday, to turn noisy sparse GPS measurements into filtered path travel times which can be used for traffic estimation. Anyone who has worked, looked at or even imagined working with this type of data will understand the hard and valuable work of Timothy and realize the importance that it had in the completion of my work. Besides Timothy, the entire team of *Mobile Millennium* has enabled the technical support and the infrastructure to develop the numerical applications of my work. In particular, a lot of the results of this dissertation would not have been possible without the work of the team to set up and maintain the data feeds and the database, to develop the *Mobile Millennium* system and the road network abstraction.

The last thanks go to my friends, who I have met over the years, and have accompanied me everyday, giving advice, sharing thoughts, discussing both important and silly matters, and enjoying life. Friends have been my adopted American family and I am grateful to have met such wonderful people. I have met too many incredible people to acknowledge them individually here. I want to emphasize how much I care for their friendship, for all the moments that we have shared in the past years and the upcoming adventures which will keep on building our ties.

I still want to acknowledge a few people who have played such a wonderful role throughout my PhD. The magic of Craigslist lead me to meet Kristen Parrish and move in with her in January 2009. Even though we had barely met, Kristen warmly welcomed me and shared her personal feelings relating to losing a close family member. It was very important to have someone I could trust and confide in. Her energy and love of people was exactly what I needed at the time. I would also like to acknowledge Ana Ferreira who has been my closest friend throughout the past four years and who I hope to count as a friend in the years to come. Woody Hoburg has also been an incredible friend with whom I have been able to share a lot of personal matters, talk about awesome outdoors activities, discuss important life choices and have fun times at Mint Leaf happy hour! Finally, I am thankful for my friends in France to always be around when I come back. Keeping friendships with thousands of miles of distance is not easy and I truly appreciate these lasting friendships which are not affected by distance and time. I would not be able to go through a long PhD journey without feeling so much positive emotions, happiness and social support around me.

Chapter 1

Introduction

Queueing theory is the mathematical study of waiting lines, or queues. The field of queuing theory goes back to the early 1900s with the work of A. K. Erlang of the Copenhagen Telephone Company to model waiting times for calls in telecommunication networks [68, 69]. Since the 1950s, the field has received a lot of attention from the scientific community. In particular, the domains of application of queuing theory have expended from telecommunication networks to general communication networks, transportation engineering, air traffic control, manufacturing or supply chain management.

Each field of application comes with its specificities in terms of the characteristics of the queuing processes, the desired features of the outcome of the mathematical analysis, the precision of the modeling and so on. For example, air traffic control has important constraints in terms of safety and models must take into account the physical characteristics of aircraft (maneuverability, minimum and maximum speed). In supply chain management, one goal is to optimize the efficiency of the entire line of production while making sure that the process is robust if a production site or engine fails. In transportation networks, the field aims at reducing the external costs due to non-optimal operations [194]. An essential step for operations and planning (routing, network optimization) is to develop the ability to estimate and forecast traffic conditions with appropriate accuracy and reliability [37].

1.1 Related work

This section reviews prior work on queuing theory. Queuing theory often refers to the analysis of a single queue. When several queues co-exist and interact, one usually refers to a queuing network. The interaction between the queues requires the development of additional modeling and statistical results on top of queuing theory results. The complexity of queuing networks often leads to (domain-specific) approximations which aim at simplifying the model and make it more tractable and computationally efficient.

Background on queuing theory

In order to analyze and optimize queuing systems, one needs a mathematical model of the physical system and its properties, known as *queuing model*. A queuing model is a mathematical abstraction of the reality which is typically represented by: (i) the system's physical configuration which specifies the number and arrangement (*e.g.* queue capacities, queue disciplines, and so on) of the *servers*, which provide *service* to the *customers*, and (ii) the statistical properties of the arrival process and of the service process. The *queue capacity* refers to the maximum number of customers which can wait to be served in the queue. The *queue discipline* refers to the manner in which customers are selected for service when a queue has formed. There are several common queue disciplines:

- *First In First Out* (FIFO): the customers are served in the same order in which they have arrived. For this reason, it is sometimes also referred to as *First Come First Serve* (FCFS).
- *Last In First Out* (LIFO): the last customer to arrive will be the first one served, yielding another common denomination as *Last Come, First Served* (LCFS).
- *Service In Random Order* (SIRO) or *Random Selection for Service* (RSS): the customer to be served is chosen randomly, independently of the arrival times.
- *Priority*: customers with high priority are served first.

In the context of communication networks, each communication channel is a server and the messages are the customers. The (random) times at which messages request the use of the channel characterize the arrival process, and the (random) duration to use the channel and transmit the message constitute the service process. The queue capacity may be considered infinite and the queue discipline FIFO or Priority.

Urban transportation networks are another domain of interest, used as recurring example in the remainder chapters of the dissertation. Each driver (customer) seeks to use the transportation network (server) to go from an *origin* to a *destination* (service). In the latter queuing network, the queue capacity is defined by the number of vehicles which can fit on each road segment. The queue discipline is typically FIFO, even though some models may consider queues with priorities to model specific types of vehicles (ambulances, police vehicles and so on).

The mathematical analysis of the models aims at investigating how the physical and stochastic parameters of the system relate to certain performance measures, such as average waiting time, server utilization, throughput, probability of buffer overflow, etc. Applied queuing theory aims at developing models which are simple enough to yield to mathematical analysis, yet contain sufficient detail to reflect the behavior of the real system. This approach will remain a center component of the dissertation.

The characteristics of a queuing processes are typically defined using a notation defined by Kendall [133]. The process is described by three factors written A/S/c. Additional factors

may be used and the notation becomes A/S/c/K/D. The different factors have the following interpretation:

- A: Characteristics of the arrival process. Common denomination include Markovian (M) corresponding to Poisson arrival, Degenerate (D) corresponding to deterministic of fixed-time arrivals, Erlang (E_k) corresponding to arrivals with an Erlang distribution with shape parameter k or General (G) corresponding to arbitrary probability distribution of arrivals.
- S: Characteristics of the service process.
- c: Number of servers.
- K (optional): Capacity of the system. Once the capacity is reached, no more customers can enter the system. This factor is only mentioned when the capacity of the system is finite.
- D (optional): Queue discipline (usually not mentioned if the queue is FIFO).

Previous work has studied the properties of different queuing models including the M/D/1 and M/D/k queues [68, 69], the M/M/1 queue or the M/G/1 queue [186, 134]. The main results of queuing theory are out of the scope of this thesis. Please see [139, 48, 219, 93] for additional references on queuing theory.

Queuing networks

In many areas, such as manufacturing, transportation networks or task management (*e.g.* distributed computing), when a customer is serviced at a node, it can join another node and queue for service. Such a system of interacting queues is called a *queuing network*. The field of queuing networks is significantly more complex than the one of queuing theory with a single queue (even with several servers) because of high-dimensional interactions and dependencies.

One of the primary goals of queuing network theory is to estimate and predict the state of the network, given specific demand patterns. Statistical results aim at characterizing the robustness of the network, detecting potential bottlenecks which reduce the overall efficiency of the system or analyzing network equilibria. For a large class of networks, the policy which describes the sequence of nodes visited by a customer can be optimized. The optimization of the policy is commonly called a routing strategy.

The complexity of queuing networks benefits from specific assumptions which facilitate the analysis and understanding of the queuing processes. This thesis focuses on a class of queuing network which represent urban road networks. In numerous parts of the world, traffic congestion has a significant impact on economic activity. An essential step towards

active congestion control is the creation of accurate, reliable traffic monitoring systems. These transportation networks have the following specific characteristics.

- *Signalized queuing networks:* These queuing networks arise whenever two queues cannot be served concurrently and signals indicate which queue is active at a given time to manage the conflicting services. The urban (arterial) transportation network is one of the most intuitive example of signalized queuing network. Other examples include logistics and communication networks with interfering channels which cannot be used concurrently.

- *Horizontal queuing networks*: Queuing networks for which the amount of space of each customer is not negligible and *travel speed in a queue is finite*. This models the fact that once a customer is served, there is a non-null time before the next customer can be served, because it needs to "travel" to the head of the queue.

- *Networks with limited and/or uncertain information*: Queuing networks for which there is little and uncertain amount of information and measurements available, both on the characteristics of the queuing network (service rates, arrival rates, sequence of service requested by a customer) and on the state of the different queues.

Historically, traffic monitoring systems have been mostly limited to highways and have relied on public or private data feeds from dedicated sensing infrastructure:

- *Loop detectors* or *inductive loops* [124] are embedded into the roadway and detect vehicles as they pass over the detector. A properly calibrated loop detector provides high-accuracy flow and occupancy data as well as velocity when two detectors are placed close together (double-loop detectors). The sensors suffer from important reliability issues requiring filtering to produce quality input data to traffic estimation algorithms. Loop detectors are commonly found on most major highways throughout the United States and Europe where they have communication capabilities to transmit the data to a central server in real-time (that can subsequently be used in traffic information systems). In the United States, most loop detectors installed on arterial roads do not have internet connection, preventing their use for arterial estimation. Rather, this data is generally used locally for signal timing control.

- *Radars* can be placed on poles along the side of the road enabling them to collect flow, occupancy and velocity data. Their deployment remains limited.

- *High-resolution video camera* placed high above the roadway track all vehicles within the view of the camera. As of the time when this thesis is written, they do not provide data in real-time due to the large amount of post-processing work that needs to be done on the images to turn them into actual vehicle trajectory data. The cost of deployment and processing limit the scale of their use to small spatio-temporal domain (in the order of one mile stretch for fifteen minutes) to validate modeling assumptions and estimation capabilities.

- *License plate readers* automatically extract the license plate identification from passing vehicles. They are generally used in pairs along the road to extract high-accuracy travel times for vehicles passing both locations. The deployment of these sensors require the identification of appropriate locations to place them and often remains limited to specific data collection studies.

- *Radio-Frequency Identification* (RFID) and *Bluetooth* readers can be used for traffic data collection by placing readers at various points along the roadway. Travel times can be collected between pairs of points and processed similarly as license plate readers data. The accuracy of travel times varies depending on the strength of the signal: stronger signals increase the chance of detection but increase the duration and area of detection, leading to a loss in accuracy especially for short distance travel times. RFID readers are generally placed far apart from each other in current deployments, making them useful for collecting long distance travel time information, but not for providing input data to detailed traffic estimation algorithms. They are placed almost exclusively on highways, making it uncommon to find this technology on arterial roads. The density of the arterial network and the high number of possible routes and itineraries decreases the probability to detect a specific vehicle at two distant readers, unless the entire network is equipped with such a technology.

- *Wireless sensors* are devices embedded into the roadway. They are similar to loop detectors but record the magnetic signature of vehicles passing them which is used for vehicle re-identification at downstream sensors with up to 80% accuracy [98]. Besides flow and occupancy, wireless sensors provide travel times between pairs of sensors for all the matched vehicles. The wireless sensors are cheaper to deploy and maintain than loop detectors. They provide travel times for a larger portion of the flow and with higher accuracy than Bluetooth readers and RFID readers. These characteristics make them appealing for large-scale deployments on arterials even though they are only available in a small number of locations at the current time and monitor specific routes rather than portions of a network. *Sensys Networks* [3] is currently one of the leading providers of these sensors.

Urban networks come with additional challenges:

- The underlying *flow physics* which governs them is more complex and highly variable (traffic lights with unknown cycles, turn movements, pedestrian traffic)

- The traffic estimation relies mostly on *probe vehicle data*, which comes from various sources, each with their own specific issues (sparsity, bias, noise, coverage):

  - *Fleet data* (FedEx, UPS, taxis, etc.) provides information from one minute sampled GPS data (the current standard in the United States) but with specific spatio-temporal travel patterns (fleets avoid congestion).

- *Participatory sensing* (GPS enabled smartphone or aftermarket device data or 2-way navigation device), for example Garmin, INRIX, Microsoft, Google, Apple, Nokia or Waze. This data is unpredictable, sparse, and no single company has ubiquitous coverage.
- *Vehicle re-identification* (*e.g.* RFID, magnetic signature [147], Bluetooth readers, Automated Plate Recognition Cameras) is also used for traffic monitoring, with deployment of readers along some small portion of the transportation network. Wireless technology provides travel time measurements of a high proportion of the flow of vehicles [147] through vehicle magnetic signature re-identification. This information remains limited to the equipped road which represents, today, a marginal fraction of the arterial network.

The next paragraphs describe different classes of models and algorithms which can be used to turn traffic data into reliable traffic information.

Models for highway traffic

Even though the highway network is not signalized, models of traffic flow on highway networks have a lot of influence on current research in signalized networks. This motivates a short overview of the state of the art of highway traffic models. For highway networks, it has become common practice to perform both system identification of highway parameters (free flow speed, traffic jam density and flow capacity) and estimation of traffic state (flow, density, length of queues, bulk speed and shockwave location) at a fine spatio-temporal scale [220, 25]. These approaches heavily rely upon both the availability of data and highway traffic flow models developed over the last half century [155, 189, 52]. They use sequential data assimilation algorithms (Kalman filtering [202] or other analogous techniques) to transform the available data into usable traffic information (see [220, 206, 117, 144] for a discussion specific to highways). Proof-of-concept studies have demonstrated the feasibility of designing highway traffic monitoring systems relying on probe data only [102, 220, 206].

Microscopic models

Microscopic models of traffic characterize the dynamics of every vehicle in the network and its interaction with the infrastructure and with the other vehicles. The state of the network encompasses microscopic properties like the position and velocity of single vehicles. For a network with N vehicles, the dimension of the state is thus $\mathcal{O}(N)$, regardless of the size of the network. There are at least two main classes of microscopic models:

- *Car following models:* Ordinary differential equations describe the dynamics of the positions of vehicles depending on the position of other vehicles and network attributes. Historically, car following models have assumed that the dynamics of a vehicle only depends on its own velocity, on the distance to the preceding vehicle and the speed of that vehicle. More general models have been developed to account for additional

aspects of vehicle dynamics. In particular, driving behavior may not only depend on the leading vehicle but on a higher order of preceding vehicles. Some examples of car following models are developed in [89, 125, 210].

- *Cellular automaton*: The time and space are discretized and the model describes how the state of each section of the network (cell) is updated at each time interval. Each road section can either be occupied by a vehicle or empty. The time scale is typically given by the reaction time of a human driver. The length of the cell determines the granularity of the model. Cellular automata are not able to model dynamics as accurately as car following models, but they are simpler and more efficient numerically and can thus be used to model larger networks.

Both car following models and cellular automata have limitations due to the dimensionality of the problem, which makes these methods challenging for any reasonably sized networks. Moreover, these models are very sensitive to calibration and require large amounts of site specific data which is rarely available at a large scale. They are often used for simulation softwares such as *PARAMICS* [35], *CORSIM* [95] or *VISSIM* [75].

Macroscopic models

Vehicular flow is represented as a continuum and characterized by macroscopic variables, often chosen to be *flow* $q(x,t)$ (veh/s), *density* $\rho(x,t)$ (veh/m) and *velocity* $v(x,t)$ (m/s). The dynamics is characterized by partial differential equations, such as the *Lighthill-Whitham-Richards model* [155, 189] or second order models [184, 182, 222, 149] which gained popularity and generated some debate within the transportation community [55, 13]. Third order and higher order models [100], as well as phase transition models [27, 45] were also developed to capture some specificities of vehicular traffic. Estimation and control of partial differential equation is an entire fields reviewed in Chapter 2.

Estimating the state of the queuing network at any location x and time t requires large amount of data on the arrival rates (arrival of vehicles in the network) and service rates (capacity of each road segment, precise signal timing and so on). Some of these characteristics are site specific and require calibration [86, 197]. Other approaches do not require as much information about the network but are not practical given the current penetration rates of probe vehicles [16]. Moreover, these methods do not characterize the *probability distribution function* (pdf) of travel times.

Vertical queuing theory

In the context of transportation networks, queuing theory, as described at the beginning of the chapter is often referred to as *vertical queuing theory* or *point-delay models*. It has been applied specifically to arterial traffic since the pioneering work of Webster in the 1950s [218, 6, 212, 152]. These contributions have studied the effect of different arrival distributions on the average delay at a signal. Some work recovered the probability distribution of delay or

of number of vehicles in the queue using analytical derivations and simulations. Results of vertical queuing theory have successfully been applied to planning applications (*e.g.* signal plans) but have limited real-time applications, as shown by [211]. Initial approaches to generalize the derivations for a network and model congestion propagation can be found in [179]. Vertical queuing theory does not model how the queue grows in space and considers that the delayed elements stack up upon one another, incurring no delay traveling to the point of congestion. This theory is well suited to model packets of data, (computer) tasks or communication of messages. However, when it comes to vehicles, the delay to travel to the point of congestion is not negligible.

Beyond estimation, vertical queuing theory has also been applied for control strategies of traffic signals [215]. There has also been significant interest to characterize Nash equilibria in both static [22] and dynamic [160] settings. Nash equilibria of congestion games are inefficient (price of anarchy [181, 40]) compared to the system optimum, in which a coordinator assigns flow as to minimize a system-wide cost function. In order to address this inefficiency, some tools have been proposed, including congestion pricing [180], capacity allocation [143] and Stackelberg routing [191, 10]. However, vertical queues show the same limitations for defining control strategies as they do for estimation; recent research takes into account the specificities of horizontal queues in the design of control policies [146, 145].

Horizontal queuing theory

To overcome the limitation of vertical queues,the work of [198] and [165] developed a *horizontal queuing theory*, which models how queues form and release in the physical space. This theory serves at the basis for the derivations presented in this dissertation (Chapters 5 and 6). It has been used by [176] and [224, 223] to model the probability distribution of delay on arterial links. Other work studies the influence of the stochasticity of overflow queues [216] on the probability distribution of travel times. These approaches assume that link travel times are available. However, the main source of data for urban traffic estimation comes from sparsely sampled probe vehicles which typically report their position at a given temporal rate (*e.g.* once per minute). The reported locations do not coincide with the network discretization, requiring a more general approach. Another line of research [46] estimates the pdf of queue length from probe vehicle data, assuming that vehicles report their position when they join the queue. This very interesting sampling scheme is not yet the standard among probe vehicles limiting the possibility to use such an approach for global monitoring systems.

Data driven models

The variability of traffic has also led to the development of data driven models, which do not directly model the physics but have the prospect to be more flexible, more scalable and to have results which improve as the amount of available data increases. Neural networks and state-space neural networks [214, 157], graphical networks (Bayesian networks and Markov

Random Fields) [183, 201, 82], regression techniques and time series analysis [87, 106] have been introduced to produce short-term traffic predictions for both freeway and arterial traffic with promising results. These articles model the spatio-temporal dependencies of the links of the network which provides more robustness when little or no data is available on some parts of the network. However, none of these articles present a comprehensive modeling approach of arterial traffic flow, which ensures physically realistic estimates when little or no data is available.

1.2 Problem statement

Section 1.1 emphasized the importance to study queuing networks for a wide variety of applications. Different applications come with specific challenges such as modeling, available information regarding both the characteristics of the network and the demand and service rates, availability of measurements of the state of the queuing network, desired outcome of the analysis of the network (estimation, control, failure detection, and so on).

Urban transportation networks have received a lot of attention in the recent past with the emergence of location aware, communication capable mobile devices (*e.g.* GPS enabled smart-phones, fleet management devices). By sharing their location, devices provide sparse measurements of the state of the network. Gathering the information from a large number of devices in a *community sensing* or *participatory sensing* paradigm [144, 70] offers new opportunities for traffic estimation, forecast and network optimization in urban environments.

Challenges of location data in urban networks

The location data sent by the mobile devices is referred to as *probe vehicle data* or *floating car data*. For privacy reasons, communication costs or battery life management considerations, the main source of data with the prospect of global coverage in the near future comes from *sparsely sampled* probe vehicles. In this paradigm, each vehicle reports its location at a low frequency; the industry standard is one location report per minute at the time this thesis was written. This fact has several consequences on the process of turning the measurements into valuable information:

- *Map-matching and Path-inference*: The GPS measurements may be noisy and must be mapped onto the road network. Moreover, the vehicle may travel more than one link between successive measurements, and the path effectively followed by the vehicle between successive measurements needs to be reconstructed. These problems can be addressed simultaneously using a *map-matching and path-inference* algorithm [120] which combines models of GPS noise and driving behavior in a Markov Random Field to reconstruct filtered trajectories between successive location reports. The algorithm returns information on the path followed by the probe vehicle and the travel time between the successive location reports. The information is represented as a tuple with the following information:

Figure 1.1: San Francisco taxi measurement locations, observed at a rate of once per minute. Each small dot represents the measurement of the location of a taxi, received between midnight and 7:00am, on March 29th, 2010. The large dots represent the location of taxis visible in the system at 7:00am on that day.

TODO: Add Tim's Ieee t-its if published by May

  - `List of links`: list of links traversed by the probe vehicle between the two successive location reports.
  - `Start offset`: (mapped) distance of the first GPS point to the upstream intersection.
  - `End offset`: (mapped) distance of the second GPS point to the upstream intersection.
  - `Start time`: Time at which the first GPS point is sent.
  - `Travel time`: Difference between the time when the second and the first GPS points are sent.

- *Travel time on partial links*: When vehicles report their location with a given frequency, the location reports do not coincide with the discretization of the network. The sampling frequency is too low to interpolate the travel time on the missing portions of the link, in particular because of the spatial heterogeneity of travel times on a link (vehicles are more likely to stop close to intersections because of the presence of signals).

- *Path travel time decomposition*: Because of the low sampling frequency, vehicles typically traverse several (partial) links on their path between successive location reports. Numerous algorithms rely on link travel time measurements [106, 166] to infer (and predict) the traffic conditions on the road network. These algorithms require that travel times of individual links be computed from the path travel times of the probe vehicles. This computation is called *travel time allocation* or *travel time decomposition* [101]

Challenges of queuing network modeling and estimation

The underlying processes of queuing networks is in general very complex. Models are required as a mathematical abstraction of the reality. They are necessary to make estimates and predictions. One important challenge of mathematical modeling is to find an appropriate trade-off between simplicity and accuracy of the model. Added complexity usually improves the features that a model can integrate, but it can decrease one's capacity to understand the behavior of the model, interpret and analyze results. It may also raise computational problems, including intractability, numerical instability and over-fitting. The choice of model and assumptions made depend on the setting in which the model is used. For example, Newton's classical mechanics is an approximate model of the real world. The model is sufficient for a wide range of applications. However, specific applications require a more precise model such as Quantum physics or Relativity theory whenever particle speeds are no longer well below the speed of light, or the system of interest does not consist of macro-particles only.

Similar challenges arise in queuing networks. As detailed in Section 1.1, previous work has investigated a wide range of models to represent queuing networks. In particular for urban traffic, models range from microscopic models to fully data driven models. On the one side, microscopic models have the potential to fit reality accurately. They prohibitive computational complexity and sensitivity to calibration of numerous parameters limit their applicability for large scale traffic estimation. On the other side, fully data driven models have the highest flexibility and the potential to perform very accurately with large amounts of training data. They do not provide guarantees regarding the realism of the estimates, which is problematic when only little and noisy data is available.

Problem statement

In light of the challenges and characteristics of the modeling and available data, this thesis analyzes the following question: *How can one leverage the realism and insights of accurate physical models while offering the flexibility and learning capability of data-driven models in queuing networks?* The thesis investigates the trade-off between simplicity and accuracy of the model for queuing networks with limited information on the specific parameters of the queue dynamics, on the parameters of the demand and with sparse measurements of the state of the network.

The thesis takes the recurring example of signalized flow networks which exhibit some specificities which require interesting modeling considerations. However, derivations are in general valid for other types of queuing networks, distributed parameter systems or dynamical systems.

1.3 Organization of the thesis and contributions

This thesis is organized as follows.

Chapter 1 reviewed existing work in queuing theory and queuing network analysis. The chapter emphasizes the variety of applications for queuing networks and exhibits some remaining challenges which remain to be solved. In particular, the chapter demonstrates the potential of probe vehicle data for large scale estimation in urban networks. The data and the modeling come with specific challenges which are investigated in the dissertation, with a focus on leveraging the potentials of both physical and data-driven models in an integrative approach.

A common approach to modeling systems governed by conservation laws leverages the theory of *distributed parameter systems*. Chapter 2 reviews existing work on distributed parameter systems which is relevant to estimation in systems governed by conservation laws. In particular, the chapter reviews some results of *Hamilton-Jacobi* equations which are extended in the following chapters.

Chapter 3 makes the assumption that queuing networks are accurately described as a distributed parameter system based on conservation laws. More specifically, the chapter investigates signalized queues for which the parameters of the signals (times when servers offer service or not) are unknown and only partial measurements are provided for the trajectory of the customers in the queue.
Contribution: The chapter formalizes the problem of estimating the parameters of the signals as a boundary condition problem for a Hamilton-Jacobi partial differential equation. The chapter derives an algorithm which exhibits a specific solution to the problem or shows that no solution exists. If a solution exists, it may not be unique but the algorithm computes the solution which has the physical characteristics required by the problem of interest.
Publication [109]: "Reconstruction of boundary conditions from internal conditions using viability theory", A. Hofleitner, C. Claudel, A. Bayen, 2012 American Control Conference, pp.640-645, June 2012.

For many applications, both the differences between the modeling and the reality on one side and the inaccuracies in the measurements of the system on the other side must be accounted for. This is typically done by doing robust modeling or by using a statistical model. Robust modeling usually provides bounds of values for parameters or state estimates given the modeled discrepancy between the model and the reality on one side and between the measurements and the state of the system on the other side. Statistical modeling considers the state of the system and its parameters as random variables and computes probability distributions over these variables. Chapter 4 uses statistical modeling to take into account the inaccuracies in the demand for service in a queue amd compute the probability distribution of the state of the queue at any point in time and space.

Contribution: The chapter extends existing work on Hamilton-Jacobi partial differential equations and viability theory by introducing randomness in the boundary conditions and characterizing the probability distribution of the solution at any point in time and space.
Publication [108]: "Probabilistic formulation of estimation problems for a class of Hamilton-Jacobi equations", A. Hofleitner, C. Claudel and A. Bayen, 51st IEEE Conference on Decision and Control, pp. 3531-3537, December 2012.

Under limited measurements, it is not realistic to expect to reconstruct the state of the network distribution at every location and time. In signalized networks, one can exploit the periodicity imposed on the system by the presence of signals (granted the parameters of the signals and the demands are stationary) to aggregate the dynamics over time and describe the average dynamics per cycle in terms of probability distributions. Chapter 5 follows this approach to characterize the probability distribution of delays and travel times between any location in signalized queues and estimate the parameters of the distributions from sparsely sampled probe data.
Contribution: The chapter leverages results from horizontal queuing theory to derive the probability distribution of travel time between any location on the network, making it adapted to measurements which include *partial* links, as mentioned in Section 1.2 . The chapter proves that the distributions are mixture of log-concave distributions. The property is used to formulate the *travel time decomposition* problem as a *Mixed Integer-Convex* problem and propose algorithms which exploit this property. The parameters of the travel time distributions are estimated independently for each link of the network as the solutions of small scale *Maximum Likelihood* problems.
Publications [114, 107]: "Probability distributions of travel times on arterial networks: a traffic flow and horizontal queuing theory approach", A. Hofleitner, R. Herring and A. Bayen, 91st Transportation Research Board Annual Meeting, Number 12-0798, Washington D.C., January 2012.
"Optimal decomposition of travel times measured by probe vehicles using a statistical traffic flow model", A. Hofleitner, A. Bayen, 14th IEEE Intelligent Transportation System Conference (ITSC 2011), pp. 815-821, Washington D.C., October, 2011.

Chapter 6 extends the statistical model of Chapter 5 by modeling the dynamics of customers as they switch queues. In Chapter 5, even though measurements span several links and cover the entire network, each queue is modeled independently and the distributions are estimated given the measurements allocated to the link. Chapter 6 extends the derivations to model a queuing network which models the propagation of congestion.
Contribution: The chapter builds upon the derivations from Chapter 5 to model queuing in urban networks as a parametric *Dynamic Bayesian Network*. The chapter investigates algorithms to learn the parameters of the Bayesian network and perform real-time estimation.
Publication [113]: "Arterial travel time forecast with streaming data: a hybrid approach of flow modeling and machine learning", A. Hofleitner, R. Herring and A. Bayen, Transportation Research Part B Vol. 46 Number 9, pp 1097-1122, November 2012.

Chapters 5 and 6 rely on assumptions on the dynamics of horizontal queues and dynamics of customers as they switch queues to provide an analytical derivation of the probability distribution of travel times. The underlying physical model ensures realistic estimates when little data is available. However, the physical approach limits the flexibility of the model when the underlying assumptions are not met. Moreover, simpler distributions such as Normal distributions have several properties which make the computations more efficient. Chapter 7 proposes a model which builds on some ideas from Chapter 6 regarding the propagation of congestion in a network but simplifies the dynamics and the distribution of travel times to improve the computational complexity and the generality of the model.
Contribution: The chapter presents a *Dynamic Bayesian Network* to model the dynamics of congestion in a queuing network. By releasing some assumptions from Chapters 5 and 6, the resulting model can be applied to a larger variety of applications.
Publication [112]: "Learning the dynamics of arterial traffic from probe data using a Dynamic Bayesian Network", A. Hofleitner, R. Herring, P. Abbeel and A. Bayen, IEEE Transactions on Intelligent Transportation Systems, Vol. 13, pp. 1679 -1693, December 2012.

The statistical models of the network dynamics presented in Chapters 6 and 7 rely on an arbitrary time discretization to update the state of the network. However, having a fixed time discretization may be limiting when conditions change rapidly (causing delays and inaccuracies in the estimation). Similarly, one may benefit from increasing the duration of the time discretization for links with stationary conditions, in particular if they receive a limited amount of measurements. Besides the time discretization, the models rely on assumptions on the conditional independence between the congestion levels of different links of the network. It is intuitive to assume that congestion first spreads locally. However, understanding more accurately the dependency between neighboring links may improve the interpretability of the results. The chapter aims at improving the real-time estimation capabilities of dynamical models such as the ones presented in Chapters 6 and 7. It uses an online data-driven approach to detect changes in the state of the network (either spatially or temporally).
Contribution: The chapter derives an algorithm to solve a generalization of the LASSO. The solution is updated as new measurements become available. The generalization of the LASSO allows to impose sparsity on a linear function of the solution (to detect spatial changes for example) or on the difference between successive estimates (to detect temporal changes).
Publication [111]: "Online least-squares estimation of time varying systems with sparse temporal evolution and application to traffic estimation, A. Hofleitner, L. El Ghaoui, A. Bayen, 50th IEEE Conference on Decision and Control and European Control Conference, pp. 2595-2601, Atlanta Fl., December 2011".

Both the modeling and the interpretability of the results can be improved by looking at the network at a large scale and identifying specific patterns of the dynamics. Chapter 9 proposes a data-driven approach to identify and analyze both spatial and temporal patterns

in the dynamics of urban networks. It identifies times of day and days of the week with similar behavior as well as links of the network which tend to follow similar congestion patterns. The outcome of such an analysis has the potential to improve dynamical models such as the ones presented in Chapters 6 and 7: (i) The analysis identifies regions of the network which have independent dynamics. These natural cuts can lead to considerable gains in the computational complexity of these models by using approximate inference algorithms [30] or by reducing the number of particles required to accurately represent probability distributions over the sub-network (ii) The algorithm clusters times of day, days of week and/or links of the network with similar dynamics. This outcome can be used to increase the robustness of the estimation using hierarchical models.

Contribution: The chapter uses a *Dimensionality Reduction* algorithm known as *Non-negative Matrix Factorization* (NMF) to perform large scale analysis of the congestion levels of a network over several months. It analyzes the dynamics of the network in the lower dimensional space to identify clusters of links with similar dynamics and to define periods of the day during which conditions are expected to remain stationary. It also uses hierarchical clustering based on a *cosine distance* to identify similarities between the days of the week.

Publication [115]: "Large scale estimation of arterial traffic and structural analysis of traffic patterns using probe vehicles", A. Hofleitner, R. Herring, A. Bayen, Y. Han, F. Moutarde and A. de La Fortelle, 91st Transportation Research Board Annual Meeting, Number 12-0598, Washington D.C., January 2012.

Chapter 2

Background on distributed parameter systems

As explained in Chapter 1, the thesis analyzes the trade-off between model simplicity and capacity to integrate important features. It investigates how to leverage models derived from the physical properties of the system and information provided by the measurements. This chapter reviews existing results which constitute a basis for further analysis of numerous physical systems: the class of *distributed parameter systems*.

A distributed parameter system is a system whose state space is infinite-dimensional (also known as infinite-dimensional systems). Distributed parameter systems include systems descibed by

- *Partial differential equations* (PDEs) [17]
- *Infinite dimensional vector systems* [59, 199]

is usually described by a function of continuous variables (space and time, multi-dimensional spaces) in contrast to a finite dimensional vector. Typical examples are systems described by *partial differential equations* (PDEs). PDEs provide an efficient way of representing physical phenomena in a mathematically compact manner: they relate derivatives of a function with respect to different variables [71]. Numerous examples can be found in fluid mechanics, continuum mechanics or studies of diffusion phenomena.

TODO: Figure out what the actual definition of distributed parameter system is

To compute the solution of the physical problem of interest, two types of information are traditionally needed:

- *Initial conditions* They represent the value of the function at an *initial* time. For example if the equation characterizes the evolution of the temperature of a beam, the initial condition is the temperature distribution in the beam at the beginning of the experiment. Sometimes, terminal conditions (instead of initial) are prescribed; for example to impose the state of the system at the end of an experiment.

- *Boundary conditions.* They represent known information at the *spatial boundaries* of the system. For example, if the PDE represents the evolution of the velocity of vehicles on a road segment, the boundary condition may be given by a radar at the entrance and exit of the road segment.

Besides initial and boundary conditions, it is desirable to provide *internal value conditions*. *Internal value conditions* represent known information on the solution in the interior of the domain of definition. For example, measurements from Lagrangian sensors are *internal value conditions*. The integration of *internal value conditions* requires a specific mathematical treatment of the solution as described in Section 2.3 and in Chapters 3 and 4.

Given a partial differential equation, initial and boundary conditions, two main theoretical questions arise:

- *Existence of a solution*: Prove that there exists (at least) one function satisfying the PDE, the boundary and initial conditions. If no function satisfies both the PDE, the boundary and initial conditions, the conditions are *incompatible* and the problem is said to be ill posed.

- *Uniqueness of the solution*: There may be several functions satisfying both the PDE, the boundary and initial conditions. However, even if the solution is not unique, there is sometimes only one of the mathematical solutions which represents the actual evolution of the physical phenomenon of interest. Discriminating between several solutions to find the proper solution to the physical problem is sometimes very difficult, and for specific problems an open question. This might require the use of a selection criterion, which is often inspired by physical principles.

This chapter is organized as follows. Section 2.1 reviews existing methods of estimation and control of partial differential equations. Section 2.2 presents an example of *distributed parameter system* for networks governed by conservation laws (such as transportation networks or flow networks). Section 2.3 reviews results from viability theory, a powerful framework for fast and exact semi-analytic estimation of scalar Hamilton-Jacobi partial differential equations. The results from Section 2.3 are extended in Chapters 3 and 4.

2.1 Data assimilation in distributed parameter systems

The problem of combining observation data (measurements) with the underlying dynamical principles governing the system under observation is called data assimilation or state estimation. It consists in incorporating data in the mathematical model of the physical system (*i.e.* represented by a partial differential equation), in order to estimate the current state of the system and forecast its future state [154, 23]. State estimation and control for PDE-based systems is more complex than for their ordinary differential equation (ODE)

based counterparts, because of the distributed nature of the state. Existing state estimation methods are detailed in [196] and summarized below. They often come from estimation and control theory as well as Bayesian statistics. State estimation is sometimes referred to as *optimal* state estimation in reference with a chosen criterion. Most common criteria include least-squares, maximum likelihood or minimax. The choice of criteria may depend on the application of interest as well as characteristics of the system (*e.g.* multi-modal).

Estimation theory encompasses theories used to estimate the state of a system by combining, sometimes with a statistical approach, all available reliable knowledge of the system including measurements and theoretical models. The a priori hypotheses and choice of estimation criterion are crucial in the estimation process since they determine the influence of dynamics and data on the state estimate.

Estimation of distributed parameter systems

At the heart of estimation theory is the scheme derived by Kalman in 1960 known as the *Kalman Filter* [129]. The *Kalman Filter* is a simplification of Bayesian estimation which was originally developed for the case of linear ordinary differential equations. The *Kalman Filter* provides a sequential, unbiased, minimum error variance estimate based upon a linear combination of all past measurements and dynamics. The *Kalman Smoother* is another unbiased, minimum error variance estimate for linear systems. It solves a variant of the estimation problem known as a *smoothing problem*: the computation of each state estimate uses all the data available, before and after the time of estimation. A common version of this scheme first computes the *Kalman Filter* estimate. The smoothing is then carried out by propagating the future data information backward in time, correcting the *Kalman Filter* estimate using error covariance and adjoint dynamical transition matrices. This implies that both the *Kalman Filter* state and error covariances need to be stored at all data-correction times, which is usually demanding on memory resources.

The main limitation of the *Kalman Filter* are its restriction to linear models with additive independent white noise in both the transition and the measurement systems. For nonlinear systems and systems for which the uncertainty is not accurately modeled by additive independent white noise, a series of approximate or suboptimal estimation schemes have been derived and employed for numerous applications. *Extended Kalman Filtering* (EKF) is a modification of *Kalman filtering* for nonlinear systems which are differentiable. EKF techniques have been applied, among others, to water channels [72] and traffic flow [7, 217]. However *Extended Kalman Filtering* performs poorly for specific nonlinear systems. In particular it is not defined at points of discontinuities of non-smooth or non-differentiable systems [26].

Ensemble Kalman Filtering (EnKF) [74] is a Monte-Carlo based method which overcomes the limitation of EKF: it does not require approximating the model by linearization around the current estimate as done in EKF, which is crucial for non-smooth systems. EnKF has been applied to traffic estimation [220], Shallow Water Equations [208] or meteorology [118]. The EnKF samples the possible current states of the system according to a probability distribution, computes the evolution of these samples, and combine them with new measurements

to obtain the best estimate of the state. The *Mobile Millennium* system [4] is an example of operational implementation of the EnKF for traffic flow estimation.

More generally, the state of distributed parameter systems can be estimated using *Monte Carlo Simulation*. A common implementation is *Particle Filtering* (PF), which can be used for general nonlinear systems, albeit with a higher computational cost [9]. A more extended review of particle filters is presented in Chapter 6.

Other estimation algorithms have been developed, relying on specific assumptions on the model and/or the measurements to limit the computational cost of the estimation. *Direct Insertion* consists of replacing the forecast values by the observed ones, at all data points where data is available. The a priori statistical hypothesis is that data are exact. The *Blending estimate* is a scalar linear combination of the forecast and data values at all data points, with user-assigned weights. The *Nudging* or *Newtonian Relaxation Scheme* "relaxes" the dynamical model towards the observations by adding a non-physical diffusive-type term to the model equations which depends on the difference between the observations and the model solution [103].

Optimal interpolation [164] is a simplification of the *Kalman Filter*. The data-forecast melding or analysis step is still a linear combination of the dynamical forecast with the data residuals, but in the *Optimal interpolation* [32] scheme, the matrix weighting these residuals or gain matrix is empirically assigned instead of being computed and updated internally. The *method of Successive Corrections*, instead of correcting the forecast only once as in previous methods, performs multiple but simplified linear combination of the data and forecast. Conditions for convergence to the *Kalman Filter* have been derived, but in practice only two or three iterations are often performed.

Estimation based on control theory and optimization

Estimation based on control theory or variational approaches usually perform a global time-space adjustment of the model solution to all observations and thus solve a smoothing problem. The goal is to minimize a cost function (*e.g.* least-squares) penalizing the distance between the data and the estimates, with the constraints of the model equations and their parameters.

A popular optimization framework is the *Adjoint method*, which provides an efficient way to compute the gradient of a system under the constraints that the solution satisfies the dynamical model. For example, consider the optimization problem where

- The cost function is the sum of two penalties: one penalty weights the uncertainties in the initial conditions, boundary conditions and parameters with their respective a priori error covariances, the other is the sum over time of all data-model misfits at observation locations, weighted by measurement error covariances.

- The constraint is the dynamical model of the system

Then the adjoint method provides an efficient way to compute the gradient of the cost function under the dynamical constraints, which allows the use of descent methods to solve the problem.

Generalized Inverse Problems [205] generalize adjoint methods by allowing the model equation to not be satisfied everywhere. The best fit is often defined in a least-squares sense: the penalty to be minimized is similar to the one used in adjoint methods, except that a third term is added to account for model discrepancies weighted by a priori model error covariances. The *Representer Method* [73] is an algorithm for solving generalized inverse problems. *Data reconciliation* algorithms use information redundancy to handle measurement errors and model inaccuracies [43, 221].

Spectral methods [31, 28] use modal decomposition techniques to transform partial differential equations into ordinary differential equation in the frequency domain. The transformation to the frequency domain leads to an inverse modeling problem which is easier to solve than the original PDE. The transformation may require the linearization of the PDE, with the common limitation involved for non-smooth equations.

When Kalman-based or adjoint-based estimation is not applicable, most estimation problems still take the form of the optimization of a cost function. They are solved using optimization algorithms which may be chosen depending on the specific problem of interest including: (i) the dimension of the problem: Newton or quasi Newton (*e.g.* DFP, BFGS, SR-1)-based methods converge faster than gradient or sub-gradient based methods but are too computationally demanding to handle large scale problems and are hard to parallelize, (ii) convexity properties: non-convex problems require specific algorithms (simulated annealing, genetic algorithms, descent algorithms with random starts) to increase the chances to find global optima.

Hamilton-Jacobi equations

In one dimensional systems, hyperbolic scalar conservation laws (such as the ones used to describe flow networks) have a direct counterpart in *Hamilton-Jacobi* (HJ) theory [71]. A *Hamilton-Jacobi* equation is a first-order, non-linear partial differential equation, which can be formulated as follows.

$$\frac{\partial S}{\partial t} + H = 0,$$

where H is the Hamiltonian function. The solutions S of *Hamilton-Jacobi* equations typically satisfy the equation in a *generalized* sense (in the sense of distributions or set valued analysis) and are called weak solutions. *Viscosity solutions* [50, 49] were the first class of weak solutions identified for HJ PDEs for which existence and uniqueness could be proved [50, 49]. They are considered as the "physical" solutions in many applications of PDEs. The solutions are continuous, but not necessarily differentiable everywhere. The concept of viscosity solutions has been extended to non-smooth solutions (sub- and super- solutions), see in particular [20] and [122] for traffic. *Barron- Jensen/Frankowska* (BJ-F) solutions [21, 79]

generalize the concept of viscosity solutions by allowing the solution to be semi-continuous. See in particular[11] in the context of traffic.

The solutions to HJ PDEs (and their conservation laws counterparts) can be computed numerically using various methods:

- Finite difference schemes such as Godunov [90] or Lax-Friedrichs [131]. Finite difference methods approximate the PDE as a finite difference equation on a computational grid. To ensure convergence of numerical scheme, the grid is often constrained by stability conditions, such as the *Courant-Friedrichs-Levy* (CFL) condition [153].

- Wave-front tracking methods [51, 33] rely on the structure of the mathematical solutions to hyperbolic conservation laws, which feature shockwaves and expansion waves. They first compute the location of these waves, and then derive the expression of the solution everywhere.

- Level set methods [168] rely on finite difference schemes to numerically approximate the solution with subgrid accuracy and avoid the high cost of grid refinement. They can be extended in some cases by fast marching methods [195], which are computationally efficient.

- Lax-Hopf formula, as detailed further in the chapter, requires the resolution of minimization problem using dynamic programmic [71] or the Lax-Hopf algorithm [148].

The solutions used in the dissertation are obtained using a Lax-Hopf formula, which expresses the solution at any given point as a minimization (or maximization) problem. The derivations rely on previous work [41, 42], which is summarized in Section 2.3.

2.2 Traffic flow theory

A special class of partial differential equation is typically used to model the dynamics of traffic flows on transportation networks. They are usually called *macroscopic models* because they describe the dynamics of traffic flow using macroscopic variables (flow, velocity and density) rather than at the individual vehicle level. The thesis focuses on the *Lighthill-Whitham- Richards* [155, 189] (LWR) model, which is a first order macroscopic flow model commonly used in transportation engineering. As seen in Chapter 1, there exist higher order macroscopic models [55, 13] which capture additional features of the traffic dynamics.

This section describes the *Lighthill-Whitham-Richards* model and its integral formulation as a Hamilton-Jacobi equation, known as the *Moskowitz* equation.

The Lighthill-Whitham-Richards model

In macroscopic traffic modeling, vehicular flow is represented as a continuum and characterized by macroscopic variables of *flow* $q(x,t)$ (veh/s), *density* $\rho(x,t)$ (veh/m) and *velocity*

$v(x,t)$ (m/s). The *Lighthill-Whitham-Richards* (LWR) model [155, 189] is a first order model obtained from the conservation of vehicles and an empirical relation between flow and density. The LWR model is formulated as a hyperbolic partial differential equation as follows:

$$\frac{\partial \rho(t,x)}{\partial t} + \frac{\partial \psi(\rho(t,x))}{\partial x} \;=\; 0 \tag{2.1}$$

The flow-density relation $\rho \mapsto \psi(\rho)$ is known as the *flux function* or the *fundamental diagram*. For small densities of vehicles, the flow increases with the density (the more vehicles, the more flow). Beyond a *critical* density, the flow ceases to increase and congestion appears. If the density increases further, the flow decreases because of congestion, until it eventually reaches zero, which corresponds to having a continuum of vehicles stopped on the road. Several models have been proposed to describe the empirical relationship ψ between flow and density since the seminal work of Greenshield, see for example [85, 18] and the trapezoidal or triangular models [53, 54]. As is the case for most of these models, the fundamental diagram is assumed to be *concave*. In particular, numerical illustrations throughout the thesis use two of the most common fundamental diagrams:

- The *triangular* fundamental diagram is fully characterized by three parameters: v_f, the free flow speed (m/s); $\rho_{\max}$, the jam (or maximum) density (veh/m); and $q_{\max}$, the capacity (veh/m). It is defined on $D_\psi = [0, \rho_{\max}]$ by

$$\psi(\rho) = \begin{cases} v_f \rho & \text{if} \quad \rho \in [0, \rho_c] \\ w(\rho_c - \rho) + v_f \rho_c & \text{if} \quad \rho \in [\rho_c, \rho_{\max}] \end{cases}, \tag{2.2}$$

  and is illustrated Figure 2.1 (left).

- The *Greenshields* fundamental diagram is parameterized by two parameters: $\rho_{\max}$, the maximum density (veh/m) and $q_{\max}$, the maximum flow (veh/h). It is defined on $D_\psi = [0, \rho_{\max}]$ by

$$\psi(\rho) = 4\frac{q_{\max}}{\rho_{\max}^2}\rho(\rho_{\max} - \rho), \tag{2.3}$$

  and is illustrated Figure 2.1 (right).

The Moskowitz Hamilton-Jacobi partial differential equation

Because density is an aggregated quantity, which cannot be measured by probe vehicles directly, the LWR PDE is difficult to use as such to incorporate trajectory data available from probe vehicles. The section presents an alternate representation of the LWR model which was introduced by Newell and Daganzo [172, 56, 57], following the work of Moskowitz [169].

Imagine assigning consecutive integer labels to vehicles entering a road segment at a user defined location $x = \xi$. The vehicles are counted from the reference point $(t = 0, x = \xi)$. The

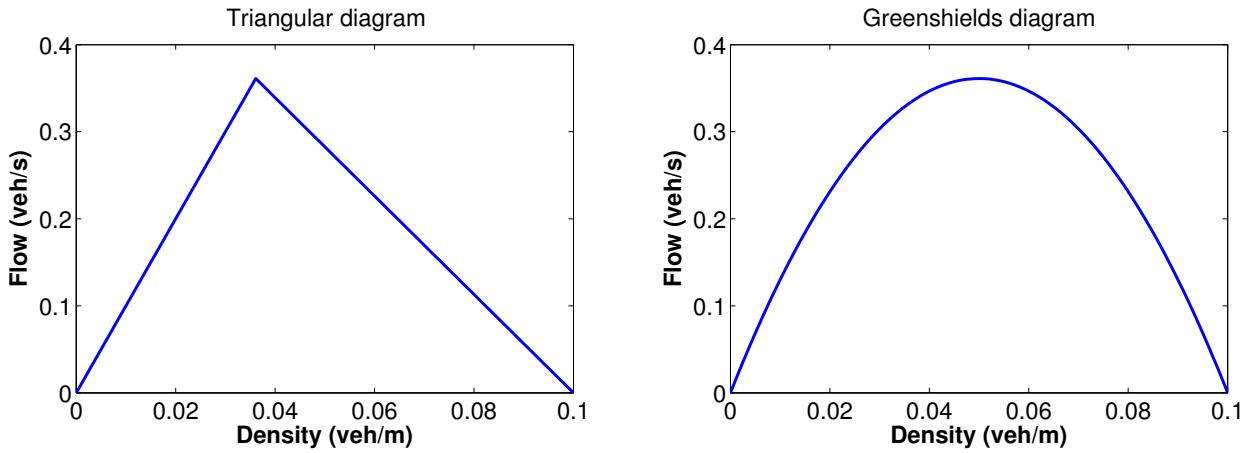


Figure 2.1: Examples of concave flux functions (fundamental diagrams), ψ, used in the numerical simulations. In the context of traffic flow, they represent the empirical relation between flow and density. **Left**: Triangular Hamiltonian, parameterized by the free flow speed ($\nu^\flat = 10$ m/s), the capacity ($q_{\max} = 1300$ veh/h) and the maximum density ($\rho_{\max} = 1/10$ veh/m). **Right**: Greenshields Hamiltonian, parameterized by the capacity ($q_{\max} = 1300$ veh/h) and the maximum density ($\rho_{\max} = 1/10$ veh/m).

first vehicle is assigned an arbitrary label[1], usually chosen to be 0. The *Moskowitz function* $\mathbf{M}(t,x)$ (also known as *cumulative number of vehicles function*) is a continuous representation of the label function. It encodes the distribution of the vehicles on the highway at all locations and times. The space and time derivatives of the Moskowitz function are related to the flow and density functions as follows [172, 56, 57]:

$$\frac{\partial \mathbf{M}(t,x)}{\partial t} = q(\rho(t,x)) \qquad \text{and} \qquad \frac{\partial \mathbf{M}(t,x)}{\partial x} = -\rho(t,x). \tag{2.4}$$

Using equation (2.4), one can transform equation (2.1) into the following *Moskowitz Hamilton-Jacobi* PDE [56, 57]:

$$\frac{\partial \mathbf{M}(t,x)}{\partial t} - q\left(-\frac{\partial \mathbf{M}(t,x)}{\partial x}\right) = 0. \tag{2.5}$$

2.3 Estimation with Eulerian and Lagrangian sensing

As mentioned in Section 2.1, the computation of numerical solutions to the *Hamilton-Jacobi partial differential equation* subject to boundary conditions, initial conditions or sometimes

[1]The choice of this arbitrary label at $t = 0$ and $x = \xi$ does not influence the results

terminal conditions is a topic which has generated significant interest in the control and numerical analysis community [195, 178, 167].

The integration of initial or boundary conditions alone may not be sufficient to solve new data reconstruction problems arising in the context of Lagrangian sensing [123]. Recall that *Lagrangian* sensing refers to measurements performed along a sensor's trajectory, such as smartphones traveling onboard cars reporting their position (and sometimes velocity) with a chosen sampling scheme. This is in contrast to *Eulerian* sensing, in which sensors are fixed (for example, video cameras or loop detectors along highways) and monitor a specific location or domain.

A model capable of mathematically handling initial, boundary and internal conditions is developed in [41, 42]. The section summarizes the main results of this work which are used in Chapters 3 and 4.

Mathematical background, notation and definitions

We consider a scalar Hamilton-Jacobi PDE, as given by (2.5).

The *Hamiltonian* (also referred to as *flux function* or *fundamental diagram*), ψ, is assumed to be *concave* on its domain of definition $D_\psi = [0, \rho_{\max}]$ and to satisfy $\psi(0) = \psi(\rho_{\max}) = 0$. The maximum value of ψ on D_ψ is denoted $q_{\max}$. The concavity of ψ imposes that ψ has a right derivative and a left derivative in the interior of D_ψ. The variables $\nu^\flat$ and $\nu^\sharp$ are defined as $\nu^\flat = \psi'(0)$ and $\nu^\sharp = -\psi'(\rho_{\max})$. The concavity and the condition that $\psi(0) = \psi(\rho_{\max}) = 0$ impose that $\nu^\flat > 0$ and $\nu^\sharp > 0$.

The mathematical properties of the solution of (2.5) require specific treatments to introduce internal value conditions. In particular, the dissertation investigates a specific control framework based on Lax-Hopf's formula and viability theory [12, 41, 42]. The convex transform φ^* of the Hamiltonian is defined as follows.

Definition 2.1 (Convex transform)**.** *Let ψ be a concave function defined on D_ψ, its convex transform φ^* takes finite values on $D_{\varphi^*} = [-\nu^\flat, \nu^\sharp]$:*

$$\varphi^*(u) = \begin{cases} \sup\limits_{p \in D_\psi} [pu + \psi(p)] & \text{if } u \in [-\nu^\flat, \nu^\sharp] \\ +\infty & \text{otherwise} \end{cases} \tag{2.6}$$

Let $\mathbf{c}$ be a lower semi-continuous function defined on a subset of $[0, t_{\max}] \times [\xi, \chi]$. It represents a *value condition*, *i.e.* a value that is imposed on the solution of (2.5). The viability epi-solution [11, 41, 42] $\mathbf{M_c}$ associated with $\mathbf{c}$ is given by a Lax-Hopf formula and is the unique generalized solution of (2.5) in the *Barron-Jensen/Frankowska* (B-J/F) sense [11]. The formula implies an inf-morphism property [11, 41, 42], which is a key property used in Chapters 3 and 4.

Fact 2.1 (Inf-morphism)**.** *Let* $\mathbf{c}$ *be the minimum of a finite number of functions* $\mathbf{c}_i$, $i \in I$. *The Lax-Hopf formula implies that:*

$$\forall\,(t,x) \in [0,t_{\max}] \times [\xi,\chi]\ \mathbf{M}_{\mathbf{c}}(t,x) = \inf_{i\in I} \mathbf{M}_{\mathbf{c}_i}(t,x)$$

The inf-morphism property is a practical tool to integrate new value conditions and separate a complex problem involving multiple value conditions into a set of more tractable subproblems [41, 42].

State estimation with affine value conditions

The solution associated with an affine initial, boundary or internal value condition has an analytical expression derived in [42]. The following notation and definitions are used in the analytical derivations of the solution. More details and proofs are available in [42].

Definition 2.2 (Upper and lower critical densities [42])**.** *The upper (resp. lower) critical density* $\overline{\rho}_c$ *(resp.* $\underline{\rho}_c$*) is the maximum (resp. minimum)* $\rho \in [0,\rho_{\max}]$ *such that* $\psi(\rho) = q_{\max}$.

Definition 2.3 (Densities associated with q [83])**.** *For* $q \in [0,q_{\max}]$, $\overline{\rho}(q)$ *(resp.* $\underline{\rho}(q)$*) is the unique solution of* $\psi(\rho) = q$ *for* $\rho \in [\overline{\rho}_c,\rho_{\max}]$ *(resp. for* $\rho \in [0,\underline{\rho}_c]$*).*

Following [29], the sub- and super-derivative (∂_- and ∂_+) are defined as follows:

$$v \in \partial_- f(x_0) \Leftrightarrow \forall x \in D_f,\ f(x) \geq f(x_0) + v(x-x_0)$$
$$v \in \partial_+ f(x_0) \Leftrightarrow \forall x \in D_f,\ f(x) \leq f(x_0) + v(x-x_0)$$

Definition 2.4. *For* $\rho \in [0,\rho_{\max}]$, $u_0^+(\rho)$ *(resp.* $u_0^-(\rho)$*) is an element of* $-\partial_+\psi(\rho) \cap \mathbb{R}^+$ *(resp.* $-\partial_+\psi(\rho) \cap \mathbb{R}^-$*). Note that* $u_0^+(\rho)$ *(resp.* $u_0^-(\rho)$*) is not uniquely defined if* ψ *is not differentiable in* ρ. *It was shown [42] that any choice of* $u_0^+(\rho)$ *(resp.* $u_0^-(\rho)$*) in* $-\partial_+\psi(\rho)$ *provides the expression of the solution of the HJ-PDE.*

Definition 2.5 (Capture times [42])**.** *The capture times* $\overline{T}_0$ *and* $\underline{T}_0$ *are defined as follows:*

$$\overline{T}_0(\rho,x) = \begin{cases} \frac{\chi-x}{u_0^+(\rho)} & \text{if } u_0^+(\rho) \neq 0 \\ +\infty & \text{otherwise} \end{cases} \forall (\rho,x) \in [\overline{\rho}_c,\rho_{\max}] \times [\xi,\chi],$$
$$\underline{T}_0(\rho,x) = \begin{cases} \frac{\xi-x}{u_0^-(\rho)} & \text{if } u_0^-(\rho) \neq 0 \\ +\infty & \text{otherwise} \end{cases} \forall (\rho,x) \in [0,\underline{\rho}_c] \times [\xi,\chi],$$

Definition 2.6 (Affine value conditions)**.** *Affine initial, upstream, downstream and internal value conditions are defined as follows:*

- *An affine initial value condition* $\mathcal{M}_{0_i}$ *is defined on* $\{0\} \times [\overline{\alpha}_i,\overline{\alpha}_{i-1}]$ *by*

$$\mathcal{M}_{0_i}(t,x) = b_i + a_i(\overline{\alpha}_i - x) \tag{2.7}$$

- *An affine upstream value condition* γ_j *is defined on* $[\bar{\gamma}_j, \bar{\gamma}_{j+1}] \times \{\xi\}$ *by*

$$\gamma_j(t,x) = d_j + (t - \bar{\gamma}_j)\psi(\rho_j), \quad \rho_j \in [0, \overline{\rho}_c] \tag{2.8}$$

- *An affine downstream value condition* β_k *is defined on* $[\bar{\beta}_k, \bar{\beta}_{k+1}] \times \{\chi\}$ *by*

$$\beta_k(t,x) = f_k + (t - \bar{\beta}_k)\psi(\rho_k), \quad \rho_k \in [\underline{\rho}_c, \rho_{\max}] \tag{2.9}$$

- *An affine internal condition* μ_l *is defined for* $x = x_l + v_l(t - \bar{\delta}_l)$ *and* $t \in [\bar{\delta}_l, \bar{\delta}_{l+1}]$ *by*

$$\mu_l(t,x) = g_l(t - \bar{\delta}_l) + h_l \tag{2.10}$$

  The variable x_{l+1} *is defined as* $x_{l+1} = x_l + v_l(\bar{\delta}_{l+1} - \bar{\delta}_l)$

Fact 2.2 (Explicit component solutions)**.** *The analytical solutions of the viability episolution associated with affine initial, upstream, downstream and internal value conditions are as follows:*

- *The solution associated with an initial condition* $\mathcal{M}_{0_i}$*, defined by* (2.7)*, takes finite values for* $(t,x) \in [0, t_{\max}] \times [\xi, \chi]$ *such that* $x \geq \overline{\alpha}_i - \nu^\sharp t$ *and* $x \leq \overline{\alpha}_{i+1} + \nu^\flat t$*. On this domain, the solution is defined as follows:*

$$\mathbf{M}_{\mathcal{M}_{0_i}}(t,x) = \begin{cases} (i)\ b_i + a_i(\overline{\alpha}_i - x) + t\psi(a_i)\ \textit{if}\ u_0(a_i) \in \left[\frac{\overline{\alpha}_i - x}{t}, \frac{\overline{\alpha}_{i-1} - x}{t}\right] \\ (ii)\ b_i + t\varphi^*\left(\frac{\overline{\alpha}_i - x}{t}\right)\ \textit{if}\ u_0(a_i) \leq \frac{\overline{\alpha}_i - x}{t} \\ (iii)\ b_i + a_i(\overline{\alpha}_i - \overline{\alpha}_{i-1}) + t\varphi^*\left(\frac{\overline{\alpha}_{i-1} - x}{t}\right)\ \textit{if}\ u_0(a_i) \geq \frac{\overline{\alpha}_{i-1} - x}{t} \end{cases} \tag{2.11}$$

- *The solution associated with an upstream condition* γ_j*, defined by* (2.8)*, takes finite values for* $(t,x) \in [0, t_{\max}] \times [\xi, \chi]$ *such that* $t \geq \bar{\gamma}_j + \frac{x - \xi}{\nu^\flat}$*. On this domain, the solution is defined as follows:*

$$\mathbf{M}_{\gamma_j}(t,x) = \begin{cases} (i)\ (t - \bar{\gamma}_j)\psi(\rho_j) + \rho_j(\xi - x) + d_j\ \textit{if}\ \underline{T}_0(\rho_j, x) \in [t - \overline{\gamma}_{j+1}, t - \overline{\gamma}_j] \\ (ii)\ d_j + (t - \overline{\gamma}_j)\varphi^*\left(\frac{\xi - x}{t - \overline{\gamma}_j}\right)\ \textit{if}\ \underline{T}_0(\rho_j, x) \geq t - \overline{\gamma}_j \\ (iii)\ (\overline{\gamma}_{j+1} - \overline{\gamma}_j)\psi(\rho_j) + d_j + (t - \overline{\gamma}_{j+1})\varphi^*\left(\frac{\xi - x}{t - \overline{\gamma}_{j+1}}\right) \\ \qquad\qquad \textit{if}\ \underline{T}_0(\rho_j, x) \leq t - \overline{\gamma}_{j+1} \end{cases} \tag{2.12}$$

- *The solution associated with a downstream condition* β_k*, defined by* (2.9)*, takes finite values for* $(t,x) \in [0, t_{\max}] \times [\xi, \chi]$ *such that* $t \geq \bar{\beta}_k + \frac{\chi - x}{\nu^\sharp}$*. On this domain, the solution*

is defined as follows:

$$\mathbf{M}_{\beta_k}(t,x) = \begin{cases} (i)\ (t-\bar{\beta}_k)\psi(\rho_k) - \rho_k(\chi - x) + f_k \ \textit{if}\ \overline{T}_0(\rho_k, x) \in [t-\bar{\beta}_{k+1}, t-\bar{\beta}_k] \\ (ii)\ f_k + (t-\bar{\beta}_k)\varphi^*\left(\frac{\chi - x}{t-\bar{\beta}_k}\right) \ \textit{if}\ \overline{T}_0(\rho_k, x) \geq t-\bar{\beta}_k \\ (iii)\ (\bar{\beta}_{k+1}-\bar{\beta}_k)\psi(\rho_k) + f_k + (t-\bar{\beta}_{k+1})\varphi^*\left(\frac{\chi - x}{t-\bar{\beta}_{k+1}}\right) \\ \qquad\qquad \textit{if}\ \overline{T}_0(\rho_k, x) \leq t-\bar{\beta}_{k+1} \end{cases} \tag{2.13}$$

- *The analytical expression of the solution $\mathbf{M}_{\mu_l}$ associated with an internal condition μ_l, defined by* (2.10) *requires some definitions and notation which are out of the scope of this thesis. They are fully detailed in [42], Section II.E. For this reason, the analytical expression of $\mathbf{M}_{\mu_l}$ is omitted. Some of the properties of the solution are used in Chapter 3, with explicit reference to [42].*

The inf-morphism property implies that the solution of (2.5) subject to piecewise affine value conditions is the minimum of the viability episolutions computed for each of the affine conditions.

Chapter 3

Deterministic estimation with Lagrangian measurements

The modeling and derivations of Chapter 2 provide a useful framework to design accurate estimation frameworks in networks governed by conservation laws [44, 163]. Signalized flow networks fit within this framework. The chapter leverages the results of Chapter 2 to study these systems. In particular, the dynamics is governed by the presence of signals, with, in general, unknown parameters, which lead to periodic drops of the capacity at intersections and to the formation of queues. Today, GPS provide Lagrangian measurements of traffic conditions, through measurements of the position of the vehicle at different times. For this reason, it is natural to model traffic with the Moskowitz function introduced in Section 2.2 and to use the estimation results for Hamilton Jacobi PDEs presented in Secion 2.3. This chapter shows how the GPS measurements can be used to reconstruct downstream boundary conditions, *i.e.* to estimate capacity drops at intersections of the road network. The state of the road network (value of the Moskowitz function, density, velocity and flow) can then be estimated at any location x and time t.

More generally, the chapter proposes an algorithm for reconstructing downstream boundary conditions for a class of Hamilton-Jacobi partial differential equations, for which initial and upstream boundary conditions are prescribed as piecewise affine functions and an internal condition is prescribed as an affine function. Based on viability theory, the chapter derives an algorithm to reconstruct the downstream boundary condition such that the solution of the Hamilton-Jacobi equation with the prescribed initial and upstream conditions and reconstructed downstream boundary condition satisfies the internal value condition.

The chapter is organized as follows. Section 3.1 presents a motivating example and gives insights on the derivations of the reconstruction algorithm in the case of urban traffic estimation. Section 3.2 introduces the mathematical background and states the reconstruction problem of the downstream boundary condition. Section 3.3 proves the existence of a solution to the reconstruction problem under some compatibility conditions between the given initial, upstream and internal value conditions. The algorithm derived to solve the reconstruction problem is detailed in Section 3.4 and illustrated numerically in Section 3.5.

3.1 Motivating example

The section motivates the derivation of an algorithm for reconstructing downstream boundary conditions through an example arising in signalized flow networks. As done in Chapter 2, the section uses the Moskowitz function to represent the state of traffic at any location $x \in [\xi, \chi]$ and time $t \in [0, T]$ on a signalized road segment. The Moskowitz function satisfies the Hamilton-Jacobi partial differential equation (2.5). Measurements of traffic conditions prescribe the value of the function on a subset of $[\xi, \chi] \times [0, T]$.

An initial condition $\mathcal{M}_0$ prescribes the state of the road segment at the initial time $t = 0$ for $x \in [\xi, \chi]$. This initial condition can be obtained by using an aerial photo of the road segment at $t = 0$. An upstream condition γ prescribes the flow of vehicles entering the segment at its upstream boundary $x = \xi$ for $t \in [0, T]$. This upstream condition can be obtained through flow measurements upstream of the road segment (*e.g.* using loop-detectors, cameras and so on). Given the initial and upstream value conditions, the solution is computed semi-analytically at any location x and time t. The solution illustrates the trajectories of the vehicles present on the road segment at the initial time $t = 0$ and arriving at the upstream boundary $x = \xi$ (Figure 3.1).

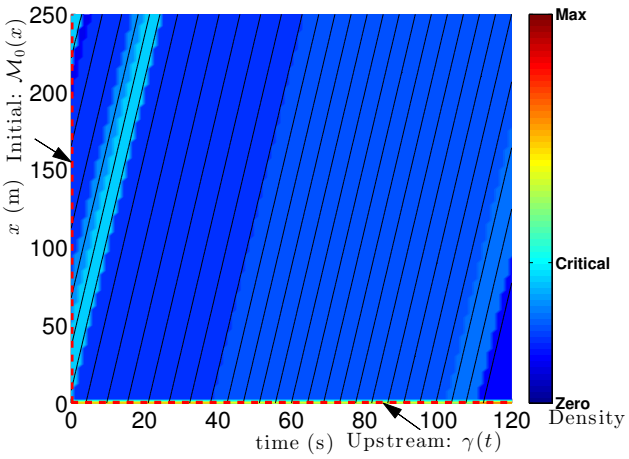


Figure 3.1: Colormap of the density of vehicles and isolines of the Moskowitz function, solution of the Hamilton-Jacobi equation (2.5) subject to initial and upstream value conditions.

The presence of a signal, or more generally of a bottleneck, at the downstream boundary ($x = \chi$) limits the maximum flow at certain times. This limitation of the maximum flow is imposed on the solution by prescribing a downstream boundary condition β at the downstream boundary $x = \chi$. When imposing this limitation on the maximum flow, the Moskowitz function, solution of the Hamilto-Jacobi partial differential equation (2.5) is illustrated Figure 3.2. In particular, the figure shows the formation of a queue upstream of the limitation of the maximum flow. Once the limitation on the maximum flow is released, the

queue dissipates, according to Equation (2.5). The speed of formation of the queue depends on the density of vehicles upstream of the queue: the more vehicle, the faster the queue formation. The speed of dissolution of the queue only depends on the density inside the queue and is thus constant throughout the queue dissipation.

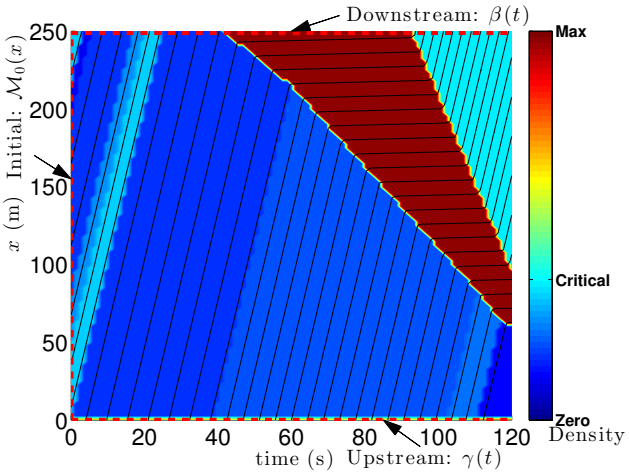


Figure 3.2: Colormap of the density of vehicles and isolines of the Moskowitz function, solution of the Hamilton-Jacobi equation (2.5) subject to initial, upstream and downstream value conditions. The downstream value condition limits the maximum flow for the duration of the signal red time (or bottleneck).

For most applications, the specific characteristics of signals or bottlenecks are unknown. Digital maps usually provide the location of traffic signals; traffic information providers have some information regarding the presence of incidents and road closures which cause bottlenecks, but the characteristics of the flow limitation (beginning and end of the limitation and maximum flow during the capacity limitation) are usually unknown. Instead, GPS devices on-board vehicles provide information about individual trajectories, *i.e* about isolines of the Moskowitz function. A trajectory measurement is integrated in the estimation framework as an internal value condition μ. The spatio-temporal domain representing the road segment during the estimation time is illustrated in Figure 3.3 (left) with the initial, upstream and internal value conditions.

The Barron-Jensen/Frankowska solution of the *Moskowitz Hamilton-Jacobi* partial differential equation subject to initial, upstream and internal value conditions is illustrated Figure 3.3 (right). It corresponds to the solution which maximizes the flow subject to the constraints on the value of the solution. In particular, the figure shows the creation of the queue upstream of the vehicle reporting its GPS trajectory. However, the solution does not leverage the information regarding the presence of a signal at the downstream boundary of the road segment, which caused the sensed vehicle to slow down.

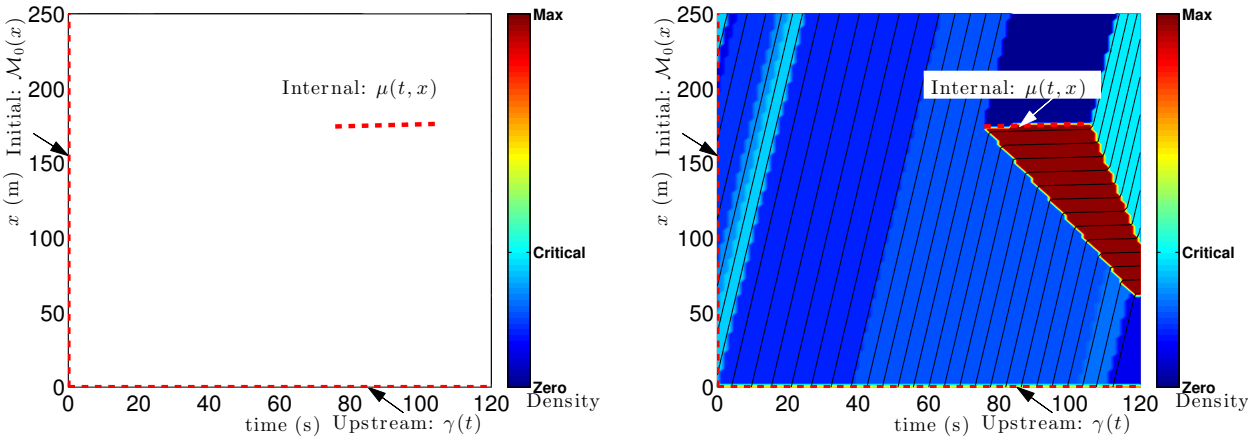


Figure 3.3: **Left**: Illustration of the spatio-temporal estimation domain with the value conditions imposed on the solution. The initial, upstream and internal value conditions represent the initial vehicles present on the road at $t = 0$, the flow of vehicles arriving at the upstream boundary and the trajectory measurement respectively. **Right**: Colormap of the density of vehicles and isolines of the Moskowitz function, solution of the Hamilton-Jacobi equation (2.5) subject to initial, upstream and internal value conditions. The internal value condition represents the trajectory of a vehicle obtained by GPS tracking.

This chapter derives an algorithm which reconstructs downstream boundary conditions, corresponding to a constant limitations of the maximum flow given initial, upstream and internal value conditions. The approach recovers the downstream boundary condition that corresponds to the minimal amount of red light time that explains the internal condition. Under the assumption that the vehicle reported its trajectory during the entire duration of its slow-down and that the reduction of capacity due to the bottleneck is constant, this results in an estimate of the flow shown in Figure 3.2. The figure represents the solution of (2.5) subject to the initial, upstream and reconstructed downstream value conditions (corresponding to the minimal bottleneck time at the downstream boundary). The modeling of congestion propagation through the *Hamilton-Jacobi* partial differential equation and the information on the presence of the signal are leveraged to reconstruct the specific characteristics of the bottleneck (beginning, end and extent of the capacity reduction) and compute the value of the Moskowitz function (representing the state of traffic) at any time t and location x.

3.2 Problem statement

Consider given continuous piecewise affine initial and upstream boundary conditions, denoted $\mathcal{M}_0$ and γ respectively. Following Definition 2.6, $\mathcal{M}_{0_i}$, $i \in \{1, \dots, I_0\}$ define affine initial value conditions (Equation 2.7) and γ_j, $j \in \{1, \dots, I_\gamma\}$ define affine upstream value conditions (Equation 2.8). The conditions are such that $\forall (t,x) \in [0, t_{\max}] \times [\xi, \chi]$, $\mathcal{M}_0(t,x) =$

$\min_{i=1}^{I_0} \mathcal{M}_{0_i}(t,x)$ and $\gamma(t,x) = \min_{j=1}^{I_\gamma} \gamma_j(t,x)$. Consider also an affine internal value condition μ_l as defined by (2.7). The function ζ_l is defined for $t \in [\bar{\delta}_l, \bar{\delta}_{l+1}]$ by $\zeta_l(t) = x_l + v_l(t - \bar{\delta}_l)$. The constants g_l and v_l are assumed to satisfy $0 \leq g_l \leq \psi(\overline{\rho}_c) - \overline{\rho}_c v_l$ and $0 \leq v_l \leq \nu^\flat$.

A downstream boundary condition β is defined as a value condition that takes finite values on a subset of $[0, t_{\max}] \times \{\chi\}$. At time t, the downstream boundary condition $\beta(t,\chi)$ characterizes the limitation of the capacity at $x = \chi$, which is important to detect and control saturation and bottlenecks before they propagate throughout the network. This motivates the following reconstruction problem:

Problem 3.1 (Initial Boundary Value Problem)**.** Given *an affine internal value condition μ_l, piecewise affine upstream boundary condition γ and initial condition $\mathcal{M}_0$;* reconstruct *the downstream boundary condition $\hat{\beta}$ such that the Barron-Jensen/Frankowska solution of the* Initial Boundary Value Problem *of the HJ-PDE* (2.5) *with the prescribed initial and boundary conditions $\mathcal{M}_0$, γ and $\hat{\beta}$ satisfies the internal condition:*

$$\forall t \in [\bar{\delta}_l, \bar{\delta}_{l+1}],\ \forall x = \zeta_l(t), \qquad \min(\mathbf{M}_{\mathcal{M}_0}, \mathbf{M}_\gamma, \mathbf{M}_{\hat{\beta}})(t,x) = \mu_l(t,x). \tag{3.1}$$

For an affine downstream boundary condition β_k, as defined in Equation (2.9), define $e_k = \psi(\rho_k)$. The expression of the solution $\mathbf{M}_{\beta_k}$ (2.13) of the HJ-PDE subject to the downstream boundary condition has specific analytical expressions in the domains (i), (ii) and (iii), defined as follows:

$$\mathbf{M}_{\beta_k}(t,x) = \begin{cases} f_k + (t - \overline{\beta}_k)\varphi^*\left(\frac{\chi - x}{t - \overline{\beta}_k}\right) \text{ if } \overline{T}_0(\rho_k, x) \geq t - \overline{\beta}_k & (i) \\ (t - \overline{\beta}_k)e_k + (\chi - x)\rho_k + f_k & (ii) \\ \qquad \text{if } \overline{T}_0(\rho_k, x) \in [t - \overline{\beta}_{k+1}, t - \overline{\beta}_k] & \\ f_k + (\overline{\beta}_{k+1} - \overline{\beta}_k)e_k + (t - \overline{\beta}_{k+1})\varphi^*\left(\frac{\chi - x}{t - \overline{\beta}_{k+1}}\right) & (iii) \\ \qquad \text{if } \overline{T}_0(\rho_k, x) \leq t - \overline{\beta}_{k+1} & \end{cases} \tag{3.2}$$

3.3 Existence of a solution

The section derives conditions on $\mathcal{M}_0$, γ and μ_l for the existence of a downstream boundary condition $\hat{\beta}$ which solves Problem 3.1. It studies uniqueness properties among piecewise affine solutions and exhibit a solution that corresponds to a constant limitation of the maximum flow in an interval $[\tau_1, \tau_2]$.

Interval with affine downstream boundary condition

Given that μ_l is affine, $\hat{\beta}$ is necessarily such that $\mathbf{M}_{\hat{\beta}}$ is affine on the trajectory ζ_l (since $\mathbf{M}_{\hat{\beta}}$ and μ_l coincide on the domain of μ_l). The derivative of $\mathbf{M}_{\hat{\beta}}$ in the direction $(1, v_l)$ should thus exist in the domain $\{(t,x) \text{ s.t. } t \in [\bar{\delta}_l, \bar{\delta}_{l+1}],\ x = \zeta_l(t)\}$ and should be equal to g_l. First, the following lemma characterizes intervals in which φ^* is affine:

Lemma 3.1 (Intervals in which φ^* is affine)**.** *The function φ^* is affine in $[u_1, u_2]$ if and only if there exists $\rho \in D_\psi$ such that $(u_1, u_2) \subset -\partial^+\psi(\rho)$.*

Proof. The function φ^* is affine in the interval $[u_1, u_2]$ if and only if its subgradient is reduced to a given ρ^* in (u_1, u_2). The subderivative of φ^* satisfies the Legendre-Fenchel inversion formula [11]:

$$u \in -\partial_+\psi(\rho) \Leftrightarrow \rho \in \partial_-\varphi^*(u).$$

Since $\partial_-\varphi^*(u) = \{\rho^*\}$ for $u \in (u_1, u_2)$, it follows that $(u_1, u_2) \subset -\partial^+\psi(\rho^*)$. □

Definition 3.1 (Density associated with v_l and g_l)**.** *Let f_{v_l} be defined for $\rho \in [0, \rho_{\max}]$ by $f_{v_l}(\rho) = \psi(\rho) - v_l\rho$. The function is concave as the sum of concave functions, and attains its maximum value $\varphi^*(-v_l)$ in a closed interval (Definition 2.1). Let ρ^* be the upper bound of this interval. Section 3.2 introduced the assumption that $0 \leq g_l \leq \psi(\overline{\rho}_c) - \overline{\rho}_c v_l = f_{v_l}(\overline{\rho}_c)$, and thus $0 \leq g_l \leq \varphi^*(-v_l)$. Since f_{v_l} is continuous and $f_{v_l}(\rho_{\max}) \leq 0$, the intermediate value theorem states that there exists a solution $\tilde{\rho}(v_l, g_l) \in [\rho^*, \rho_{\max}]$ such that $f_{v_l}(\tilde{\rho}(v_l, g_l)) = g_l$. Given that f_{v_l} is concave and given the definition of ρ^*, f_{v_l} is strictly decreasing on $[\rho^*, \rho_{\max}]$ which proves that $\tilde{\rho}(v_l, g_l)$ is unique. Given that $g_l \leq f_{v_l}(\overline{\rho}_c)$, then $\tilde{\rho}(v_l, g_l) \geq \overline{\rho}_c$.*

Definition 3.2 (Compatibility conditions)**.** *A necessary condition for Problem* (3.1) *to be well posed is to have* compatible *initial, upstream and internal conditions, as defined in [41, 42]. The compatibility of the value conditions is necessary and sufficient for the existence of a solution which satisfies* (2.5) *and all value conditions. It means that all these conditions can be imposed simultaneously and is written as*

$$\begin{aligned} &\min(\mathbf{M}_{\mathcal{M}_0}, \mathbf{M}_\gamma)(t,x) \geq \mu_l(t,x)\ \forall t \in [\bar{\delta}_l, \bar{\delta}_{l+1}],\ x = \zeta_l(t) \\ &\min(\mathbf{M}_{\mathcal{M}_0}, \mathbf{M}_\mu)(t,x) \geq \gamma(t,x)\ \forall (t,x) \in [0, t_{\max}] \times \{\xi\} \\ &\min(\mathbf{M}_\gamma, \mathbf{M}_\mu)(t,x) \geq \mathcal{M}_0(t,x)\ \forall (t,x)\{0\} \times [\xi, \chi] \end{aligned} \tag{3.3}$$

The variables ρ_{out} and q_{out} are defined as $\rho_{\text{out}} = \tilde{\rho}(v_l, g_l)$ and $q_{\text{out}} = \psi(\rho_{\text{out}})$. The compatibility conditions between $\mathcal{M}_0$, γ and μ_l are assumed to be satisfied.

Proposition 3.1 (Affine boundary condition)**.** *If the internal condition μ_l is such that ψ is differentiable at ρ_{out}, there exists an interval $[\tilde{\tau}_1, \tilde{\tau}_2]$ in which any piecewise affine solution of Problem 3.1 is necessarily affine, with temporal derivative equal to q_{out}.*

Proof. Let $\hat{\beta}$ be a potential piecewise affine solution. If such a solution exists, there exists a set of functions $(\beta_k)_{k \in K}$, defined by (2.9) such that $\forall (t,x) \in [0, t_{\max}] \times \{\chi\}$, $\hat{\beta}(t,x) = \min_{k \in K} \beta_k(t,x)$.

For each $k \in K$, let $\underline{t}_k$ and $\overline{t}_k$ be such that, in the domain defined by $t \in [\underline{t}_k, \overline{t}_k]$ and $x = \zeta_l(t)$, $\mathbf{M}_{\hat{\beta}}(t,x) = \mathbf{M}_{\beta_k}(t,x)$. The proof shows that he points $(\underline{t}_k, \zeta_l(\underline{t}_k))$ and $(\overline{t}_k, \zeta_l(\overline{t}_k))$ necessarily belong to the domain (ii) of the downstream boundary condition β_k. It then shows that the temporal derivative of β_k is necessarily equal to q_{out} and concludes.

Since $\mathbf{M}_{\hat{\beta}}$ is a solution of Problem 3.1, it takes finite values at $(\overline{t}_k, \zeta_l(\overline{t}_k))$ and $(\underline{t}_k, \zeta_l(\underline{t}_k))$. These points necessarily belong to one of the domains (i), (ii) or (iii) of $\mathbf{M}_{\beta_k}$.

- If $(\overline{t}_k, \zeta_l(\overline{t}_k))$ belongs to domain (iii), let $\delta_k \geq \underline{t}_k$ be the first time such that $(\delta_k, \zeta_l(\delta_k))$ is in domain (iii). The function $\mathbf{M}_{\beta_k}$ is necessarily affine along the trajectory ζ_l with derivative equal to g. For any $t \in [\delta_k, \overline{t}_k]$ such that $\mathbf{M}_{\beta_k}$ is differentiable in $(t, \zeta_l(t))$, its total derivative along the trajectory ζ_l is given by

$$\frac{d\mathbf{M}_{\beta_k}}{dt}(t, \zeta_l(t)) = \varphi^*(u(t)) - (v + u(t))\,(\varphi^*)'\,(u(t)), \tag{3.4}$$

  with $u(t) = \frac{\chi - \zeta_l(t)}{t - \overline{\beta}_{k+1}}$. It follows that

$$\frac{d^2\mathbf{M}_{\beta_k}}{dt^2}(t, \zeta_l(t)) = 0 \Leftrightarrow (\varphi^*)''\,(u(t)) = 0,\ \forall t \in [\delta_k, \overline{t}_k].$$

  Necessarily, φ^* is affine on $[u(\delta_k), u(\overline{t}_k)]$ and Lemma 3.1 proves that there exists $\rho^* \in D_\psi$ such that $[u(\delta_k), u(\overline{t}_k)] \subset -\partial^+\psi(\rho^*)$. It implies that, on the trajectory ζ_l, $\varphi^*(u(t)) = \psi(\rho^*) + u(t)\rho^*$ and $(\varphi^*)'\,(u(t)) = \rho^*$. The total derivative of $\mathbf{M}_{\beta_k}$ along the trajectory is thus given by

$$\frac{d\mathbf{M}_{\beta_k}}{dt}(t, \zeta_l(t)) = \psi(\rho^*) - v\rho^*.$$

  Since $\frac{d\mathbf{M}_{\beta_k}}{dt}(t, \zeta_l(t)) = g$, $\rho^* = \tilde{\rho}(v, g)$; since ψ is differentiable at $\rho_{\text{out}} = \tilde{\rho}(v, g)$, $-\partial^+\psi(\rho^*)$ is reduced to a singleton. This implies that $u(\delta_k) = u(\overline{t}_k)$ and thus $\delta_k = \overline{t}_k$. The point $(\overline{t}_k, \zeta_l(\overline{t}_k))$ is at the boundary of the domains (ii) and (iii).
  Similarly, if $(\underline{t}_k, \zeta_l(\underline{t}_k))$ is in the domain (i), it is also in the domain (ii) and thus at the intersection of the two domains.

- In the domain (ii), $\mathbf{M}_{\beta_k}$ is affine and its total derivative along the trajectory ζ_l is given by

$$\frac{d\mathbf{M}_{\beta_k}}{dt}(t, \zeta_l(t)) = \psi(\rho_k) - v\rho_k$$

  Necessarily, $\rho_k = \rho_{\text{out}}$ and $f_k = \psi(\rho_k) = q_{\text{out}}$. For the points $(\underline{t}_k, \zeta_l(\underline{t}_k))$ and $(\overline{t}_k, \zeta_l(\overline{t}_k))$ to be included in the domain (ii), the following must hold:

$$\overline{\beta}_k \leq \underline{t}_k - \frac{\chi - \zeta_l(\underline{t}_k)}{u_0^+(\rho_{\text{out}}, \zeta_l(\underline{t}_k))} \text{ and } \overline{\beta}_{k+1} \geq \overline{t}_k - \frac{\chi - \zeta_l(\overline{t}_k)}{u_0^+(\rho_{\text{out}}, \zeta_l(\overline{t}_k))}$$

For all k such that $\mathbf{M}_{\hat{\beta}}(t,\zeta_l(t)) = \mathbf{M}_{\beta_k}(t,\zeta_l(t))$ for $t \in [\underline{t}_k, \bar{t}_k]$, β_k has a temporal derivative equal to q_{out}. The continuity of $\mathbf{M}_{\hat{\beta}}$ imposes that there exists a unique $k = k^*$ such that $\mathbf{M}_{\hat{\beta}}(t,x) = \mathbf{M}_{\beta_{k^*}}(t,x)$ on the trajectory ζ_l. Let $\tilde{\tau}_1$ and $\tilde{\tau}_2$ be defined as follows:

$$\tilde{\tau}_1 = \bar{\delta}_l - \frac{\chi - x_l}{u_0^+(\rho_{\text{out}}, x_l)} \text{ and } \tilde{\tau}_2 = \bar{\delta}_{l+1} - \frac{\chi - x_{l+1}}{u_0^+(\rho_{\text{out}}, x_{l+1})}. \tag{3.5}$$

The boundary condition β_{k^*} takes finite values in a domain including $[\tilde{\tau}_1, \tilde{\tau}_2] \times \{\chi\}$ in which its temporal derivative is equal to q_{out}.

If ψ is not differentiable in ρ_{out}, the choice of β_{k^*} also leads to the equality of the derivatives of μ_l and $\mathbf{M}_{\beta_{k^*}}$ on the trajectory ζ_l, even though this choice may no longer be unique. □

Existence under compatibility conditions

Proposition 3.2 (Existence)**.** *If the internal value condition μ_l is affine, if the initial and upstream boundary conditions are piecewise affine and if the feasibility conditions of Definition 3.2 are satisfied, there exists a downstream boundary condition $\hat{\beta}$ solution of Problem 3.1. The algorithm exhibits a solution which is affine on the smallest interval $[\tau_1, \tau_2] \supset [\tilde{\tau}_1, \tilde{\tau}_2]$, representing a constant limitation of the maximum flow.*

Proof. The proof first searches for a potential solution $\hat{\beta}$ of Problem 3.1 which represents a constant limitation of the maximum flow during a time interval $[\tau_1, \tau_2]$. To achieve this goal, the algorithm searches for $\tau_1 \leq \tilde{\tau}_1$, $\tau_2 \geq \tilde{\tau}_2$, m and ρ such that $\hat{\beta}(t,\chi) = m + (t - \tau_1)\psi(\rho)$, $\forall t \in [\tau_1, \tau_2]$. Let $\hat{\beta}^*$ represent the restriction of $\hat{\beta}$ in $[\tau_1, \tau_2] \times \{\chi\}$ and let $\mathbf{M}_{\hat{\beta}^*}$ be the associated viability episolution. For $t \leq \tau_1$, there is no downstream constraint in the flow and $\hat{\beta}(t,\chi) = \min(\mathbf{M}_{\mathcal{M}_0}, \mathbf{M}_\gamma)(t,\chi)$ satisfies the requirements. For $t \geq \tau_2$, there is no limitation of the maximum flow at $x = \chi$. The flow at $x = \chi$ is given by $\min(\mathbf{M}_{\mathcal{M}_0}, \mathbf{M}_\gamma, \mathbf{M}_{\hat{\beta}^*})(t,\chi)$, it depends on the upstream, initial and upstream boundary condition $\hat{\beta}^*$.

From the results of Proposition 3.1, it follows that the choice $\rho = \rho_{\text{out}}$ is a valid choice. With this choice, the trajectory ζ_l is included in the domain (ii) of $\mathbf{M}_{\hat{\beta}^*}$. The function $\mathbf{M}_{\hat{\beta}^*}$ is affine in domain (ii) and its derivative along the trajectory ζ_l is equal to g_l.

• *Equation satisfied by τ_1 and τ_2*: In the domain (ii), the expression of $\mathbf{M}_{\hat{\beta}^*}$ is given by $\mathbf{M}_{\hat{\beta}^*}(t,x) = (t - \tau_1)q_{\text{out}} + (\chi - x)\rho_{\text{out}} + m$. The following condition on $\mathbf{M}_{\hat{\beta}^*}(\bar{\delta}_l, x_l)$ must be satisfied:

$$\mathbf{M}_{\hat{\beta}^*}(\bar{\delta}_l, x_l) = \mu_l(\bar{\delta}_l, x_l) = h.$$

This condition imposes a relation between τ_1, m and h:

$$(\bar{\delta}_l - \tau_1)q_{\text{out}} + (\chi - x_l)\rho_{\text{out}} + m = h. \tag{3.6}$$

The function $\tilde{h}$ is defined for $t \in [0, t_{\max}]$ by $\tilde{h}(t) = h - \rho_{\text{out}}(\chi - x_l) + (t - \bar{\delta}_l)q_{\text{out}}$. With this notation, (3.6) is written $m = \tilde{h}(\tau_1)$. The continuity of $\hat{\beta}$ at (τ_1, χ) imposes that $m = \min(\mathbf{M}_{\mathcal{M}_0}, \mathbf{M}_\gamma)(\tau_1, \chi)$ which leads to the following equation for $\tau_1 \in [0, \tilde{\tau}_1]$

$$\tilde{h}(\tau_1) - \min(\mathbf{M}_{\mathcal{M}_0}, \mathbf{M}_\gamma)(\tau_1, \chi) = 0. \tag{3.7}$$

The choice $\tau_2 = \tilde{\tau}_2$ satisfies the condition imposed to solve Problem 3.1. Note that larger values of τ_2 are possible, leading to a longer limitation of the maximum flow at $x = \chi$ but the smallest solution for τ_2 leads to the shortest limitation of the maximum flow at $x = \chi$. The observation of μ_l only provides a lower bound for the value of τ_2. If Equation (3.7) has a solution in the interval $[0, \tilde{\tau}_1]$, let τ_1 be the largest such solution. Section 3.4 presents an algorithm to compute this solution. Otherwise, let $\tau_1 = 0$. Specific feasibility conditions for the existence of a solution to Problem 3.1 are detailed in Definition 3.2. □

Proposition 3.3 (Feasibility conditions)**.** *The search for a piecewise affine limitation of the maximum flow implies that* $\beta^*(t,\chi) \leq \min(\mathbf{M}_{\mathcal{M}_0}, \mathbf{M}_\gamma)(t,\chi), \forall t \in [\tilde{\tau}_1, \tilde{\tau}_2]$ i.e.

$$\forall t \in [\tilde{\tau}_1, \tilde{\tau}_2],\ \min(\mathbf{M}_{\mathcal{M}_0}, \mathbf{M}_\gamma)(t,\chi) \geq \tilde{h}(t). \tag{3.8}$$

If there is no solution to (3.7) *in* $[0, \tilde{\tau}_1]$*, the feasibility conditions require the existence of* $\hat{x} \in [\xi, \chi]$ *such that the spatial derivative of* $\mathcal{M}_0$ *is* $-\rho_{out}$ *for* $(t,x) \in \{0\} \times [\hat{x}, \chi]$ *and such that* $\mathcal{M}_0(0,\hat{x}) = h - (\hat{x} - x_l)\rho_{out} - \bar{\delta}_l q_{out}$*.*

If these conditions are satisfied, the construction of ρ_{out}*,* τ_2*,* τ_1 *and* m *leads to a solution of Problem 3.1.*

Proof. This is true by construction. Let $\hat{\beta}^*(t,\chi)$ be defined for $t \in [\tau_1, \tau_2]$ by $\hat{\beta}^*(t,\chi) = m + (t - \tau_1)\psi(\rho_{\text{out}})$. The solution $\hat{\beta}$ takes finite values in $[0, t_{\max}] \times \{\chi\}$:

$$\hat{\beta}(t,x) = \begin{cases} \min(\mathbf{M}_{\mathcal{M}_0}, \mathbf{M}_\gamma)(t,\chi) \text{ if } t \leq \tau_1 \\ \hat{\beta}^*(t,\chi) \text{ if } t \in [\tau_1, \tau_2] \\ \min(\mathbf{M}_{\mathcal{M}_0}, \mathbf{M}_\gamma, \mathbf{M}_{\hat{\beta}^*})(t,\chi) \text{ if } t \geq \tau_2 \end{cases}.$$

The function defined as the minimum of $\mathbf{M}_{\mathcal{M}_0}$, $\mathbf{M}_\gamma$ and $\mathbf{M}_{\hat{\beta}}$ in the domain $[0, t_{\max}] \times [\xi, \chi]$ is a solution of the HJ-PDE (2.5). The compatibility conditions ensure that the boundary conditions are satisfied and the construction of $\hat{\beta}$ ensures that the function takes the same values as the internal condition μ_l for all (t,x) on the trajectory defined by ζ_l. □

3.4 Solution computation algorithm

This section presents an algorithm which computes the largest solution of (3.7) in the interval $[0, \tilde{\tau}_1]$ or proves that there is no solution on this interval. The algorithm leverages the inf-morphism property (Proposition 2.1) and the convexity of $\mathbf{M}_{\mathbf{c}_i}$ for any convex target function $\mathbf{c}_i$ [42].

Proposition 3.4 (Algorithm to compute τ_1)**.** *If* (3.7) *has a solution in* $[0, \tilde{\tau}_1]$*, its largest solution can be computed by solving a finite number of scalar convex optimization programs and scalar linear equations (Algorithm 1). If there is no solution in* $[0, \tilde{\tau}_1]$*, the same algorithm provides a proof that no solution exists.*

Algorithm 1 Algorithm for computing τ_1

Define $\underline{\kappa}_i$, κ_i^1 and κ_i^2 for $i \in \{1, \dots, I_0 + I_\gamma\}$,
$\mathcal{K} = \cup_i \{\underline{\kappa}_i, \kappa_i^1, \kappa_i^2\}$, $\tau_{\max} = \bar{\delta}_l - \frac{\chi - x_l}{u_0(\rho_{\text{out}})}$,
$\tau_{\min} = \max\{[0, \tau_{\max}) \cap \mathcal{K}\}$, $T = \emptyset$.
while $T == \emptyset$ **do**
  $I = \{i \in \{1, \dots, I_0 + I_\gamma\} : \underline{\kappa}_i \leq \tau_{\max}\}$
  for $i \in I$ **do**
    $\underline{n}_i = \mathbf{M}_{\mathbf{c}_i}(\tau_{\min}, \chi)$, $\underline{p}_i = \frac{\partial \mathbf{M}_{\mathbf{c}_i}}{\partial t}(\tau_{\min}^+, \chi)$, $\overline{n}_i = \mathbf{M}_{\mathbf{c}_i}(\tau_{\max}, \chi)$, $\overline{p}_i = \frac{\partial \mathbf{M}_{\mathbf{c}_i}}{\partial t}(\tau_{\max}^-, \chi)$
    if $n_i \leq \tilde{h}(\tau_{\min})$ **then**
      θ is the unique solution of $\mathbf{M}_{\mathbf{c}_i}(t, \chi) = \tilde{h}(t)$ on $[\tau_{\min}, \tau_{\max}]$.
      if $\mathbf{M}_{\mathbf{c}_i}(\theta, \chi) = \mathbf{M}_{\mathbf{c}}(\theta, \chi)$, $T = T \cup \{\theta\}$
    else if $\underline{n}_i + \underline{p}_i(\tau_{\max} - \tau_{\min}) \leq \tilde{h}(\tau_{\max})$ **and** $\overline{n}_i - \overline{p}_i(\tau_{\max} - \tau_{\min}) \leq \tilde{h}(\tau_{\min})$ **then**
      t^* is the largest minimizer of $\mathbf{M}_{\mathbf{c}_i}(t, \chi) - \tilde{h}(t)$ in $[\tau_{\min}, \tau_{\max}]$, $\delta = \mathbf{M}_{\mathbf{c}_i}(t^*, \chi) - \tilde{h}(t^*)$
      if $\delta \leq 0$ **then**
        θ is the unique solution of $\mathbf{M}_{\mathbf{c}_i}(t, \chi) = \tilde{h}(t)$ in $[t^*, \tau_{\max}]$
        if $\mathbf{M}_{\mathbf{c}_i}(\theta, \chi) = \mathbf{M}_{\mathbf{c}}(\theta, \chi)$, $T = T \cup \{\theta\}$
      end if
    end if
  end for
  $\tau_{\max} = \tau_{\min}$, $\tau_{\min} = \max\{[0, \tau_{\max}) \cap \mathcal{K}\}$
end while

Proof. According to the feasibility conditions (3.8), $\min(\mathbf{M}_{\mathcal{M}_0}, \mathbf{M}_\gamma)(\tilde{\tau}_1, \chi) \geq \tilde{h}(\tilde{\tau}_1)$. Let $\mathbf{c}_i$ denote the value condition i, *i.e.* $\mathbf{c}_i = \mathcal{M}_{0_i}$ if $i \leq I_0$ and $\mathbf{c}_i = \gamma_{i-I_0}$ if $i > I_0$. Let $\mathbf{c}$ be defined as $\mathbf{c} = \min_i \mathbf{c}_i$. The algorithm searches for the largest $t \leq \tilde{\tau}_1$ such that there exists $i \in \{1, \dots, I_0 + I_\gamma\}$ satisfying $\mathbf{M}_{\mathbf{c}_i}(t, \chi) = \tilde{h}(t) = \mathbf{M}_{\mathbf{c}}(t, \chi)$. If no such t exists, there is no solution to (3.7) in $[0, \tilde{\tau}_1]$, otherwise, this value of t is also the largest solution of (3.7) in $[0, \tilde{\tau}_1]$.

Let T represent the current set of solutions of (3.7), initialized to the empty set. The variable $\tau_{\max}$ is initialized as $\tau_{\max} = \tilde{\tau}_1$. The algorithm iteratively updates $\tau_{\max}$ such that, if $T = \emptyset$, there is no solution of (3.7) in $[\tau_{\max}, \tilde{\tau}_1]$, otherwise the algorithm terminates and the largest element of T is the largest solution of (3.7) in $[0, \tilde{\tau}_1]$. More precisely, $\forall t \in [\tau_{\max}, \tilde{\tau}_1], \forall i \in \{1, \dots, I_0 + I_\gamma\}$, $\mathbf{M}_{\mathbf{c}_i}(t, \chi) \geq \tilde{h}(t)$.

This condition is true at initialization $\tau_{\max}$ because of the compatibility condition (3.8). Each component $\mathbf{M}_{\mathbf{c}_i}$ can be computed explicitly [42]. There exists three domains in which the solution has a specific analytical expression. For $i \in \{1, \dots, I_0 + I_\gamma\}$, let $\underline{\kappa}_i$, κ_i^1 and κ_i^2 be such that $\underline{\kappa}_i \leq \kappa_i^1 \leq \kappa_i^2$ and correspond to the boundaries of the three different domains in $x = \chi$. Note that $\mathbf{M}_{\mathbf{c}_i}(t, \chi) = +\infty$ if and only if $t \leq \underline{\kappa}_i$ and $t \mapsto \mathbf{M}_{\mathbf{c}_i}(t, \chi)$ is affine on the interval $[\kappa_i^1, \kappa_i^2]$. For a given $\tau_{\max}$, $\tau_{\min}$ is defined by $\tau_{\min} = \max\{[0, \tau_{\max}) \cap \mathcal{K}\}$.

The solution $\mathbf{M}_{\mathbf{c}_i}$ associated with the convex target function $\mathbf{c}_i$ is convex [42] which implies the convexity of $t \mapsto \mathbf{M}_{\mathbf{c}_i}(t,\chi)$. Let $\underline{n}_i$ and $\overline{n}_i$ be defined as $\underline{n}_i = \mathbf{M}_{\mathbf{c}_i}(\tau_{\min},\chi)$ and $\overline{n}_i = \mathbf{M}_{\mathbf{c}_i}(\tau_{\max},\chi)$; $\underline{p}_i$ is the right derivative of $t \mapsto \mathbf{M}_{\mathbf{c}_i}(t,\chi)$ at $\tau_{\min}$ (denoted $\frac{\partial \mathbf{M}_{\mathbf{c}_i}}{\partial t}(\tau_{\min}^+,\chi)$) and $\overline{p}_i$ is the left derivative of $t \mapsto \mathbf{M}_{\mathbf{c}_i}(t,\chi)$ at $\tau_{\max}$ (denoted $\frac{\partial \mathbf{M}_{\mathbf{c}_i}}{\partial t}(\tau_{\max}^-,\chi)$). For $i \in \{1,\ldots,I_0+I_\gamma\}$, the algorithm inspects following conditions:

1. *If* $\underline{n}_i \leq \tilde{h}(\tau_{\min})$: The function $t \mapsto \mathbf{M}_{\mathbf{c}_i}(t,\chi) - \tilde{h}(t)$ is convex in $[\tau_{\min},\tau_{\max}]$ as the sum of two convex functions. It is negative at $\tau_{\min}$ and positive at $\tau_{\max}$. The function has a unique zero in $[\tau_{\min},\tau_{\max}]$, which is added to the set T if $\mathbf{M}_{\mathbf{c}_i}(\theta,\chi) = \mathbf{M}_{\mathbf{c}}(\theta,\chi)$.

2. *If* $\underline{n}_i + \underline{p}_i(\tau_{\max}-\tau_{\min}) \leq \tilde{h}(\tau_{\max})$ *and* $\overline{n}_i - \overline{p}_i(\tau_{\max}-\tau_{\min}) \leq \tilde{h}(\tau_{\min})$: The convex function $t \mapsto \mathbf{M}_{\mathbf{c}_i}(t,\chi) - \tilde{h}(t)$ is positive in $[\tau_{\min},\tau_{\max}]$ if and only if its minimum on this interval is positive. Since the function is convex it has a unique minimum δ reached on a closed interval. The upper bound of this interval is denoted t^*. If $\delta \leq 0$, there exists a unique zero in the interval $[t^*,\tau_{\max}]$ which is added to the set T if $\mathbf{M}_{\mathbf{c}_i}(\theta,\chi) = \mathbf{M}_{\mathbf{c}}(\theta,\chi)$.

3. *If none of the previous conditions is satisfied, the function* $t \mapsto \mathbf{M}_{\mathbf{c}_i}(t,\chi) - \tilde{h}(t)$ *is positive in* $[\tau_{\min},\tau_{\max}]$: Necessarily, the following inequality holds $\underline{n}_i > \tilde{h}(\tau_{\min})$ and at least one of the following conditions holds: (1) $\underline{n}_i + \underline{p}_i(\tau_{\max}-\tau_{\min}) > \tilde{h}(\tau_{\max})$ or (2) $\overline{n}_i - \overline{p}_i(\tau_{\max}-\tau_{\min}) > \tilde{h}(\tau_{\min})$. If the first condition holds, the function $t \mapsto \mathbf{M}_{\mathbf{c}_i}(t,\chi)$ is convex in $[\tau_{\min},\tau_{\max}]$ so $\mathbf{M}_{\mathbf{c}_i}(t,\chi) \geq \underline{n}_i + (t-\tau_{\min})\underline{p}_i$. Given that $\underline{n}_i > \tilde{h}(\tau_{\min})$, and $\underline{n}_i + \underline{p}_i(\tau_{\max}-\tau_{\min}) > \tilde{h}(\tau_{\max})$, the linear function $t \mapsto \underline{n}_i + \underline{p}_i(t-\tau_{\min})$ is greater than $\tilde{h}$ at $t=\tau_{\min}$ and $t=\tau_{\max}$ and thus, in the entire interval $[\tau_{\min},\tau_{\max}]$. It implies that $t \mapsto \mathbf{M}_{\mathbf{c}_i}(t,\chi) - \tilde{h}(t)$ is positive in $[\tau_{\min},\tau_{\max}]$. If the second condition holds, a similar reasoning implies that $t \mapsto \mathbf{M}_{\mathbf{c}_i}(t,\chi) - \tilde{h}(t)$ is positive in $[\tau_{\min},\tau_{\max}]$ which concludes the proof.

Stopping condition: After checking conditions 1, 2 and 3 above for all i, there are two possible cases:
• If $T = \emptyset$, the function $t \mapsto \mathbf{M}_{\mathbf{c}_i}(t,\chi) - \tilde{h}(t)$ is positive in $[\tau_{\min},\tau_{\max}]$ for all i. Set $\tau_{\max} = \tau_{\min}$. The property that $\mathbf{M}_{\mathbf{c}_i}(t,\chi) - \tilde{h}(t) \geq 0$, $\forall t \in [\tau_{\max},\tilde{\tau}_1]$ still holds. Update $\tau_{\min} = \max\{[0,\tau_{\max}) \cap \mathcal{K}\}$ and iterate.
• If $T \neq \emptyset$, its largest element is the largest solution of (3.7) in the interval $[\tau_{\min},\tau_{\max}]$ and thus in the interval $[0,\tilde{\tau}_1]$. The algorithm terminates. □

Remark 3.1 (Analytical solution of τ_1)**.** *In the intervals* $[\tau_{\min},\tau_{\max}]$ *such that* $\tau_{\min} \geq \kappa_i^1$ *and* $\tau_{\max} \leq \kappa_i^2$*, the function* $t \mapsto \mathbf{M}_{\mathbf{c}_i}(t,\chi)$ *is affine. Its minimum or zeros are computed by solving a scalar linear equation.*

3.5 Numerical implementation

Consider a concave Hamiltonian ψ, piecewise affine upstream and initial boundary conditions γ and $\mathcal{M}_0$. The upstream and initial boundary conditions simulate value conditions of a road segment. The section illustrates the importance of the resolution of Problem 3.1 to reconstruct capacity reductions in flow networks. Algorithm 1 solves the reconstruction problem and computes the corresponding solution of Problem 3.1.

Experimental setting

Consider given piecewise affine initial and upstream boundary conditions $\mathcal{M}_{0_i}$, $i \in \{1, \dots, I_0\}$ and γ_j, $j \in \{1, \dots, I_\gamma\}$. They are generated randomly for the numerical example of interest. In the context of traffic flows, this corresponds to information on vehicle counts at the upstream boundary of the road segment. Consider also an affine internal value condition μ that satisfies the compatibility conditions with the initial and upstream boundary conditions and represents a vehicle reporting information on a portion of its trajectory, during which its speed is considered constant. The computations are performed for two concave Hamiltonians (illustrated Figure 2.1), which are commonly used in transportation engineering. The numerical solution is computed using a toolbox developed for Matlab [163], which evaluates the exact solution numerically at any point (t, x) with a low computational cost.

Solution with piecewise affine initial and upstream boundary conditions and one affine internal condition

The solution of (2.5) is computed for the prescribed piecewise affine initial and upstream boundary conditions and the affine internal condition as the minimum of $\mathbf{M}_{\mathcal{M}_0}$, $\mathbf{M}_\gamma$ and $\mathbf{M}_\mu$ [42]. This solution does not take into account the fact that the internal value condition results from both the initial, upstream *and* downstream boundary conditions (even though not observed directly), resulting in a domain of null flow and density downstream of the internal value condition between θ_1 and θ_2 (Figure 3.4).

A strong motivation for solving Problem 3.1 is the following. Let $\overline{\beta}$ be the value of the solution of (2.5) in $[0, t_{\max}] \times \{\chi\}$ with the prescribed value conditions $\mathcal{M}_0$, γ and μ. The solution of (2.5) with prescribed value conditions $\mathcal{M}_0$, γ and $\overline{\beta}$ leads to a different solution, in particular one which does not coincide with μ, as shown in Figure 3.5. The motivation is also intuitive in the context of traffic flow engineering, where Figure 3.4 corresponds to having a vehicle suddenly breakdown when there is no obstacle in front of it. Slow downs are expected to be due to queues caused by downstream capacity reductions.

Resolving the domains of null flow and density

Solving Problem 3.1 is necessary to take into account the fact that the internal condition is not only caused by the initial and upstream conditions but also by the downstream con-

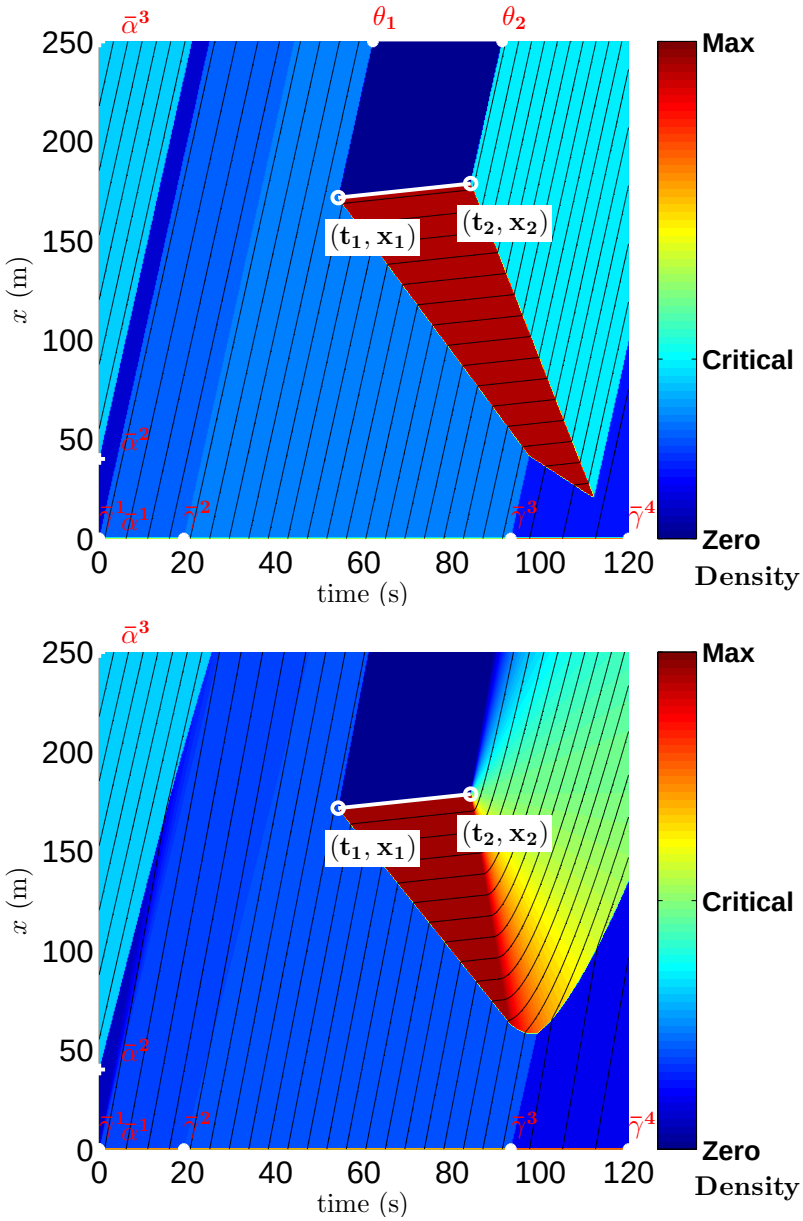


Figure 3.4: Solution of the *Moskowitz Hamilton-Jacobi partial differential equation* given initial and upstream piecewise affine boundary conditions and one affine internal value condition. The internal condition is imposed between $(\bar{\delta}_l, x_l)$ and $(\bar{\delta}_{l+1}, x_{l+1})$. **Top:** Solution computed for a triangular Hamiltonian. **Bottom:** Solution computed for a Greenshields Hamiltonian.

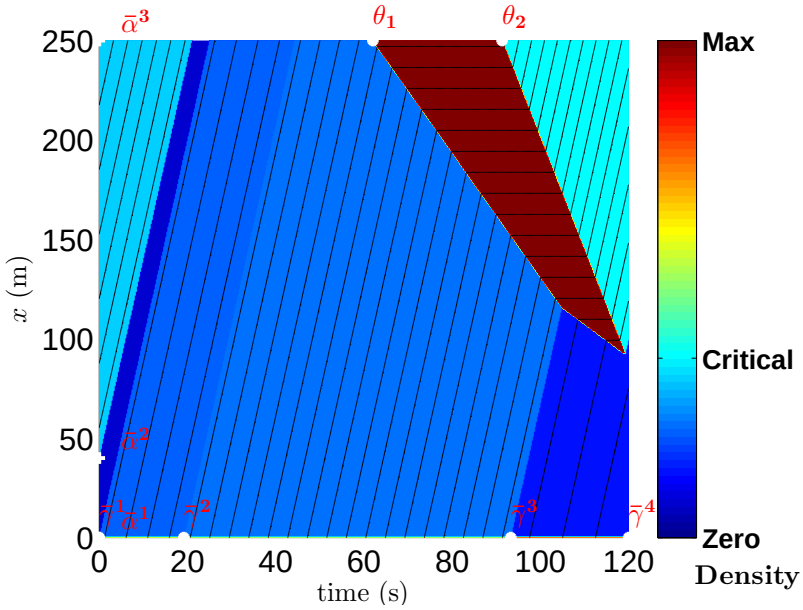


Figure 3.5: Solution of the *Moskowitz Hamilton-Jacobi partial differential equation* subject to initial, upstream and downstream value conditions before solving the boundary condition reconstruction problem. The downstream boundary condition imposed to the solution is $\bar{\beta}$, *i.e.* the value of the solution when the prescribed initial, upstream and internal value conditions are imposed to the solution.

dition. The problem is solved using Algorithm 1: it reconstructs a downstream boundary condition that "*caused*" the internal value condition. The algorithm computes a solution that represents a constant limitation of the maximum flow for a time interval $[\tau_1, \tau_2]$, as illustrated in Figure 3.6 for the two concave Hamiltonians. Note that the solution is unique (among the piecewise affine solutions) for an interval $[\tilde{\tau}_1, \tilde{\tau}_2]$ included in $[\tau_1, \tau_2]$ and that other downstream boundary conditions are possible out of this interval.

3.6 Conclusion and discussion

The chapter studied a reconstruction problem of downstream boundary conditions from Lagrangian sensing and prescribed upstream and initial conditions, with important applications in flow networks estimation and control. Under compatibility conditions, a downstream boundary condition representing a constant capacity drop can be reconstructed. The chapters presents a computationally efficient algorithm that numerically computes the solution.

The chapter discusses the uniqueness of the solution on specific domains, among piecewise affine boundary conditions. The choise of piecewise affine boundary conditions integrates the physical characteristics of the signal (succession of service and no-service times). The derivations assume that the noise and inaccuracies in the measurements are negligible. They

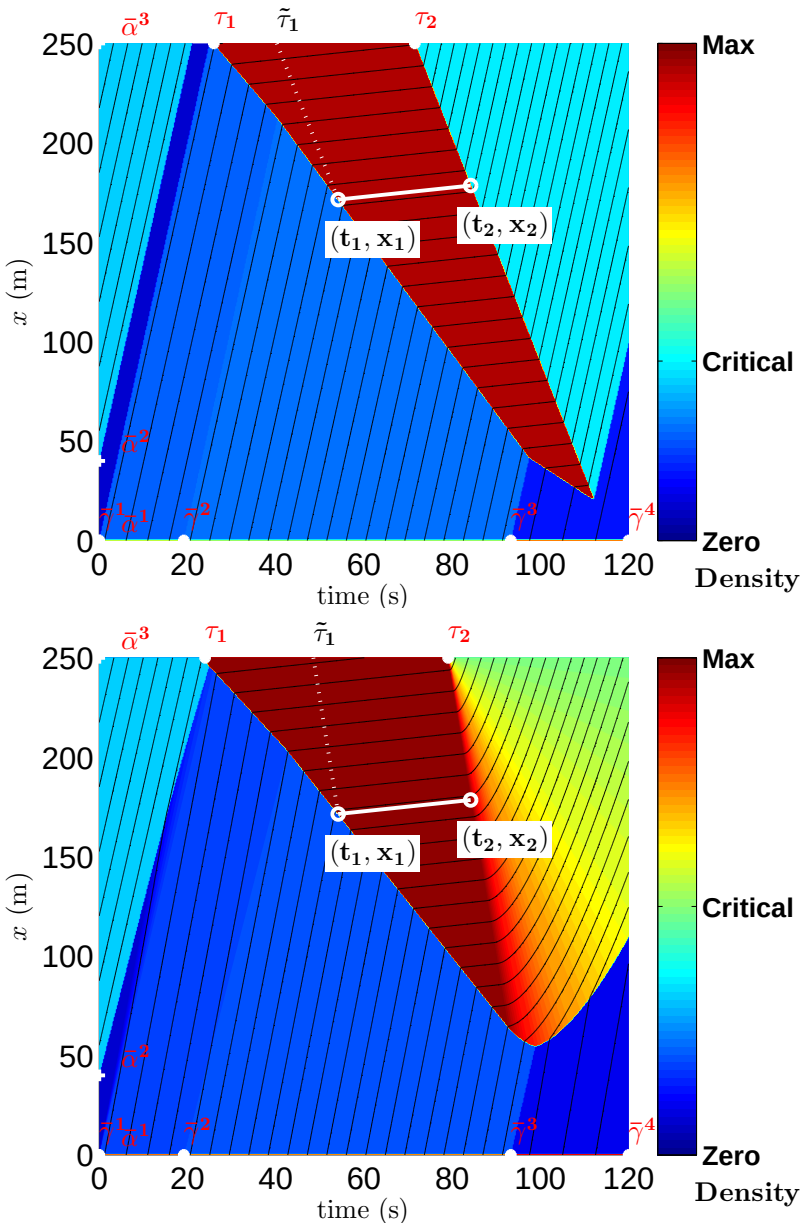


Figure 3.6: Solution of the reconstruction problem for the *Moskowitz Hamilton-Jacobi partial differential equation* given initial and upstream piecewise affine value conditions and one affine internal value condition. The internal value condition is prescribed between $(\bar{\delta}_l, x_l)$ and $(\bar{\delta}_{l+1}, x_{l+1})$. **Top:** Solution computed for a triangular Hamiltonian. **Bottom:** Solution computed for a Greenshields Hamiltonian.

also do not account for the discrepancies between the physical reality and its mathematical abstraction (mathematical model characterized by the partial differential equation). For this reason, it is interesting to take into account the uncertainty in the measurements and/or in the model. The following chapter analyzes a statistical estimation framework which extends the estimation capabilities by taking into account the uncertainty in the measurements.

Chapter 4

Characterization of the distribution of the solution under noisy measurements

Chapter 3 presented an algorithm to reconstruct boundary conditions based on internal value conditions imposed on the solution of a *Hamilton-Jacobi partial differential equation*. The results have valuable applications for estimation in flow networks, to detect capacity reductions, such as the one appearing in signalized networks.

The algorithm assumes that the discrepancy between the reality and its mathematical abstraction on one side, and between the measurements and the actual state of the system on the other side, are negligible. In numerous applications, this assumption does not hold. The example of Chapter 3 illustrates this limitation. As more noisy measurements are added, they constrain the solution of the problem, until the problem becomes unfeasible (because the compatibility conditions no longer hold). For these applications, initial, boundary and internal value conditions shall be regarded as random processes, rather than deterministic functions. Stochastic formulations of the HJ-PDE have been studied, in particular in the financial mathematics community [185, 67]. This research uses diffusion theory and in particular Itô's formula to show existence and uniqueness of the solution under certain conditions [67]. The research focuses on specific classes of stochastic HJ-PDE, such as backward differential equations, which is a different problem than what the thesis is focusing on.

The chapter derives the probability distribution of the solution of a class of HJ-PDEs subject to random value conditions. The derivations lead to analytical or semi-analytical expressions of the probability distribution function at any point in the domain in which the solution is defined. The characterization of the distribution of the solution at any point is a first step towards the estimation of the parameters defining the random value conditions.

As mentioned in Chapter 3, an application of interest consists in the design of estimation frameworks in flow networks, and in particular in signalized flow networks. In such networks, the derivations can be used to design reliable real-time traffic monitoring systems [4, 206]. Arterial traffic is inherently probabilistic. It is natural to consider probabilistic representations

of the internal, boundary and value conditions and to estimate the probability distribution of the macroscopic state variables (flow, density and velocity) at any location x and time t. For example, the speed of formation of the queue upstream of a signal or a bottleneck depends on the density of arrival vehicles. Uncertainties on the flow of arrival vehicles lead to uncertainties on the speed of queue formation, maximum length of the queue, time of the full queue dissipation and so on. The chapter allows us to integrate the uncertainty and to compute the probability distribution of the Moskowitz function, and thus of the other variables of interest (speed of formation of the queue, maximum queue length and so on).

The chapter is organized as follows. Section 4.1 extends the derivations of [41, 42] presented in Chapter 2. It derives the probability distribution of the solution to the HJ-PDE when the boundary conditions are random. The section indicates how these derivations can be used to estimate the parameters of value conditions statistically. The potential of the approach is illustrated through an example in traffic flow networks in Section 4.2. The section analyzes the effects of random capacity reductions on the dynamics of congestion.

4.1 Probability distribution of the solution of the Hamilton-Jacobi partial differential equation

The section generalizes the computation framework of Chapters 2 and 3 to take into account the randomness of the value conditions. It details the derivations in the case of an upstream or a downstream boundary condition. The derivations can easily be extended to initial and internal value conditions. The probability distribution of the solution subject to piecewise affine value conditions is computed from the derivations for each affine value conditions.

Random upstream boundary condition

Let γ_j be an upstream boundary condition , as defined by (2.8), *i.e.* defined on $[\bar{\gamma}_j, \bar{\gamma}_{j+1}] \times \{\xi\}$ by $\gamma_j(t,x) = d_j + (t - \bar{\gamma}_j)\psi(\rho_j)$. The main contribution of this chapter is to consider ρ_j as a random variable, with given distribution $p_{\rho_j}(\rho_j)$ and support included in $[0, \overline{\rho}_c]$. For any location (t,x), $\mathbf{M}_{\gamma_j}(t,x)$ is a random variable, the realization of which is fully determined by the realization of ρ_j. Let $\underline{\phi}_{t,x}$ be defined on $[0, \overline{\rho}_c]$ by $\underline{\phi}_{t,x} : \rho_j \mapsto \mathbf{M}_{\gamma_j}(t,x)(\rho_j)$.

Proposition 4.1 (Injectivity)**.** *There exists a unique $\underline{\rho}^*(t,x) \leq \underline{\rho}_c$ and a unique $\underline{\rho}^\diamondsuit(t,x) \leq \underline{\rho}^*(t,x)$ such that (i) the restriction of $\underline{\phi}_{t,x}$ to $[\underline{\rho}^*(t,x), \overline{\rho}_c]$ is constant and derived from Equation (2.12-ii), (ii) the restriction of $\underline{\phi}_{t,x}$ to $[0, \underline{\rho}^*(t,x)]$ is injective and is derived from (2.12-i) for $\rho \in [0, \underline{\rho}^\diamondsuit(t,x)]$ and from (2.12-iii) for $\rho \in [\underline{\rho}^\diamondsuit(t,x), \underline{\rho}^*(t,x)]$. The following properties hold (i) $\frac{x-\xi}{t-\bar{\gamma}_j} \in \partial_+\psi(\underline{\rho}^*(t,x))$ and (ii) $\frac{x-\xi}{t-\bar{\gamma}_{j+1}} \in \partial_+\psi(\underline{\rho}^\diamondsuit(t,x))$ if $t \geq \bar{\gamma}_{j+1} + \frac{x-\xi}{\nu^\flat}$ and $\underline{\rho}^\diamondsuit(t,x) = 0$ otherwise.*

Proof. In domain (ii), $\underline{\phi}_{t,x}$ is constant. The expression (ii) is valid for $t - \bar{\gamma}_j \leq \frac{x-\xi}{-u_0^-(\rho_j)}$, which restricts ρ_j algebraically. The concavity of ψ implies that $-u_0^-$ is non-increasing[1] on $[0, \underline{\rho}_c]$, with $-u_0^-(0) = \nu^\flat$ and $0 \in -u_0^-(\underline{\rho}_c)$. There exists a unique $\underline{\rho}^*(t,x) \in [0, \underline{\rho}_c]$ such that $\frac{x-\xi}{t-\bar{\gamma}_j} \in \partial_+\psi(\underline{\rho}^*(t,x))$. In particular, $\rho > \underline{\rho}^*(t,x) \Rightarrow -u_0^-(\rho) < \frac{x-\xi}{t-\bar{\gamma}_j}$ and $\rho < \underline{\rho}^*(t,x) \Rightarrow -u_0^-(\rho) > \frac{x-\xi}{t-\bar{\gamma}_j}$.
The concavity of $\underline{\phi}_{t,x}$ [43] and the definition of $\underline{\rho}^*(t,x)$ imply that $\underline{\phi}_{t,x}$ is strictly increasing on $[0, \underline{\rho}^*(t,x)]$.
• If $t \geq \bar{\gamma}_{j+1} + \frac{x-\xi}{\nu^\flat}$, there exists a unique $\underline{\rho}^\diamondsuit(t,x) \in [0, \underline{\rho}^*(t,x)]$ such that $\frac{x-\xi}{t-\bar{\gamma}_{j+1}} \in \partial_+\psi(\underline{\rho}^\diamondsuit(t,x))$. In particular, $\rho > \underline{\rho}^\diamondsuit(t,x) \Rightarrow -u_0^-(\rho) < \frac{x-\xi}{t-\bar{\gamma}_{j+1}}$ and $\rho < \underline{\rho}^\diamondsuit(t,x) \Rightarrow -u_0^-(\rho) > \frac{x-\xi}{t-\bar{\gamma}_{j+1}}$. Expression (iii) is valid for $\rho_j \in [0, \underline{\rho}^\diamondsuit(t,x)]$ and expression (i) is valid for $\rho_j \in [\underline{\rho}^\diamondsuit(t,x), \underline{\rho}^*(t,x)]$.
• If $t \leq \bar{\gamma}_{j+1} + \frac{x-\xi}{\nu^\flat}$, expression (i) is valid for $\rho_j \in [0, \underline{\rho}^*(t,x)]$ and it follows that $\underline{\rho}^\diamondsuit(t,x) = 0$. □

Proposition 4.2 (Bijection)**.** *The restriction of $\underline{\phi}_{t,x}$ to $[0, \underline{\rho}^*(t,x)]$ defines a bijection from $[0, \underline{\rho}^*(t,x)]$ to $[\underline{\phi}_{t,x}(0), \underline{\phi}_{t,x}(\underline{\rho}^*(t,x))]$. The expressions of $\underline{\phi}_{t,x}(0)$, $\underline{\phi}_{t,x}(\underline{\rho}^*(t,x))$ and $\underline{\phi}_{t,x}(\underline{\rho}^\diamondsuit(t,x))$ are computed analytically as follows:*

$$\underline{\phi}_{t,x}(0) = \begin{cases} d_j & \text{if } t \leq \bar{\gamma}_{j+1} + \frac{x-\xi}{\nu^\flat} \\ d_j + (t - \bar{\gamma}_{j+1})\varphi^*\left(\frac{\xi - x}{t - \bar{\gamma}_{j+1}}\right) & \text{if } t \geq \bar{\gamma}_{j+1} + \frac{x-\xi}{\nu^\flat} \end{cases}$$
$$\underline{\phi}_{t,x}(\underline{\rho}^*(t,x)) = d_j + (t - \overline{\gamma}_j)\varphi^*\left(\frac{\xi - x}{t - \bar{\gamma}_j}\right)$$
$$\underline{\phi}_{t,x}(\underline{\rho}^\diamondsuit(t,x)) = d_j + (\bar{\gamma}_{j+1} - \bar{\gamma}_j)\psi(\underline{\rho}^\diamondsuit(t,x)) + (t - \bar{\gamma}_{j+1})\varphi^*\left(\frac{\xi - x}{t - \bar{\gamma}_{j+1}}\right)$$

Proof. The proof is derived from Proposition 4.1 (injectivity of $\underline{\phi}_{t,x}$ on $[0, \underline{\rho}^*(t,x)]$) and Equation (2.12). □

Proposition 4.3 (Differentiability)**.** *If ψ is differentiable on $[0, \underline{\rho}^*(t,x)]$, the restriction of $\underline{\phi}_{t,x}$ to $[0, \underline{\rho}^*(t,x)]$ is differentiable.*

Proof. The expression of $\underline{\phi}_{t,x}$ imply that $\underline{\phi}_{t,x}$ is continuously differentiable on the intervals $[0, \underline{\rho}^\diamondsuit(t,x))$ and $(\underline{\rho}^\diamondsuit(t,x), \underline{\rho}^*(t,x)]$. If $t \leq \bar{\gamma}_{j+1} + \frac{x-\xi}{\nu^\flat}$, this terminates the proof as $\underline{\rho}^\diamondsuit(t,x) = 0$. The reminder of the proof considers the case where $t \geq \bar{\gamma}_{j+1} + \frac{x-\xi}{\nu^\flat}$

The differentiability of ψ at $\underline{\rho}^\diamondsuit(t,x)$ and the definition of $\underline{\rho}^\diamondsuit(t,x)$ imply that $\psi'(\underline{\rho}^\diamondsuit(t,x)) = \frac{x-\xi}{t-\bar{\gamma}_{j+1}}$. The left derivative of $\underline{\phi}_{t,x}$ at $\underline{\rho}^\diamondsuit(t,x)$ is computed using expression (2.12-i). The left derivative is given by $(t - \bar{\gamma}_j)\frac{x-\xi}{t-\bar{\gamma}_{j+1}} + \xi - x$, which can be written as $(x - \xi)\frac{\bar{\gamma}_{j+1} - \bar{\gamma}_j}{t - \bar{\gamma}_{j+1}}$. From expression (2.12-$iii$), it follows that the right derivative is equal to the left derivative and thus $\underline{\phi}_{t,x}$ is continuously differentiable on $[0, \underline{\rho}^*(t,x)]$. □

[1]Since u_0^- may not be uniquely defined, non-increasing is understood in the following sense: $\forall(\rho, \rho') \in [0, \overline{\rho}_c]^2$ s.t. $\rho < \rho'$, $\forall u_0^-(\rho) \in -\partial_+\psi(\rho)$, $\forall u_0^-(\rho') \in -\partial_+\psi(\rho')$, then $-u_0^-(\rho) \geq -u_0^-(\rho')$.

Proposition 4.4 (Diffeomorphism)**.** *If ψ is differentiable on $[0, \underline{\rho}^*(t,x)]$, the restriction of $\underline{\phi}_{t,x}$ to $(0, \underline{\rho}^*(t,x))$ defines a diffeomorphism from $(0, \underline{\rho}^*(t,x))$ to $\left(\underline{\phi}_{t,x}(0), \underline{\phi}_{t,x}(\underline{\rho}^*(t,x))\right)$.*

Proof. The proof relies on the global inversion theorem (see for example [8]). The above Propositions indicate that $\underline{\phi}_{t,x}$ is injective and continuously differentiable on the open interval $(0, \underline{\rho}^*(t,x))$. They also prove that the differential function is invertible on this interval. Since $\underline{\phi}_{t,x}$ is concave and strictly increasing on $(0, \underline{\rho}^*(t,x))$, the derivative is strictly positive on this interval, thus invertible and $\underline{\phi}_{t,x}$ defines a diffeomorphism on $(0, \underline{\rho}^*(t,x))$. □

The following definition, first introduced and proved in [42] is used to derive the expression $\underline{\phi}_{t,x}^{-1}$:

Definition 4.1 (Densities associated with v_j and g_j [42])**.** *Let v_j in $[0, \nu^\flat]$ and g_j be in $[0, \varphi^*(-v_j)]$. Let $\rho \in [0, \rho_{\max}]$ be such that*[2] *$\psi(\rho) - \rho v_j = \varphi^*(-v_j)$. There exists two solutions to the equation $\psi(\rho_j) - v_j\rho_j = g_j$ on $[0, \rho_{\max}]$, denoted $\rho_1(v_j, g_j)$ and $\rho_2(v_j, g_j)$ with $\rho_1(v_j, g_j) \leq \rho \leq \rho_2(v_j, g_j)$.*

Proposition 4.5 (Expression of $\underline{\phi}_{t,x}^{-1}$)**.** *The inverse of the diffeomorphism induced by the restriction of $\underline{\phi}_{t,x}$ to $(0, \underline{\rho}^*(t,x))$ onto its image is denoted $\underline{\phi}_{t,x}^{-1}$ and can be computed analytically as:*

$$\underline{\phi}_{t,x}^{-1}(m) = \rho_1\left(\frac{x-\xi}{t-\bar{\gamma}_j}, \frac{m-d_j}{t-\bar{\gamma}_j}\right) \tag{4.1}$$

Proof. For $\rho_j \in [0, \underline{\rho}^\diamondsuit(t,x)]$, the expression of $\underline{\phi}_{t,x}(\rho_j)$ is given by (iii). Let m be in the image of $[0, \underline{\rho}^\diamondsuit(t,x)]$ under $\underline{\phi}_{t,x}$, there exists a unique $\rho_j \in [0, \underline{\rho}^\diamondsuit(t,x)]$ such that:

$$\psi(\rho_j) = \frac{m - d_j - (t - \overline{\gamma}_{j+1})\varphi^*\left(\frac{\xi - x}{t - \overline{\gamma}_{j+1}}\right)}{\bar{\gamma}_{j+1} - \bar{\gamma}_j}. \tag{4.2}$$

This implies that the right hand side of (4.2) is in $[0, q_{\max}]$ and that the unique solution is given by

$$\underline{\phi}_{t,x}^{-1}(m) = \underline{\rho}\left(\frac{m - d_j - (t - \overline{\gamma}_{j+1})\varphi^*\left(\frac{\xi - x}{t - \overline{\gamma}_{j+1}}\right)}{\bar{\gamma}_{j+1} - \bar{\gamma}_j}\right),$$

where $\underline{\rho}$ is defined in Definition 2.3.

For $\rho_j \in [\underline{\rho}^\diamondsuit(t,x), \underline{\rho}^*(t,x)]$, the expression of $\underline{\phi}_{t,x}(\rho_j)$ is given by (i). Let m be in $\underline{\phi}_{t,x}([\underline{\rho}^\diamondsuit(t,x), \underline{\rho}^*(t,x)])$. There exists a unique $\rho_j \in [\underline{\rho}^\diamondsuit(t,x), \underline{\rho}^*(t,x)]$ such that

$$\frac{m - d_j}{t - \bar{\gamma}_j} = \psi(\rho_j) - \rho_j \frac{x - \xi}{t - \bar{\gamma}_j}.$$

[2]The existence of such a ρ comes from the definition of φ^*.

The following uses the fact that $\frac{x-\xi}{t-\bar{\gamma}_j} \in [0, -u_0^-(\rho_j)]$ and in particular that $\frac{x-\xi}{t-\bar{\gamma}_j} \in [0, \nu^\flat)$. The existence of ρ_j and the definition of φ^* also imply that $\frac{m-d_j}{t-\bar{\gamma}_j} \leq \varphi^*(-\frac{x-\xi}{t-\bar{\gamma}_j})$. The variable v_j is defined by $v_j = \frac{x-\xi}{t-\bar{\gamma}_j}$ and $g_j = \frac{m-d_j}{t-\bar{\gamma}_j}$. Let ρ be such that $\psi(\rho) - \rho v_j = \varphi^*(-v_j)$, then $v_j \in \partial_+\psi(\rho)$ and $-u_0^-(\underline{\rho}^*(t,x)) > v_j$. The concavity of ψ implies that $\rho > \underline{\rho}^*(t,x)$. According to Definition 4.1, there exists two solutions $\rho_1(v_j, g_j)$ and $\rho_2(v_j, g_j)$ to the equation $\psi(\rho_j) - \rho_j v_j = g_j$, satisfying $\rho_1(v_j, g_j) \leq \rho \leq \rho_2(v_j, g_j)$. Since $\rho > \underline{\rho}^*(t,x)$, only the first solution is possible which yields (4.1). □

The previous results are used to derive the probability distribution of $\mathbf{M}_{\gamma_j}(t,x)$. It is first necessary to define

$$\text{and} \quad \begin{aligned} w(t,x) &= \int_{\underline{\rho}^*(t,x)}^{\overline{\rho}_c} p_{\rho_j}(\rho_j)\, d\rho_j \\ \mathbf{M}_{\gamma_j}^{\text{init}} &= d_j + (t - \overline{\gamma}_j)\varphi^*\left(\tfrac{\xi - x}{t-\overline{\gamma}_j}\right) \end{aligned}.$$

Proposition 4.6 (Probability distribution of $\mathbf{M}_{\gamma_j}(t,x)$)**.** *If ψ is continuously differentiable on $[0, \underline{\rho}^*(t,x)]$, the probability distribution of $\mathbf{M}_{\gamma_j}(t,x)$ is given by*

$$\begin{aligned} p_{\mathbf{M}_{\gamma_j}(t,x)}(m) = w(t,x)\delta(m - \mathbf{M}_{\gamma_j}^{init}) \\ +(1 - w(t,x)) \left| \tfrac{d}{dm}\left(\underline{\phi}_{t,x}^{-1}(m)\right)\right| p_{\rho_j}(\underline{\phi}_{t,x}^{-1}(m)). \end{aligned}$$

Proof. Using the law of total probability, it follows that

$$\begin{aligned} p_{\mathbf{M}_{\gamma_j}(t,x)}(m) = w(t,x) p_{\mathbf{M}_{\gamma_j}(t,x)|\rho_j}(m|\rho_j \in [\underline{\rho}^*(t,x), \underline{\rho}_c]) \\ +(1 - w(t,x)) p_{\mathbf{M}_{\gamma_j}(t,x)|\rho_j}(m|\rho_j \in [0, \underline{\rho}^*(t,x)]) \end{aligned}.$$

Given the event "$\rho_j \in [\underline{\rho}^*(t,x), \underline{\rho}_c]$", the value of $\mathbf{M}_{\gamma_j}(t,x)$ is deterministic, and equal to $\mathbf{M}_{\gamma_j}^{\text{init}}$. The probability distribution of $\mathbf{M}_{\gamma_j}(t,x)$ conditioned on the event "$\rho_j \in [\underline{\rho}^*(t,x), \underline{\rho}_c]$" is a Dirac Delta distribution (mass probability) at $\mathbf{M}_{\gamma_j}^{\text{init}}$, which can be written $p_{\mathbf{M}_{\gamma_j}(t,x)|\rho_j}(m|\rho_j \in [\underline{\rho}^*(t,x), \underline{\rho}_c]) = \delta(m - \mathbf{M}_{\gamma_j}^{\text{init}})$.

Given the event "ρ_j is in $[0, \underline{\rho}^*(t,x)]$" and given that the restriction of $\underline{\phi}_{t,x}$ on this interval induces a diffeomorphism, the probability distribution of $\mathbf{M}_{\gamma_j}(t,x)$ is derived from the probability distribution of ρ_j using the change of variable $\rho_j = \underline{\phi}_{t,x}^{-1}(m)$. □

Random downstream boundary condition

As done for a random upstream boundary condition, the following derives the probability distribution of a component associated with a random downstream boundary condition.

The proofs are similar to the proofs for the derivations of the solution subject to a random upstream boundary condition.

Consider a downstream boundary condition β_k, defined on $[\bar{\beta}_k, \bar{\beta}_{k+1}] \times \{\chi\}$ by $\beta_k(t,x) = f_k + (t - \bar{\beta}_k)\psi(\rho_k)$, for which the parameter ρ_k is a random variable, with given distribution $p_{\rho_k}(\rho_k)$. For any location (t,x), $\overline{\phi}_{t,x}$ is defined on the interval $[\underline{\rho}_c, \rho_{\max}]$ by $\overline{\phi}_{t,x} : \rho_k \mapsto \mathbf{M}_{\beta_k}(t,x)(\rho_k)$.

Proposition 4.7 (Injectivity)**.** *There exists a unique $\overline{\rho}^*(t,x) \geq \underline{\rho}_c$ and a unique $\overline{\rho}^\diamondsuit(t,x) \geq \overline{\rho}^*(t,x)$ such that (i) the restriction of $\overline{\phi}_{t,x}$ to $[\overline{\rho}_c, \overline{\rho}^*(t,x)]$ is constant and derived from Equation (2.13-ii), (ii) the restriction of $\overline{\phi}_{t,x}$ to $[\overline{\rho}^*(t,x), \rho_{\max}]$ is injective and is derived from (2.13-i) for $\rho \in [\overline{\rho}^*(t,x), \overline{\rho}^\diamondsuit(t,x)]$ and from (2.13-iii) for $\rho \in [\overline{\rho}^\diamondsuit(t,x), \rho - \max]$. Moreover, the following properties hold: (i) $\frac{\chi - x}{t - \bar{\beta}_k} \in -\partial_+\psi(\overline{\rho}^*(t,x))$ and (ii) $\frac{\chi - x}{t - \bar{\beta}_{k+1}} \in -\partial_+\psi(\overline{\rho}^\diamondsuit(t,x))$ if $t \geq \bar{\beta}_{k+1} + \frac{\chi - x}{\nu^\sharp}$ and $\overline{\rho}^\diamondsuit(t,x) = \rho_{\max}$ otherwise.*

Proof. The proof is similar to the proof of Proposition 4.1 and omitted for brevity. □

Proposition 4.8 (Bijection)**.** *The restriction of $\overline{\phi}_{t,x}$ to $[\overline{\rho}^*(t,x), \rho_{\max}]$ defines a bijection from $[\overline{\rho}^*(t,x), \rho_{\max}]$ to $[\overline{\phi}_{t,x}(\rho_{\max}), \overline{\phi}_{t,x}(\overline{\rho}^*(t,x))]$. The expressions of $\overline{\phi}_{t,x}(\rho_{\max})$, $\overline{\phi}_{t,x}(\overline{\rho}^\diamondsuit(t,x))$ and $\overline{\phi}_{t,x}(\overline{\rho}^*(t,x))$ are computed analytically as follows:*

$$\overline{\phi}_{t,x}(\rho_{\max}) = \begin{cases} f_k & \text{if } t \leq \bar{\beta}_{k+1} + \frac{\chi - x}{\nu^\sharp} \\ f_k + (t - \bar{\beta}_{k+1})\varphi^*\left(\frac{\xi - x}{t - \bar{\beta}_{k+1}}\right) & \text{if } t \geq \bar{\beta}_{k+1} + \frac{\chi - x}{\nu^\sharp} \end{cases}$$

$$\overline{\phi}_{t,x}(\overline{\rho}^*(t,x)) = f_k + (t - \overline{\beta}_k)\varphi^*\left(\frac{\xi - x}{t - \bar{\beta}_k}\right)$$

$$\overline{\phi}_{t,x}(\overline{\rho}^\diamondsuit(t,x)) = f_k + (\bar{\beta}_{k+1} - \bar{\beta}_k)\psi(\overline{\rho}^\diamondsuit(t,x)) + (t - \bar{\beta}_{k+1})\varphi^*\left(\frac{\xi - x}{t - \bar{\beta}_{k+1}}\right)$$

Proof. The proof is derived from Proposition 4.7 (injectivity of $\overline{\phi}_{t,x}$) and Equation (2.12). □

Proposition 4.9 (Differentiability)**.** *If ψ is differentiable on $(\overline{\rho}^*(t,x), \rho_{\max}]$, the restriction of $\overline{\phi}_{t,x}$ to $(\overline{\rho}^*(t,x), \rho_{\max}]$ is differentiable.*

Proof. The proof is readily adapted from the proof of Proposition 4.3. It relies on the expression of $\overline{\phi}_{t,x}$ for $\rho_k \in [\overline{\rho}^*, \overline{\rho}^\diamondsuit(t,x))$ and for $\rho_k \in (\overline{\rho}^\diamondsuit(t,x), \rho_{\max}]$ given in (2.13) to show the differentiability on each of the two intervals and the continuity of the differential at $\rho = \overline{\rho}^\diamondsuit(t,x)$. □

Proposition 4.10 (Diffeomorphism)**.** *If ψ is differentiable on $(\overline{\rho}^*(t,x), \rho_{\max}]$, the restriction of $\overline{\phi}_{t,x}$ to $(\overline{\rho}^*(t,x), \rho_{\max})$ defines a diffeomorphism from $(\overline{\rho}^*(t,x), \rho_{\max})$ to $\left(\overline{\phi}_{t,x}(\rho_{\max}), \overline{\phi}_{t,x}(\overline{\rho}^*(t,x))\right)$.*

Proof. As for the proof of Proposition 4.4, the proof relies on the global inversion theorem and uses the injectivity and differentiability of $\overline{\phi}_{t,x}$ on the open interval $(\overline{\rho}^*(t,x), \rho_{\max})$, as well as the invertibility of the differential on this interval. □

Proposition 4.11 (Expression of $\overline{\phi}_{t,x}^{-1}$)**.** *The inverse of the diffeomorphism induced by the restriction of $\overline{\phi}_{t,x}$ to $(\overline{\rho}^*(t,x), \rho_{\max})$ onto its image is denoted $\overline{\phi}_{t,x}^{-1}$ and can be computed analytically as*

$$\overline{\phi}_{t,x}^{-1}(m) = \rho_2\left(\frac{\chi - x}{t - \bar{\beta}_k}, \frac{m - f_k}{t - \bar{\beta}_k}\right) \tag{4.3}$$

Proof. The proof is similar to the proof of Proposition 4.5 and omitted for brevity. □

Leveraging the previous results, the following derives the probability distribution of $\mathbf{M}_{\beta_k}(t,x)$. It is first useful to define

$$\text{and} \quad \begin{aligned} w(t,x) &= \int_{\overline{\rho}_c}^{\overline{\rho}^*(t,x)} p_{\rho_k}(\rho_k)\, d\rho_k \\ \mathbf{M}_{\beta_k}^{\text{init}} &= f_k + (t - \overline{\beta}_k)\varphi^*\left(\tfrac{\xi - x}{t - \bar{\beta}_k}\right) \end{aligned} .$$

Proposition 4.12 (Probability distribution of $\mathbf{M}_{\beta_k}(t,x)$)**.** *If ψ is continuously differentiable on $[\overline{\rho}^*(t,x), \rho_{\max}]$, the probability distribution of $\mathbf{M}_{\beta_k}(t,x)$ is given by*

$$\begin{aligned} p_{\mathbf{M}_{\beta_k}(t,x)}(m) &= w(t,x)\delta(m - \mathbf{M}_{\beta_k}^{init}) \\ &+ (1 - w(t,x))\left|\tfrac{d}{dm}\left(\overline{\phi}_{t,x}^{-1}(m)\right)\right| p_{\rho_k}(\overline{\phi}_{t,x}^{-1}(m)). \end{aligned}$$

Proof. The proof is similar to the proof of Proposition 4.6 and uses the law of total probability and the change of variable $\rho_k = \overline{\phi}_{t,x}^{-1}(m)$ for $m \in [\overline{\phi}_{t,x}(\rho_{\max}), \overline{\phi}_{t,x}(\overline{\rho}^*(t,x))]$. □

Probability distribution of the solution

The beginning of the section derives the probability distribution of a solution associated with an affine upstream or downstream boundary condition. Similar derivations are performed to compute the probability distribution of the solution associated with an affine initial or internal value condition, using the deterministic solution derived in [42] and defining appropriate conditioning and change of variables to derive the probability distribution of the solution. The inf-morphism property (Proposition 2.1) allows to compute the probability distribution of the solution associated with piecewise affine initial, upstream, downstream and internal value conditions. Consider I random value conditions $\mathbf{c}_i$ and denote by $p_{\mathbf{M}_{\mathbf{c}_i}(t,x)}$ the probability distribution of the corresponding component i at each location x and time t. The value conditions are assumed to be independent, and thus the random variables $\mathbf{M}_{\mathbf{c}_i}(t,x)$ are independent. Let $\mathbf{c}$ be the minimum of the value conditions $\mathbf{c}_i$, $i \in I$, the probability distribution of the solution at time t and location x associated with the random value condition $\mathbf{c}$ is denoted $\mathbf{M}_{\mathbf{c}}(t,x)$. For any realization of the random value condition, the solution

satisfies the inf-morphism property. The random variable $\mathbf{M}_{\mathbf{c}}(t,x)$ is the minimum of the random variables $\mathbf{M}_{\mathbf{c}_i}(t,x)$:

$$\mathbf{M}_{\mathbf{c}}(t,x) = \min_{i\in I} \mathbf{M}_{\mathbf{c}_i}(t,x)$$

Let $\mathbf{P}_{\mathbf{M}_{\mathbf{c}_i}(t,x)}$ denote the cumulative probability distribution of the random variable $\mathbf{M}_{\mathbf{c}_i}(t,x)$ associated with the random value condition $\mathbf{c}_i$. It is defined by: $\mathbf{P}_{\mathbf{M}_{\mathbf{c}_i}(t,x)}(m) = \int_{-\infty}^{m} p_{\mathbf{M}_{\mathbf{c}_i}(t,x)}(\tilde{m})\, d\tilde{m}$.

Proposition 4.13 (Probability distribution of $\mathbf{M}_{\mathbf{c}}(t,x)$)**.** *The probability distribution of the solution $\mathbf{M}_{\mathbf{c}}(t,x)$, corresponding to the random value condition $\mathbf{c}$ is computed from the probability distribution of the components $\mathbf{M}_{\mathbf{c}_i}(t,x)$ associated with the affine value conditions $\mathbf{c}_i$ as follows:*

$$p_{\mathbf{M}_{\mathbf{c}}(t,x)}(m) = \sum_{i\in I} p_{\mathbf{M}_{\mathbf{c}_i}(t,x)}(m) \prod_{j\neq i} (1 - \mathbf{P}_{\mathbf{M}_{\mathbf{c}_j}(t,x)}(m))$$

Remark 4.1 (Parameter estimation)**.** *The probability distribution of the solution of the HJ-PDE at time t and location x is parametric. The parameters characterize the probability distribution of the initial, upstream, downstream and internal value conditions. They can be estimated from (noisy) measurements of the solution, using likelihood maximization for example.*

4.2 Numerical implementation

A simulation of probabilistic traffic flows using the *Moskowitz* function [169, 172, 56] illustrates the derivations of Section 4.1. Consider a concave Hamiltonian ψ, initial and upstream boundary conditions, specified in the form of two piecewise affine functions taking finite values on the domains $\{0\} \times [\xi, \chi]$ and $[0, T] \times \{\xi\}$ respectively. A reduction of the output capacity at $x = \chi$, leading to the potential formation of a queue is simulated during time interval $[\bar{\beta}_0, \bar{\beta}_1]$. The reduction of the output capacity is represented by a random variable ρ_0 with support in $[\underline{\rho}_c, \rho_{\max}]$, corresponding to the density at the intersection during the capacity reduction. The randomness of the downstream boundary condition leads to randomness in the queue formation, which is illustrated numerically.

The numerical analysis considers a *Greenshields* Hamiltonian, as illustrated Figure 2.1 (right). The Greenshields Hamiltonian is defined on $[0, \rho_{\max}]$ by $\psi(\rho) = 4\frac{q_{\max}}{\rho_{\max}^2}\rho(\rho_{\max} - \rho)$. The expression is parameterized by the maximum density $\rho_{\max} = 0.1$ veh/m and the maximum flow $q_{\max} = 1300$ veh/h. The solution of the HJ-PDE is computed on the domain $[0, T] \times [\xi, \chi]$ with $T = 80$ s, $\xi = 0$ m and $\chi = 100$ m. The numerical analysis investigates a random capacity reduction during the time interval $[\bar{\beta}_0, \bar{\beta}_1]$ at $x = \xi$, with $\bar{\beta}_0 = 20$ s and $\bar{\beta}_1 = 50$ s. During this time interval, the output density is a random variable, ρ_0 uniformly distributed on $[0.08, 0.1]$.

Figure 4.1 shows the deterministic solution of the Moskowitz HJ-PDE for output densities ρ_0 equal to 0.08, 0.09 and 0.1 veh/m, corresponding to maximum output flows $\psi(\rho_0)$

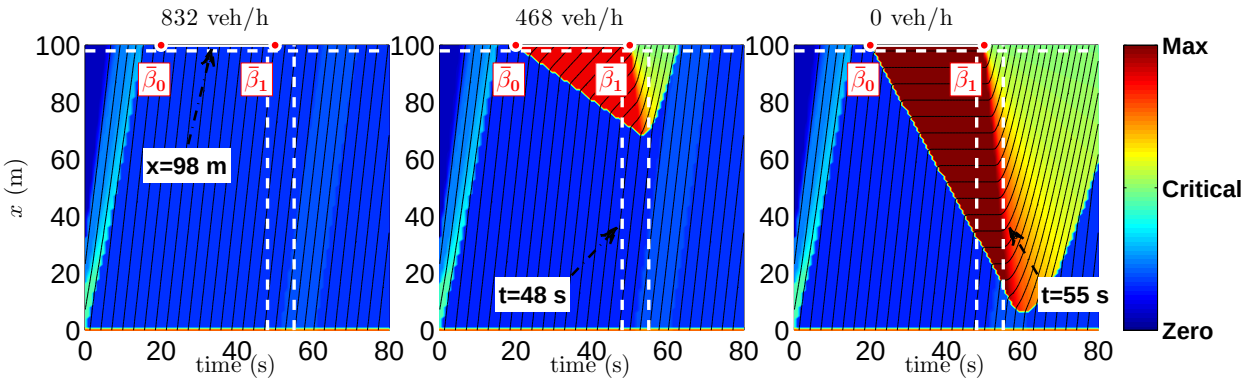


Figure 4.1: Deterministic solution of the *Moskowitz Hamilton-Jacobi partial differential equation* under given initial and upstream boundary conditions and with three different values for the capacity reduction. From left to right, the outflow is limited to 832, 468 and 0 vehicles per hour respectively. The color scale represents the spatial derivative of the solution (density). Black lines represent the isolines of the solution (
emphi.e. vehicle trajectories). The different values of the maximum outflow influence the formation of a queue upstream of the capacity reduction.

equal to 832, 468 and 0 veh/h respectively. The Figure displays isolines of the Moskowitz function and a colormap of the spatial partial derivative, which is a common two dimensional representation of the solution. The figure illustrates the differences in the solution of the HJ-PDE, under different downstream boundary conditions and underlines the importance to study the probability distribution of the solution when the conditions are noisy or cannot be estimated accurately. Depending on the importance of the capacity reduction, the solution may exhibit shock-waves, corresponding to discontinuities in the density ρ (and in the flow $\psi(\rho)$). In the context of transportation, it is common to refer to these shock-waves as queue formations or queue dissipations. Note that a flow of 832 veh/h does not create any queue formation because the capacity at time t and location χ is greater than the flow imposed by the initial and upstream boundary conditions at time t and location χ. As the maximum flow decreases, a queue forms. The speed of formation of the queue depends on the importance of the capacity reduction. At the end of the capacity reduction, the queue dissipates.

The probability distribution of the solution is computed according to the derivations of Section 4.1 and illustrated in Figures 4.2 and 4.3 using percentiles. For the random variable $\mathbf{M}(t,x)$, the n^{th}-percentile, denoted $\mathbf{M}^n(t,x)$ for $n \in [0,100]$, satisfies $\mathbf{P}_{\mathbf{M}(t,x)}(m \leq \mathbf{M}^n(t,x)) = n/100$. Percentiles are commonly used to represent probability distributions. Figure 4.2 illustrates the probability distribution at a fixed location $x = 98$ m (upstream of the end of the segment), as it evolves over time. The location is indicated in Figure 4.1, with a dashed line labeled $x = 98$ m. The slope of each curve represents the flow at the corresponding time and location. Points where the curve is not differentiable correspond to the presence of a shock-wave at the corresponding time and location. At the beginning

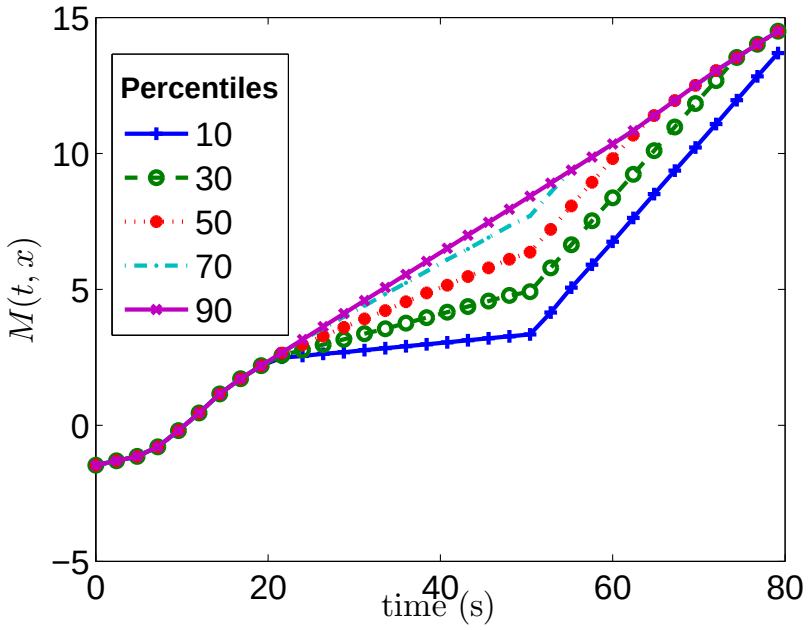


Figure 4.2: Distribution of the solution of the *Moskowitz Hamilton-Jacobi partial differential equation* at a fixed location, upstream of the capacity reduction. The figure represents the solution at $x = 98$ m (2 meters upstream of the capacity reduction). The value at $t = 0$ represents the label of the vehicle at $x = 98$ at the initial time, which is determined up to a constant chosen by the initialization of the Moskowitz function at $(\xi, 0)$.

of the capacity reduction, the flow decreases when the capacity reduction is sufficient to cause the formation of a queue. This creates a shock-wave, *i.e.* a non-differentiability of the solution. At the end of the capacity reduction, the queue dissipates (second shock-wave), corresponding to another non-differentiability of the solution. The duration of the congestion varies depending on the importance of the capacity reduction. Figure 4.3 represents the probability distribution of the solution at two time instances, $t = 48$ s and $t = 55$ s. The time instances are indicated in Figure 4.1, with dashed lines labeled $t = 48$ s and $t = 55$ s respectively. The figure illustrates the distribution of the queue at the specified times. Recall the relation between the temporal derivative of $\mathbf{M}$ and the flow: $\frac{\partial \mathbf{M}(t,x)}{\partial t} = q(t,x)$. It follows that the slope of each curve corresponds to the density of the solution at the specified time and location.

4.3 Conclusion and discussion

The chapter shows the importance to consider randomness of measurements and inaccuracies of the mathematical modeling of physical phenomena. It derives the probability distribution of the solution to a *Hamilton-Jacobi partial differential equation* for which the prescribed

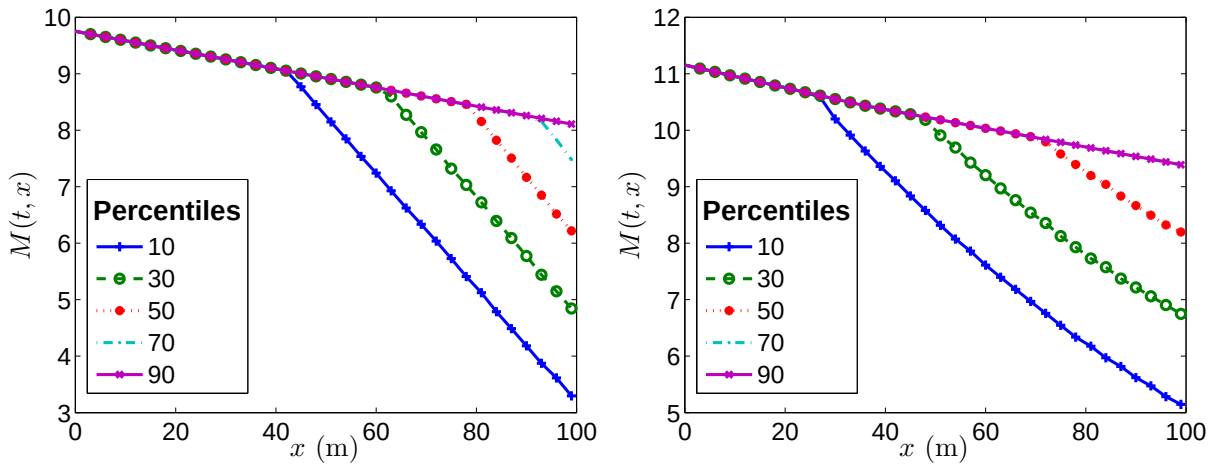


Figure 4.3: Distribution of the solution of the *Moskowitz Hamilton-Jacobi partial differential equation* at two distinct fixed times. **Left:** Solution at $t = 48$ seconds, 28 seconds after the beginning of the capacity reduction. **Right:** Solution at $t = 55$ seconds, 5 seconds after the end of the capacity reduction.

value conditions are probabilistic. The derivations allow for the analysis of the effects of the randomness of the value conditions on the solution, as illustrated in Section 4.2. Another application of this approach is the estimation of the parameters characterizing the probability distribution of the value conditions, through *maximum-likelihood* estimation for example. The chapter also introduces the necessary derivations to statistically estimate the parameters characterizing the distribution of value conditions. The derivations lead to an instantaneous computation of the distribution given the distribution of the value condition without the need for sampling, simulation, or computation on a fixed grid.

This work has important applications for systems described by a *Hamilton-Jacobi* equations for which the value conditions are noisy or cannot be measured accurately. The chapter illustrates the applicability in the context of traffic flows with random capacity reduction, leading to probabilistic congestion and queue formations. It is of particular interest for horizontal queuing networks with random capacity reductions. It provides a framework to take into account the uncertainty in the estimation process while leveraging the physical properties of the system.

A characteristic of many signalized networks is the alternation of service times and no-service times, in a periodical fashion. This property is not exploited in the model and as a result, the model requires measurements during each no-service time to detect the characteristics of the signal. This level of data is not always available. The following chapters analyze how the periodicity of numerous signalized networks can be leveraged to limit the data requirements to develop estimation capabilities.

Chapter 5

Statistical model of horizontal queue dynamics

The estimation framework of Chapters 3 and 4 captures specific features of the physics of queues. It integrates the measurements to refine the quality of the estimates at each location x and time t. Chapter 4 introduces the capability to account for noise in the measurements. However, to produce estimates, the model relies on strong assumptions regarding the *data availability*.

We now revisit the numerical example of Chapter 3. The probe data is used to infer the characteristics of the capacity reduction (start, end, maximum flow during the capacity reduction) and estimate the state of the queue at each location and time. However, if no probe measurement is available, the capacity reduction cannot be detected. Moreover, most probe data available today, with the prospect of global coverage in the near future, does not have the level of detail assumed in Chapters 3 and 4. In these chapters, it is assumed that probe data is available as piecewise affine trajectories, which provide information on the location and duration of stops. Most probe vehicles report their location at low sampling frequencies (once every minute). The model of Chapters 3 and 4 does not leverage a critical piece of information. In numerous applications, signals are periodic, with service times and no-service times (*e.g.* green and red times in an arterial network) succeeding each other.

The present chapter describes how to leverage this insight to develop an aggregated model of the dynamics: the model describes the dynamics over the duration of a cycle in a probabilistic setting. The arrival time of a vehicle within the cycle is a random variable. The temporal aggregation reduces the ability of the model to retrieve specific events: the model cannot estimate the exact time at which service started or ended. Instead, it estimates the typical duration of service times and no-service times. This aggregation limits the data requirements to develop estimation platforms.

As mentioned earlier, urban traffic dynamics are driven by the presence of traffic signals, which lead to important vehicle-to-vehicle travel time variability. The present chapter introduces a horizontal queuing theory model to derive an analytic expression of the probability distribution of travel time, parameterized by physical parameters (signals, free flow speed

distribution, queue lengths). The main contribution of the chapter is the design of modeling and estimation frameworks which address the specificities of sparsely sampled probe vehicle data. First, the distributions are derived between any two locations on a link as the locations reported by the vehicles are not necessarily at the beginning and at the end of the links. Second, a travel time allocation algorithm allows us to incorporate measurements when vehicles traverse several links between successive location reports and learn the parametric distribution of travel time. The allocation algorithm relies on the proof that the derived travel time distributions are mixtures of log-concave distributions. Numerical experiments using probe data show that the derived distribution more accurately represents the empirical distribution of travel time than other common distributions. These estimation capabilities can be further improved by integrating prior information on the physical parameters characterizing the distribution.

This chapter is organized as follows. Section 5.1 summarizes horizontal queuing theory results, developed over the past fifty years. These results are used to derive delay (Section 5.2) and travel time (Section 5.3) distributions between any two points on an arterial link. The chapter proves that the derived travel time distributions are mixtures of log-concave distributions and use this property in Section 5.4 to develop a machine learning algorithm to estimate traffic parameters and levels of congestion on each link of the network. The estimation capabilities of the algorithm in Section 5.5 using data collected using *Sensys* hardware [98].

5.1 Horizontal queuing theory

Traffic modeling

This section makes assumptions on the dynamics of traffic flow. The assumptions represent trade-offs between the model complexity and the information which can be extracted from the data. As typical penetration and sampling rates of probe vehicles remain low (positions reported on average once per minute), the model aims at estimating trends in traffic conditions, while keeping a realistic physical description of the dynamics of traffic flows. For a congestion state, the model represents the variability of delays and travel times due to the presence of signals. The parameters can be learned from travel times between arbitrary locations on the network. The assumptions are as follows.

1. *Macroscopic LWR model*: This is a common assumption introduced and used in Chapters 2 -4. The *Hamiltonian* (or *fundamental diagram*) is assumed to be *triangular*, as done previously in [58, 88, 172]. The diagram was previously introduced in Chapter 2. Its analytic expression is given in Equation (2.2) and it is illustrated Figure 2.1. The model implies that, as queue dissipates (*e.g.* as the signal turns green), vehicles are released with the maximum flow—capacity $q_{\max}$ and critical density $\rho_c = q_{\max}/v_f$.

2. *Discrete time dynamical system*: during a time interval, the parameters of the model (red time R, cycle time C, driving behavior) and the state of the system (queue length) are constant. The variability of the queue length affects the link travel time distribution [216]. However, the chapter focuses on estimation when measurements are *sparse* and may not be able to capture this phenomenon. The chapter does not detail the effect of turn movements and dedicated lanes. A potential approach would be to model each lane as a different queue with its specific parameters [200]. The parameters can be estimated from the probe vehicles with the corresponding turn movement (known from the path of the vehicle). As for other modeling choices, the decision to take into account dedicated lanes is a *trade-off between the model complexity and the level of information contained in the data*.

3. *Uniform vehicle arrivals*: over the time discretization, the arrival density has a constant value $\rho_a \leq \rho_c$. It is common to consider a time invariant distribution of arrivals such as a Poisson distribution, assuming that the effect of light synchronization on the arrival rate is negligible. The main source of vehicle to vehicle travel time variability is the presence of a signal rather than the randomness of the arrival rate, as can be observed in the distributions derived by [224]. The assumption of constant arrivals enables analytic derivations of the travel time pdf between *arbitrary locations* on the network, which is necessary to leverage data sent by sparsely sampled probe vehicles. In [15], the assumption of constant arrivals is relaxed to explicitly take into account signal coordination. It is possible to include these results in the model derived in the present chapter. However, the analytic expressions become cumbersome and are not presented here. They also induce a larger number of parameters which potentially increases the risk of over-fitting.

4. *Model for differences in driving behavior*: the free flow pace (inverse of the free flow speed) is a random variable with pdf φ^p, parameterized by θ_p (*e.g.* a Gamma distribution with $\theta_p = (\bar{p}_f, \sigma_p)^T$ where $\bar{p}_f$ and σ_p are the mean and the standard deviation of the random variable[1]). Features of driving behavior have been studied in the literature [141, 39, 204]. and are particularly important for highway traffic modeling and estimation. For arterials, the main source of travel time variability is the presence of traffic signals which leads to the formation and dissolution of queues and causes variability in the delay experienced by different vehicles. However, existing driving

[1]Note that Gamma distributions are usually parameterized using a shape parameter and a scale parameter or a shape parameter α and inverse scale parameter β. According to the later parametrization, the probability distribution of a Gamma random variable reads:

$$f(x;\alpha,\beta) = \beta^\alpha \frac{1}{\Gamma(\alpha)} x^{\alpha-a} e^{-\beta x} \quad \text{for } x \geq 0,\ \alpha,\ \beta > 0.$$

According to this parameterization, the mean of the Gamma random variable is given by α/β and its variance is α/β^2.

behavior models can be incorporated by modifying the probability distribution of free flow pace accordingly.

Horizontal queuing theory

Traffic is driven by the formation and the dissipation of queues upstream of locations where demand exceeds capacity (bottlenecks). Two discrete traffic regimes: *undersaturated* and *congested* represent different dynamics of the arterial link depending on the presence (respectively the absence) of a remaining queue when the light switches from green to red. The assumptions lead to an exact analytic solution of the LWR model. Figure 5.1 illustrates the corresponding vehicle trajectories for both regimes. The speed of formation and dissolution of the queue are respectively called v_a and w. Their expression is derived from the Rankine-Hugoniot condition [71] by

$$v_a = \frac{\rho_a v_f}{\rho_{\max} - \rho_a} \quad \text{and} \quad w = \frac{\rho_c v_f}{\rho_{\max} - \rho_c}. \tag{5.1}$$

Undersaturated regime. In this regime, the queue fully dissipates within the green time. This queue is called the *triangular queue* (from its triangular shape on the space-time diagram of trajectories). It is defined as the spatio-temporal region where vehicles are stopped on the link. Its length is denoted l and is computed as follows:

$$l = R\frac{wv_a}{w - v_a} = R\frac{v_f}{\rho_{\max}}\frac{\rho_c \rho_a}{\rho_c - \rho_a}. \tag{5.2}$$

Congested regime. In this regime, there exists a part of the queue downstream of the triangular queue called *remaining queue* with length l_r corresponding to vehicles which have to stop multiple times before going through the intersection.

Periodicity of the two regimes. The assumptions made earlier imply the C-periodicity of the queue dynamics. In particular, the congested regime is exactly at *saturation*: the number of vehicles entering and exiting the link are equal. Saturation is an idealized notion that is considered valid for each discretization interval. The difference between the number of vehicles entering and exiting the link is accounted for in the variation of the queue length between discretization intervals. This effect is studied in Chapter 6 using a model of traffic dynamics and congestion propagation on the network. At saturation, the arrival density is $\rho_a^s = \frac{C-R}{C}\rho_c$. The triangular queue length at saturation l_s is computed by replacing $\rho_a = \rho_a^s$ in equation (5.2) or by noticing that the number of vehicles that stop in the queue ($l_s \rho_{\max}$) is equal to the number of vehicles that exit the link in the duration of a cycle ($(C-R)v_f\rho_c$):

$$l_s = v_f \rho_c (C - R)/\rho_{\max}. \tag{5.3}$$

Note that l_s is the distance traveled between successive stops on a congested link.

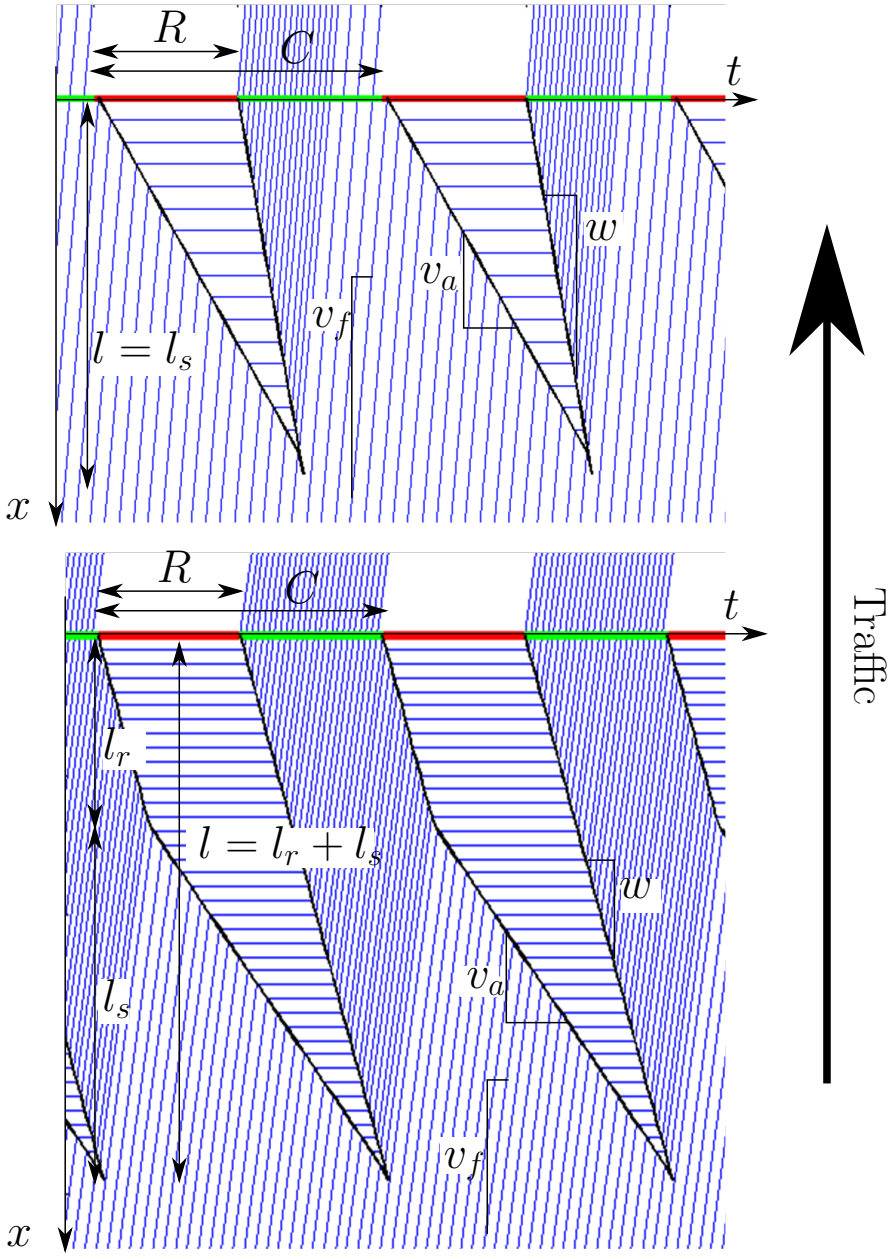


Figure 5.1: Space time diagram of vehicle trajectories with uniform arrivals under an under-saturated traffic regime (top) and a congested traffic regime (bottom).

Notation

The list below summarizes the set of variables which is sufficient to characterize the model of horizontal queues. The parameters are specific for each network link j. The index j is omitted for notation simplicity. The *model parameters* are:

- the free flow pace, p_f (seconds/meter), with pdf φ^p, parameterized by θ_p,
- the cycle time, C,
- the red time, R,
- the length of the link, L and
- the queue length at saturation l_s.

The *traffic state* is represented by the queue length (back of the queue), l, where l is the length of the triangular queue in the undersaturated regime and $l = l_s + l_r$ in the congested regime. The remaining queue length, l_r, is null in the undersaturated regime. Note that the length of the link is not estimated as it is given by the network topology.

The aforementioned parameters can be estimated with probe vehicle measurements, as detailed in the remainder of this chapter. In the following, the location x on a link corresponds to the distance to the *downstream* intersection. This setting appears naturally in the derivations because the formation of queues starts from the downstream intersection. The following sections derive probability distributions for the delay δ_{x_1,x_2} and the travel time y_{x_1,x_2} between two locations x_1 and x_2 on a link of the network, noted respectively $h(\delta_{x_1,x_2})$ and $g(y_{x_1,x_2})$. Additional indexing of these functions will appear as necessary for clarity. For notational simplicity, the derivations do not make an explicit distinction between a random variable X and its realization x.

5.2 Probability distribution of delay

Delays on arterial networks is mainly conditioned on two factors: (i) differences between the demand (number of vehicles which travel on a link) and the service (number of vehicle which go through an intersection during a cycle) dictate the level of congestion (indicated by the queue length) experienced by all the vehicles entering the link; (ii) the entrance time determines the duration of the delay in the queue due to the periodic dynamics imposed by the traffic signal (vehicle-to-vehicle variability [175]). Travel time measurements come from vehicles sampled uniformly in time. They send tuples of the form (x_1, t_1, x_2, t_2) where x_1 is the location of the vehicle at t_1 and x_2 is the position of the vehicle at t_2. This is representative of probe data available today. The travel time $t_2 - t_1$ is typically in the order of one minute.

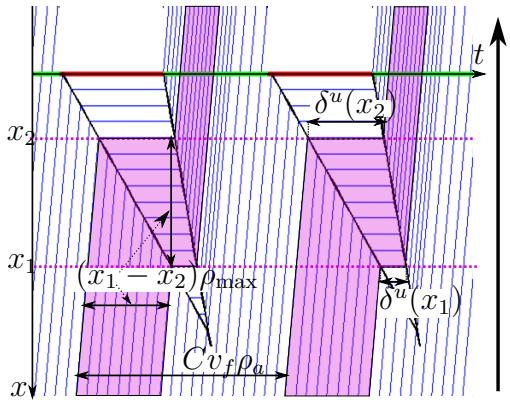


Figure 5.2: Proportion of delayed vehicles between two locations on a link: $\eta^u_{x_1,x_2}$ is the ratio between the number of vehicles joining the queue between x_1 and x_2 over the total number of vehicles entering the link in one cycle. The highlighted trajectories represent the trajectories of vehicles delayed between x_1 and x_2.

Probability distribution of delay in the undersaturated regime

Let $\eta^u_{x_1,x_2}$ be the fraction of the vehicles entering the link in a cycle that experience a delay between x_1 and x_2. The remainder of the vehicles travels from x_1 to x_2 without delay. The proportion $\eta^u_{x_1,x_2}$ is the ratio of vehicles joining the queue between x_1 and x_2 over the total number of vehicles entering the link in one cycle (Figure 5.2, left). The number of vehicles joining the queue between x_1 and x_2 is the number of vehicles stopped between x_1 and x_2: $\big(\min(l, x_1) - \min(l, x_2)\big)\,\rho_{\max}$. The number of vehicles entering the link is $v_f C \rho_a$. It follows that $\eta^u_{x_1,x_2} = (\min(x_1, l) - \min(x_2, l))\frac{\rho_{\max}}{v_f C \rho_a}$. The expression of $\eta^u_{x_1,x_2}$ in function of the model parameters R, C and l_s and the state variable l is obtained by multiplying the nominator and denominator by l, using equation (5.2) to eliminate ρ_a and equation (5.3). It follows that

$$\eta^u_{x_1,x_2} = \frac{\min(x_1, l) - \min(x_2, l)}{l}\left(\frac{R}{C} + \left(1 - \frac{R}{C}\right)\frac{l}{l_s}\right). \tag{5.4}$$

In (5.4), the first factor scales the proportion of stopping vehicles as a function of the locations x_1 and x_2. The second factor represents the proportion of stopping vehicles if x_1 is upstream of the queue and x_2 is at the intersection, *i.e* the fraction of vehicle stopping on the entire link $\eta^u_{0,L}$. Notice that $\eta^u_{0,L}$ tends to R/C as the queue length l tends to zero and that it increases linearly with the queue length until it reaches one at saturation ($l = l_s$).

The *stopping time* experienced when stopping at x is denoted by $\delta^u(x)$ for the undersaturated regime. Because the arrival of vehicles is homogeneous and the FD is triangular, the delay $\delta^u(x)$ increases linearly with x. At the intersection ($x = 0$), the delay is the duration of the red light R. At the end of the queue ($x = l$) and upstream of the queue ($x \geq l$), the

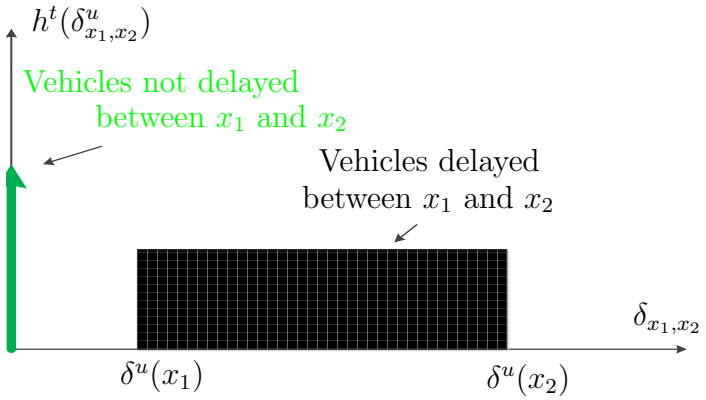


Figure 5.3: Probability distribution of delay between arbitrary locations on an arterial linkin the undersaturated regime.

delay is null:

$$\delta^u(x) = R\left(1 - \frac{\min(x,l)}{l}\right). \tag{5.5}$$

Given that the arrival of vehicles is uniform in time, the distribution of the location where the vehicles reach the queue between x_1 and x_2 is uniform in space. For vehicles reaching the queue between x_1 and x_2, the probability to experience a delay between locations x_1 and x_2 is uniform with support $[\delta^u(x_1), \delta^u(x_2)]$, corresponding to the minimum and maximum delay between x_1 and x_2.

The delay experienced between x_1 and x_2 is a random variable with a mixture distribution with two components: (i) a mass distribution in 0 corresponding to the vehicles that are not delayed between x_1 and x_2 and (ii) a uniform distribution on $[\delta^u(x_1), \delta^u(x_2)]$ corresponding to the vehicles reaching the queue between x_1 and x_2. Let $\mathbf{1}_A(\cdot)$ denote the indicator function of set A, and $\mathrm{Dir}_{\{a\}}(\cdot)$ the Dirac distribution centered in a, used to represent a mass probability. The pdf of delay between x_1 and x_2 (Figure 5.3, left) reads:

$$h^t(\delta_{x_1,x_2}) = (1-\eta^u_{x_1,x_2})\mathrm{Dir}_{\{0\}}(\delta_{x_1,x_2}) + \frac{\eta^u_{x_1,x_2}}{\delta^u(x_2)-\delta^u(x_1)}\mathbf{1}_{[\delta^u(x_1),\delta^u(x_2)]}(\delta_{x_1,x_2}).$$

The cumulative distribution function of delay $H^t(\cdot)$ reads:

$$H^t(\delta_{x_1,x_2}) = \begin{cases} 0 & \text{if} \quad \delta_{x_1,x_2} < 0, \\ (1-\eta^u_{x_1,x_2}) & \text{if} \quad \delta_{x_1,x_2} \in [0, \delta^u(x_1)], \\ (1-\eta^u_{x_1,x_2}) + \eta^u_{x_1,x_2}\frac{\delta_{x_1,x_2}-\delta^u(x_1)}{\delta^u(x_2)-\delta^u(x_1)} & \text{if} \quad \delta_{x_1,x_2} \in (\delta^u(x_1), \delta^u(x_2)], \\ 1 & \text{if} \quad \delta_{x_1,x_2} > \delta^u(x_2). \end{cases}$$

Because of the temporal aggregation of the dynamics, it is not possible to estimate all the parameters which appear in the derivations. Propositions 5.1 and 5.2 analyze the estimation capabilities depending on the available data.

Proposition 5.1. ***Parameter estimation using link delay measurements:*** *The pdf of delay on a link is parameterized by two independent parameters. In particular, the red time R and the fraction of stopping vehicles $\eta^u_{L,0}$ can be chosen to characterize the distribution. The fraction of stopping vehicles is a function of the parameters R, C, l and l_s and is an indicator of the level of congestion ($\eta^u_{L,0} = 1$ at saturation). The cycle length C, the queue length l and the saturation queue length l_s cannot be estimated independently using link stopping times only.*

Proof. The pdf of stopping time on a link is a mixture distribution with two components: a mass probability in zero (weight $1 - \eta^u_{L,0}$) and a uniform distribution on $[0, R]$ (weight $\eta^u_{L,0}$), which is characterized by R and $\eta^u_{L,0}$. The parameters C, l and l_s are related to $\eta^u_{L,0}$ and R by the implicit relation $\eta^u_{L,0} = \frac{R}{C} + \left(1 - \frac{R}{C}\right) \frac{l}{l_s}$ and cannot be uniquely estimated from link delay measurements. □

Proposition 5.2. ***Parameter estimation using delay measurements between locations*** x_1 ***and*** x_2***:*** *The pdf of delay between locations x_1 and x_2 which satisfy $x_2 < x_1 < l$ is parameterized by three independent parameters. In particular, the red time R, the queue length l and the fraction of stopping vehicles on a link $\eta^u_{L,0}$ characterize the distribution. The parameters C and l_s are functions of these parameters.*

Proof. The pdf of delay between x_1 and x_2 is a mixture distribution with two components: a mass probability in zero (weight $1 - \eta^u_{x_1,x_2}$) and a uniform distribution on $[\delta^u(x_1), \delta^u(x_2)]$ (weight $\eta^u_{x_1,x_2} = \frac{x_1 - x_2}{l} \eta^u_{L,0}$). There exists a bijective change of variables in the appropriate sub-spaces of $\mathbb{R}^3$ written as

$$\phi : (\delta^u(x_1), \delta^u(x_2), \eta^u_{[x_1,x_2]}) \mapsto (R, l, \eta^u_{L,0}).$$

It follows that R, l and $\eta^u_{L,0}$ can be chosen as independent parameters to characterize the pdf of delay. □

Probability distribution of delay in the congested regime

This section derives the pdf of delay in the congested regime, when the queue does not fully dissipate before the end of the red time, *i.e* when $l_r > 0$.

As for the undersaturated regime, the delay distribution is computed by deriving the delay experienced between x_1 and x_2 for each arrival time in a cycle. Let n_s be the maximum number of stops experienced by the vehicles in the remaining queue between the locations x_1 and x_2. The delay experienced at location x when reaching the triangular queue at x is readily derived from the expression of the delay in the undersaturated regime, after noticing that for $x \in [0, l_r]$, the stopping time at location x is the duration of the red time R. The expression of the delay at location x is then

$$\delta^c(x) = \begin{cases} R & \text{if } x \leq l_r, \\ R \frac{l_r + l_s - x}{l_s} & \text{if } x \in [l_r, l_r + l_s], \\ 0 & \text{if } x \geq l_r + l_s. \end{cases}$$

The derivations are detailed and illustrated in Appendix A. They encompass the different cases depending on the relative location of x_1 and x_2 with respect to l_r and l. As mentioned earlier, the distance traveled by vehicles in the queue in the duration of a light cycle is l_s.

Concavity properties

Proposition 5.3. ***Mixture of log-concave distributions*** *The pdf of delay between arbitrary locations x_1 and x_2 on an arterial link is a finite mixture of log-concave distributions with at most three components. Each component corresponds to a different* delay pattern *experienced by the vehicles.*

Proof. In the undersaturated regime, the pdf of delay is a mixture of mass probabilities and uniform distributions with at most two components. Each component represents a *delay pattern*: (i) not stopping between x_1 and x_2, mass distribution, or (ii) stopping between x_1 and x_2, uniform distribution.

In the congested regime, the pdf of delay is derived in Appendix A and can be represented as a mixture distribution with at most three components. Each component represents a *delay pattern*. The maximum number of component depends on the location of x_1 and x_2 with respect to the queue and is bounded by three. □

5.3 Probability distribution of travel time

On a path between x_1 and x_2, the travel time y_{x_1,x_2} is the sum of two independent random variables: the delay δ_{x_1,x_2} and the free flow travel time $y_{f;\,x_1,x_2} = p_f(x_1 - x_2)$. Recall that the free flow pace p_f has distribution φ^p with support $\mathcal{D}_{\varphi^p}$. For convenience, the prolongation of φ^p by zero out of $\mathcal{D}_{\varphi^p}$ is still called φ^p (with a slight abuse of notation). Using a linear change of variables, the pdf $\varphi^y_{x_1,x_2}$ of the free flow travel time $y_{f;\,x_1,x_2}$ between x_1 and x_2 is given by:

$$p_f \sim \varphi^p(p_f) \Rightarrow \varphi^y_{x_1,x_2}(y_{f;\,x_1,x_2}) = \varphi^p\left(\frac{y_{f;\,x_1,x_2}}{x_1 - x_2}\right)\frac{1}{x_1 - x_2}.$$

The pdf of travel time is derived from the following fact:

Fact 5.1. ***Sum of independent random variables*** *If X and Y are two independent random variables with respective pdf f_X and f_Y, then the pdf f_Z of the random variable $Z = X + Y$ is given by $f_Z(z) = (f_X * f_Y)(z)$.*

This classical result in probability is derived by computing the conditional pdf of Z given X and then integrating over the values of X according to the *total probability law*. For a congestion state $s \in \{u, c\}$ (undersaturated or congested), the pdf of travel time reads:

$$g^s(y_{x_1,x_2}) = \left(h^s * \varphi^y_{x_1,x_2}\right)(y_{x_1,x_2}).$$

From the property that the convolution product is linear and that the pdf of total or measured delay is a mixture distribution, it follows that the pdf of travel time as a mixture distribution. Each component corresponds to the convolution between a component of the delay distribution (*delay pattern*) and the pdf of free flow travel time. For a link i the pdf of delay between any locations x_1 and x_2 is a mixture with K_i component (K_i may depend on x_1 and x_2 and on the congestion state even though this dependency is not explicitly indicated in the notation). The pdf of delay representing the k^{th} component ($k \in \{1, \ldots, K_i\}$) is denoted $h^{i,k}_{x_1,x_2}$. It represents the pdf of the k^{th} *delay pattern* such as stopping, non-stopping and so on (see Section 5.2 and Appendix A). The pdf of travel time corresponding to delay pattern k is denoted $g^{i,k}_{x_1,x_2}$ and given by $g^{i,k}_{x_1,x_2} = h^{i,k}_{x_1,x_2} * \varphi^y_{x_1,x_2}$. To derive analytic expressions of the pdf of travel time, it suffices to derive analytic expressions of the pdf of travel times associated with the different types of delays. The delay distributions are mixtures of mass probabilities and uniform distributions. The following derives the general expression of the travel time distribution when vehicles experience a delay with mass probability in Δ and when vehicles experience a delay with uniform distribution on $[\delta_{\min}, \delta_{\max}]$.

Travel time distribution

The delay is equal to Δ (mass probability)

This *delay pattern* represents trajectories with $n_s \geq 0$ stops in the remaining queue. It also includes non-stopping vehicles in the undersaturated regime. The corresponding travel time distribution is derived as

$$\begin{aligned} g(y_{x_1,x_2}) &= \left(\text{Dir}_{\{\Delta\}} * \varphi^y_{x_1,x_2}\right)(y_{x_1,x_2}) \\ &= \varphi^y_{x_1,x_2}(y_{x_1,x_2} - \Delta). \end{aligned} \tag{5.6}$$

The delay is uniformly distributed on $[\delta_{\min}, \delta_{\max}]$.

This *delay pattern* represents trajectories with a stop in the triangular queue. The probability of observing a travel time y_{x_1,x_2} is given by

$$\begin{aligned} g(y_{x_1,x_2}) &= \left(\mathbf{1}_{[\delta_{\min},\delta_{\max}]} * \varphi^y_{x_1,x_2}\right)(y_{x_1,x_2}) \\ &= \frac{1}{\delta_{\max} - \delta_{\min}} \int_{-\infty}^{+\infty} \mathbf{1}_{[\delta_{\min},\delta_{\max}]}(y_{x_1,x_2} - z)\ \varphi^y_{x_1,x_2}(z)\, dz. \end{aligned} \tag{5.7}$$

The integrand is not null if and only if $y_{x_1,x_2} - z \in [\delta_{\min}, \delta_{\max}]$ and $z \in \mathcal{D}_\varphi$, *i.e.* if $z \in [y_{x_1,x_2} - \delta_{\max},\ y_{x_1,x_2} - \delta_{\min}] \bigcap \mathcal{D}_\varphi$.

As an illustration, the following derives the pdf of travel time on a partial link (between x_1 and x_2) in the undersaturated regime, for a pace distribution with support on $\mathbb{R}^+$. The delay distribution (Figure 5.4, left) is a mixture of a mass probability at 0 and a uniform distribution on $[\delta^u(x_1), \delta^u(x_2)]$. The pdf of travel time for each delay pattern is computed

from Equations (5.6) and (5.7) and scaled with their respective weights ($1-\eta^u_{x_1,x_2}$ and $\eta^u_{x_1,x_2}$), as illustrated Figure 5.4 (center). From the linearity of the convolution operator, it follows that the pdf of travel times is computed by summing the weighted components (Figure 5.4, right):

$$g^u(y_{x_1,x_2}) = \begin{cases} 0 & \text{if } y_{x_1,x_2} \leq 0, \\ (1-\eta^u_{x_1,x_2})\varphi^y_{x_1,x_2}(y_{x_1,x_2}) & \text{if } y_{x_1,x_2} \in [0, \delta^u(x_1)], \\ (1-\eta^u_{x_1,x_2})\varphi^y_{x_1,x_2}(y_{x_1,x_2}) + \dfrac{\eta^u_{x_1,x_2}}{\delta^u(x_2)-\delta^u(x_1)} \displaystyle\int_0^{y_{L,0}-\delta^u(x_1)} \varphi^y_{x_1,x_2}(z)\,dz & \\ \qquad \text{if } y_{x_1,x_2} \in [\delta^u(x_1), \delta^u(x_2)], & \\ (1-\eta^u_{x_1,x_2})\varphi^y_{x_1,x_2}(y_{x_1,x_2}) + \dfrac{\eta^u_{x_1,x_2}}{\delta^u(x_2)-\delta^u(x_1)} \displaystyle\int_{y_{L,0}-\delta^u(x_2)}^{y_{L,0}-\delta^u(x_1)} \varphi^y_{x_1,x_2}(z)\,dz & \\ \qquad \text{if } y_{x_1,x_2} \geq \delta^u(x_2). & \end{cases} \tag{5.8}$$

The derivations are similar in the congested regime. For example, the pdf of link travel time (see Case 1 in Appendix A for details of the derivations). For link travel times, $x_1 = L$, length of the link and $x_2 = 0$. The distribution is computed via Equation (5.7) and reads

$$g^c(y_{L,0}) = \begin{cases} 0 & \text{if} \quad y_{L,0} \leq \delta_{\min}, \\ \frac{1}{\delta_{\max}-\delta_{\min}} \int_0^{y_{L,0}-\delta_{\min}} \varphi^y_{L,0}(z)dz & \text{if} \quad y_{L,0} \in [\delta_{\min}, \delta_{\max}], \\ \frac{1}{\delta_{\max}-\delta_{\min}} \int_{y_{L,0}-\delta_{\max}}^{y_{L,0}-\delta_{\min}} \varphi^y_{L,0}(z)\,dz & \text{if} \quad y_{L,0} \geq \delta_{\max}, \end{cases} \tag{5.9}$$

$$\text{with} \quad \begin{cases} \delta_{\min} = \delta^c(n_s l_s) + (n_s - 1)R, \\ \delta_{\max} = \delta^c(n_s l_s) + n_s R, \\ n_s = \left\lceil \frac{l_r}{l_s} \right\rceil. \end{cases}$$

Finite mixture of log-concave distributions

Proposition 5.4. ***Mixture of log-concave distributions*** *If the pdf of free flow pace is log-concave, the pdf of travel time between arbitrary locations x_1 and x_2 on an arterial link is a finite mixture of log-concave distributions with at most three components. Each component corresponds to a different* delay pattern.

Proof. Since $y_{x_1,x_2} = \delta_{x_1,x_2} + y_{f;x_1,x_2}$ and since δ_{x_1,x_2} and $y_{f;x_1,x_2}$ are independent r.v., the pdf of travel time is given by the convolution of the pdf of delay and the pdf of free flow travel time. From the linearity of the convolution and Proposition 5.3, it follows that the travel times have a finite mixture distribution with at most three components. Each component is the convolution of the pdf of free flow travel time with a log-concave distribution. Using the

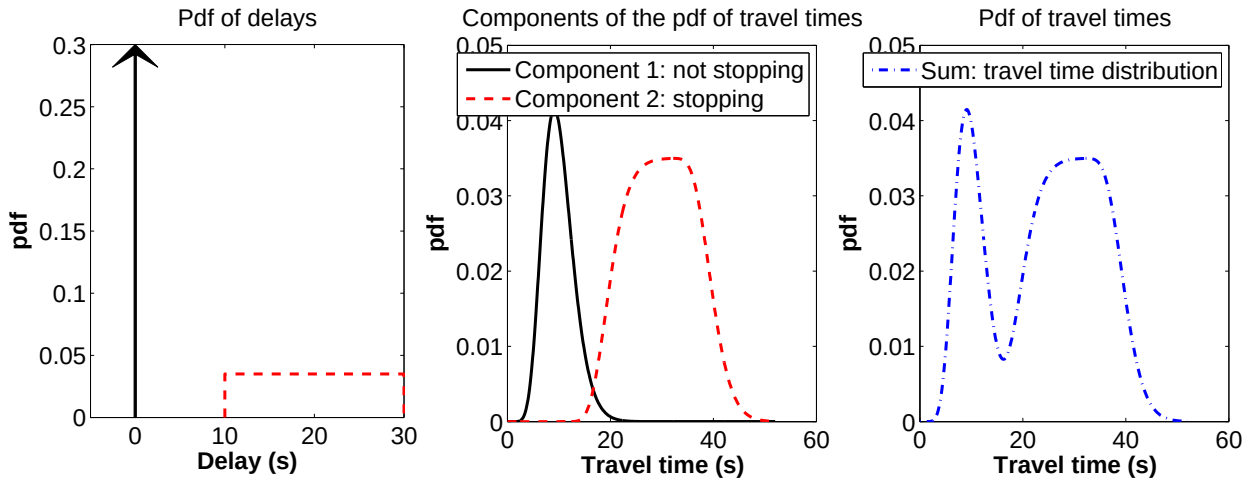


Figure 5.4: Probability distribution of travel time between arbitrary locations on an arterial link in the undersaturated regime. The figure represents the pdf of travel times between x_1 and x_2 in the case where both x_1 and x_2 are in the triangular queue. The figure illustrates the distribution for $\delta^u(x_1) = 10$ s, $\delta^u(x_2) = 30$ s and $\eta^u_{x_1,x_2} = 0.7$. The free flow travel time between x_1 and x_2 has a Gamma distribution with mean 10 s and standard deviation 3 s.

fact that log-concavity is closed under multiplication (concavity is closed under addition), and results on the integration of log-concave functions [187], it follows that the convolution of two log-concave functions is log-concave (Section 3.5 of [29]). If the pdf of free flow pace is log-concave, so is the pdf of free flow travel time and it follows that the pdf of travel time is a mixture of log-concave distributions with at most three components. □

5.4 Learning queue dynamics from sparsely sampled probe vehicles

From traffic flow theory, Section 5.3 derives the pdf of travel time between arbitrary locations on an arterial link parameterized by the network parameters (average red and cycle time, driving behavior, saturation queue length) and the level of congestion (queue length). As probe vehicles report their location periodically in time, the duration between two successive location reports x_1 and x_2 is a measurement of the travel time of the vehicle on its path from x_1 to x_2. This section investigates how to use these travel time observations to learn the parameters of the travel time distributions. Common sampling rates for probe vehicles are around one measurement per minute. Vehicles typically traverse several links between successive location reports. This section does not explicitly model the dependency between link travel times and assumes that they are independent. A *Hidden Markov Model*, as introduced in [188], can be used to represent the dependency between link travel times.

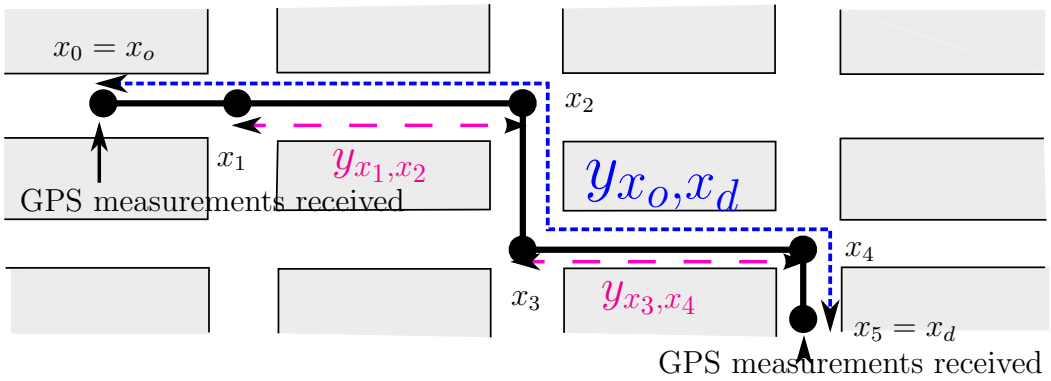


Figure 5.5: Travel time allocation: decomposition of the path travel time into (partial) link travel times. The vehicle sends location measurements in x_o and x_d for a travel time y_{x_o,x_d}. The path extends only on a fraction of the first and last links (partial links). The travel time y_{x_o,x_d} is decomposed into five (partial) link travel times $y_{x_o,x_d} = \sum_{m=0}^{4} y_{x_m,x_{m+1}}$, corresponding to the most likely times spent on each (partial) link given the parameters of the network and the state of traffic.

In the present model, the transition probabilities of the *Markov Model* would represent the probability to stop on a link given that the vehicle stopped (or did not stop) on the upstream link. The pdf of travel time on the path of the probe vehicle can be computed from the pdf of (partial) link travel time using convolution. However, unless the (partial) link travel times are normally distributed, the computation of these convolutions is difficult. To overcome this difficulty, the section proposes an iterative algorithm inspired from the *Expectation Maximization* (EM) algorithm [62]. The mathematical properties and derivations of the EM algorithm are reviewed in Chapter 6. The algorithm is proven to converge to a local optimum of the likelihood function and performs very well in practice. This section uses a variation of this algorithm, referred to as *hard EM* because the E step corresponds to a *hard* assignment:

1. *Hard E step*: for each probe vehicle travel time measurement between successive location reports, compute the most likely travel times of individual (partial) links traversed by the probe vehicle (travel time allocation or decomposition [101, 107]).

2. *M step*: using the travel times allocated to each (partial) link of the network during the *Hard E step*, estimate the parameters of the pdf of travel time by maximizing likelihood of the allocated travel times.

Travel time decomposition

For a vehicle traveling from an origin x_o to a destination x_d through $M \geq 0$ intersections, the travel time y_{x_o,x_d} is the sum of the travel times on each of the (partial) links (Figure 5.5):

$$y_{x_o,x_d} = \sum_{m=0}^{M} y_{x_m,x_{m+1}}. \tag{5.10}$$

For $m \in \{1 \dots M\}$, the point x_m represents the most downstream location on the m^{th} link on the path, $x_0 = x_o$ and $x_{M+1} = x_d$. Let i_m be the m^{th} link of the path, let g^{i_m} be the pdf of travel time on (partial) link i_m, traveled between x_m and x_{m+1}. Note that the index x_m, x_{m+1} is dropped for notation simplicity. Let $g^{i_m,k}$ denote the pdf of travel time corresponding to delay pattern k. The delay patterns implicitly depend on the level of congestion (*underaturated* or *congested* regime) even though this dependency is not denoted explicitly. The optimal decomposition of a path travel time y_{x_o,x_d} into (partial) travel times $(y_{x_m,x_{m+1}})_{m=0\dots M}$ maximizes the probability to receive travel time observations $(y_{x_m,x_{m+1}})_{m=0\dots M}$ under the constraint that they sum to y_{x_o,x_d}. The optimization problem reads:

$$\begin{aligned} \underset{(y_{x_m,x_{m+1}})_{m=0\dots M}}{\textbf{minimize}} \quad & : \quad \sum_{m=0}^{M} -\ln(g^{i_m}(y_{x_m,x_{m+1}})) \\ \textbf{s.t.} \; & : \quad y_{x_o,x_d} = \sum_{m=0}^{M} y_{x_m,x_{m+1}} \text{ and } \forall m \; y_{x_m,x_{m+1}} \geq 0. \end{aligned} \tag{5.11}$$

As formulated, (5.11) is not convex. Common optimization algorithms (*e.g* based on gradient or Hessian) are only guaranteed to find local optima. Global optimization algorithms [116, 225] can solve the problem of local optima but are out of the scope of this thesis. The remainder of this section investigates different algorithms to solve the problem. They leverage the convexity property of the travel time distribution functions to find convex formulations of (5.11). The performance of the different algorithms are analyzed in Section 5.5.

1. ***Gradient*** *algorithm:* To limit the risk of getting stuck in local optima, the gradient descent algorithm is run with several random starts. As the feasible set of Problem (5.11) is bounded and low-dimensional, the algorithm is expected to perform decently. However it does not exploit the convexity property of the probability distribution functions (Proposition 5.4). The following algorithms leverage this property.

2. ***Expectation-Maximization*** *(EM) algorithm:* After an initial allocation $(y^0_{x_m,x_{m+1}})$ of the travel times to the links of the path (e.g. random allocation, allocation proportional to the mean or the free flow travel times), the algorithm iterates between an analytical computation (E step) and a small scale optimization problem (M step). It is only guaranteed to converge to local optima but exploits the convexity properties of the probability distribution functions:

- *E step:* at iteration n, the travel time allocated to link i_m is $y^n_{x_m,x_{m+1}}$. Compute the probability $\tilde{\beta}^n_{i_m,k}$ that the vehicle experienced delay pattern k on link i_m:

$$\tilde{\beta}^n_{i_m,k} = \frac{\upsilon_k g_{i_m,k}\left(y^n_{x_m,x_{m+1}}\right)}{\sum_{k'=0}^{K_{i_m}} \upsilon_{k'} g_{i_m,k'}\left(y^n_{x_m,x_{m+1}}\right)}. \tag{5.12}$$

- *M step:* solve the convex optimization program (5.13) and go to E Step until convergence.

$$\underset{(y^{n+1}_{x_m,x_{m+1}})_{m=0}^M}{\textbf{minimize}} \sum_{m=0}^{M}\sum_{k=0}^{K_{i_m}} -\tilde{\beta}_{i_m,k}\ln(g^{i_m,k}(y^{n+1}_{x_m,x_{m+1}})),\ \textbf{s.t.}\ (5.10). \tag{5.13}$$

3. ***Given stop*** *algorithm:* Given the model of Section 5.3, a vehicle has one *delay pattern* on each link of its path. For $k \in K_{i_m}$, let $\beta_{i_m,k} \in \{0,1\}$ equal 1 if the vehicle has delay pattern k on link i_m and 0 otherwise. If the sampling strategy detects the stops of the vehicle, the variables $\beta_{i_m,k}$ are known and the travel time allocation solves the following convex optimization problem:

$$\begin{aligned} \underset{\left(y_{x_m,x_{m+1}}\right)_m}{\textbf{minimize}}: &\quad \sum_{m=0}^{M}\sum_{k=0}^{K_{i_m}} -\beta_{i_m,k}\ln(g^{i_m,k}(y_{x_m,x_{m+1}})) \\ \textbf{s.t.}: &\quad y_{x_o,x_d} = \sum_{m=0}^{M} y_{x_m,x_{m+1}} \text{ and } \forall m\ y_{x_m,x_{m+1}} \geq 0. \end{aligned} \tag{5.14}$$

4. ***Enumeration*** *algorithm:* Sampling strategies rarely provide the value of the binary variables $\beta_{i_m,k}$ which must be considered as decision variables in (5.14), with the constraints

$$\forall m \sum_{k=1}^{K_{i_m}} \beta_{i_m,k} = 1,\ \ \forall\,(m,k)\,\beta_{i_m,k} \in \{0,1\}. \tag{5.15}$$

The constraints ensure that a vehicle has exactly one delay pattern on each link. When the sampling strategy does not provide the type of delay experienced by the vehicle, the decomposition problem is given by:

$$\begin{aligned} \underset{\substack{\left(y_{x_m,x_{m+1}}\right)_m \\ (\beta_{i_m,k})_{m,k}}}{\textbf{minimize}}: &\quad \sum_{m=0}^{M}\sum_{k=0}^{K_{i_m}} -\beta_{i_m,k}\ln(g^{i_m,k}(y_{x_m,x_{m+1}})) \\ \textbf{s.t.}: &\quad y_{x_o,x_d} = \sum_{m=0}^{M} y_{x_m,x_{m+1}} \text{ and } \forall m\ y_{x_m,x_{m+1}} \geq 0, \\ &\quad \beta_{i_m,k} \in \{0,1\},\ \ \sum_{k=1}^{K_{i_m}} \beta_{i_m,k} = 1. \end{aligned} \tag{5.16}$$

Problem (5.16) can be solved by enumerating the $\prod_{m=0}^{M} K_{i_m}$ convex optimization programs corresponding to the different sets of feasible $(\beta_{i_m,k})_{i_m,k}$. The complexity is exponential in the number of links traversed by the vehicle. The upper bounds on the number of links in a path (probe vehicles typically send their location every minute and their speed is bounded) and the number of mixture components ($K_{i_m} \leq 5$) maintain the *tractability* of this algorithm.

5. ***Hard EM*** *algorithm:* Problem (5.16) can also be solved using a *hard EM algorithm*. The Hard EM algorithm forces the vehicle to have exactly one delay pattern on each link of the path, instead of using the probability of each delay pattern (5.12). Given a travel time allocation at iteration n, the hard E step computes $\beta^n_{i_m,k}$ such that it is equal to 1 if delay pattern k is the most likely on link i_m and to 0 otherwise:

$$\beta^n_{i_m,k} = \begin{cases} 1 & \text{if } k = \arg\max\limits_{k' \in K_{i_m}} \tilde{\beta}^n_{i_m,k'}, \\ 0 & \text{otherwise} \end{cases} \tag{5.17}$$

Recall that $\tilde{\beta}^n_{i_m,k'}$ is computed according to (5.12). The M step solves (5.13) with $\tilde{\beta}_{i_m,k} = \beta^n_{i_m,k}$. Similar to the EM algorithm, the hard EM algorithm exploits the underlying structure of the optimization problem but only guarantees convergence to local optima and random starts are helpful to increase the chances of convergence to the global optimum. Compared to the *EM* algorithm, both the *enumeration* and the *Hard EM* algorithm leverage additional information regarding the physics of the problem: each vehicle has exactly one *delay pattern* on each traversed link.

Estimation of traffic conditions

The travel time distribution between any location x_1 and x_2 on a link i is characterized by the network parameters (C^i, R^i), θ^i_p and l^i_s) and by the state variable (queue length, l^i). The (partial) link travel times allocated to link i (denoted $(y^j_{x_1,x_2})_{j=1:J^i}$) enable the estimation of a subset of these parameters (Proposition 5.2) by maximizing the likelihood (or more conveniently the log-likelihood) of the allocated (partial) travel times with respect to the network and state parameters:

$$\begin{array}{ll} \underset{R^i,\eta^i_{0,L},l^i_s,\theta^i_p,l^i}{\textbf{minimize}} & \sum_{j=1}^{J^i} -\ln(g^i(y^j_{x_1,x_2})) \\ \textbf{s.t.} & R^i \leq C^i. \end{array} \tag{5.18}$$

The locations x_1 and x_2 differ for each measurement, even though the dependency of x_1 and x_2 on j is not denoted explicitly. Additional constraints and bounds may be added to limit the feasible set to physically acceptable values of the parameters. Note that problem (5.18) is not convex. However, the search space is limited (low dimensional optimization problem with bounds on each of the variables). A grid search with local descent algorithm for the B best sets of parameters of the grid search performs well for this type of problem.

Remark 5.1 (A priori information)**.** *Additional constraints on the optimization problem may improve the results by adding a-priori information about the physics and the dynamics. For example, it is possible to impose similar free flow parameters (parameters sharing) in different parts of the city or on links with similar features. These constraints couple the optimization problems, resulting on fewer but potentially larger optimization problems (depending on the constraints and parameter sharing among the links). From historical data or prior information, it is also possible to input* a-priori *information on the free flow pace or on traffic signals by fixing the value of the corresponding parameters. These parameters no longer appear in the list of variables of the optimization problem* (5.18)*.*

5.5 Numerical experiment and results

This section validates the derivation of the statistical distribution of travel times as well as the learning algorithm from sparsely sampled probe vehicles. First, the section analyzes numerically the capacity of the travel time distributions, derived from the physics of traffic flows, to represent the empirical distribution of travel times more accurately than classical distributions such as normal, log-normal or Gamma distributions. Second, the section studies the performance and the accuracy of the different travel time decomposition algorithms presented in Section 5.4 and presents trade-off between them in terms of computation time and accuracy.

Validation of the travel time distributions

The model presented in this chapter relies on assumptions on the dynamics of traffic flows on each link of the network to derive probability distributions of travel times. The capacity of the derived distributions to fit experimental data is compared to those of "classic" classes of distributions: the *normal* distribution, the *log-normal* distribution and the *Gamma* distribution. For each class of distributions, the capacity to fit empirical data is computed using a statistical test. The test hypothesis is that link travel times are distributed according to the chosen distribution (the complementary hypothesis is that the travel times are *not* distributed according to this distribution).

The experimental data was collected during a *field experiment* from the 29th of June to the 1st of July 2010 as part of the *Mobile Millennium* project [4]. Twenty drivers, each carrying a GPS device, drove for 3 hours (3:15pm to 6:15pm) around two distinct loops in San Francisco illustrated Figure 5.6. The first loop was 1.89 miles long and the second one 2.31 miles long. The GPS devices recorded the location of the vehicles every second and provided detailed information on the trajectories of the drivers. For the sake of validating the modeling assumptions and the accuracy of the learning algorithm, the detailed data was used to provide the link travel times experienced by the probe vehicles.

Figure 5.6: Routes of the network used for field test validation. The drivers drove around two distinct loops consisting in Van Ness Ave. South bound and Franklin St. north Bound for the first routes and Van Ness Ave. North bound and Gough St. South bound for the second route. Signalized intersections are indicated with a circle.

Validation framework

The link travel times extracted from the GPS trajectories of the probe vehicles are separated into two complementary data sets: a training set and a validation set. For each link of the network and each class of distribution (traffic distribution, Normal, Log-Normal, Gamma), the maximum likelihood estimates of the distribution parameters are computed using measurements from the training set. In the numerical experiments, the amount of measurements from the training set which is used to learn the distributions varies. This analyses how the amount of data used for the training influences the accuracy of the learning. It aims at recommending required amount of data to have confidence in the quality of the results and at comparing the different models in their data requirements.

For each link and each class of distribution, the hypothesis H_0 is as follows: *the link travel times are distributed according to the chosen distribution.* The hypothesis is tested on the

Table 5.1: Outcome of statistical tests.

		True hypothesis	
		H_0 is true	H_0 is false
Decision	Accepts H_0	Right decision	Wrong decision, Type II error, rate β
	Rejects H_0	Wrong decision, Type I error, rate α	Right decision

validation data set using the *Kolmogorov-Smirnov* test [162], also referred to as K-S test. The K-S test is a standard non-parametric test to state whether samples are distributed according to an hypothetical distribution (in opposition to other tests like the T-test that tests uniquely the mean, or the chi-squared test that assumes that the data is normally distributed). The test is based on the K-S statistics which is computed as the maximum difference between the empirical and the hypothetical cumulative distributions. The test provides a p-value which informs us on the goodness of the fit. Low p-values indicate that the data does not follow the hypothetical distribution. For each hypothetical distribution, Figure 5.7 (left) shows the average p-value of the links of the network as the percentage of training data increases. The hypothesis H_0 is rejected for p-values inferior to the significance level α. The significance level α corresponds to the percentage of Type-I error allowed by the test (rejecting the null hypothesis when it is actually true). Table 5.1 illustrates the outcome of statistical tests and the different types of error. Figure 5.7 (right) shows the evolution of the percentage of links that passes the K-S test at significance level $\alpha = 0.1$. Both figures show that the traffic distribution represents a better fit of the travel time distributions than any of the other distributions tested in this article. The relative superiority of the traffic model is more significant when little data is available. This may be a sign of the robustness of the model when little data is available (because of the intrinsic structure of the distributions representing the physical model). This is precisely the goal of the algorithm (and model), which was specifically created to handle low volumes of probe data. As for the "classic" distribution, the Log-Normal model performs better than both the Normal and the Gamma distribution.

As the Log-Normal distribution out-performs both the Normal and the Gamma distribution, the remainder analysis focuses on comparing the traffic and the Log-Normal distribution. In particular, comparing both distributions with the empirical distribution of travel times provides important insight in terms of the specific characteristics of the distribution which the traffic model is able to capture. Figure 5.8 shows the distributions learned by the traffic model, the Log-Normal fit as well as the distribution of the empirical data collected in the field test experiment. The figure represents both the pdf and the cdf of the traffic (solid blue line) and log-normal (dashed red line) distributions. The histograms represent interval counts of the probe travel times, normalized so that the area of the histogram sums to one. The black line with circles represents the *empirical cumulative distribution* (Kaplan-Meier estimate [132]) of the travel times collected by the probes.

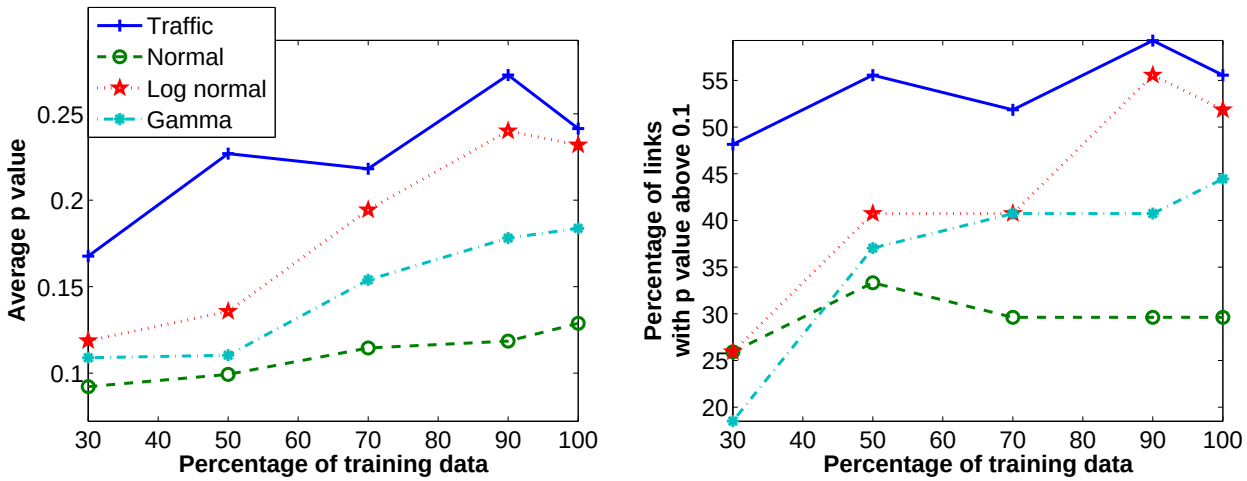


Figure 5.7: Goodness of fit of the model depending on the percentage of training data used to learn the parameters. (**Left**) Average p-value of the links of the network for the different hypothetical distributions (traffic, normal, log-normal and Gamma). (**Right**) Percentage of links that pass the KS test with a significance level $\alpha = 0.1$.

The traffic distribution captures the specific characteristics of traffic dynamics. There is a distinctive peak in the distribution representing the vehicles that do not stop on the link and travel at their free flow speed. For higher travel times, the distribution is approximately uniform, representing the vehicles that are delayed on the link, between a minimum delay (0, for the last vehicle stopping in the queue) and a maximum delay (the duration of the red time, for the first vehicle stopping at the intersection). As for the Log-Normal distribution, it cannot capture these specifics of the travel time distribution and the parameters are harder to interpret.

On link 1, both distributions capture the long tail of the distribution but only the traffic distribution is able to represent the peak in the pdf due to the non stopping vehicles and to estimate accurately the maximum delay. On link 2, there are very few travel times between 35 and 50 seconds, likely due to important synchronization with the upstream link. None of the traffic or Log-Normal distribution is able to capture this. However, the traffic distribution models accurately the peak due to the non stopping vehicles and estimate the maximum delay.

Due to light synchronization, some links have arrivals with platoons, and thus do not follow the hypothesis of constant arrivals. On these links, delays are not uniformly distributed among the stopping vehicles and the derivations of the queuing model have to be adapted [5, 15]. Basically, the delay function $\delta^r(x)$ ($r \in \{u, c\}$ representing the undersaturated and congested regimes) is piecewise linear and the derivations of the statistical distributions must be updated accordingly, adding parameters to the model. Figure 5.8 (right) represents the empirical and hypothetical distribution of travel times for a link with platoon arrivals. There are very few vehicles with a travel time between 30 and 50 seconds, representing a time

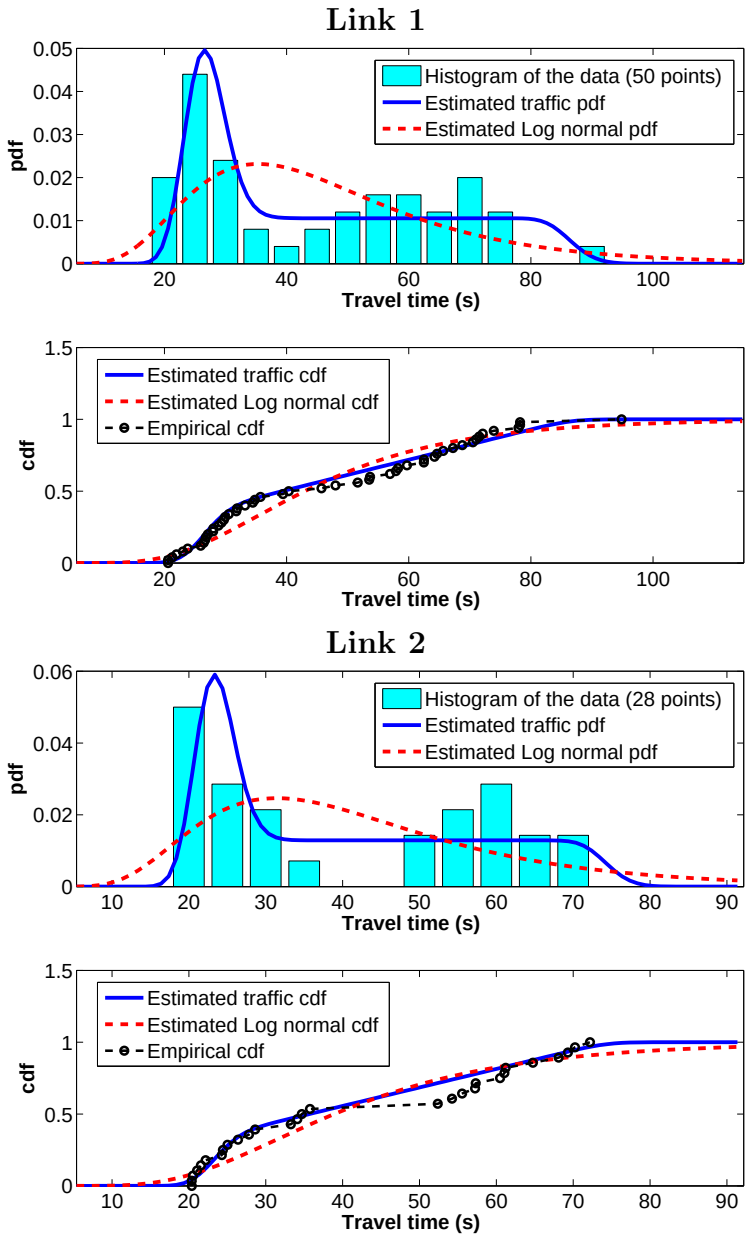


Figure 5.8: Comparison of the traffic and the log-normal distributions with the empirical distribution of travel times on two links of the network.

interval during which there is very few arrivals on the links, likely when the upstream signal is red. The log-normal distribution does not capture this characteristics of the distribution either. Moreover, the traffic model provides an estimation of the red time, the free flow speed and the fraction of stopping vehicles (representing congestion) which is important information for traffic management and operations.

Travel time decomposition algorithm

After validating the model and analyzing its limitation, this section assesses the computational complexity and the accuracy of the travel time decomposition algorithms presented in Section 5.4. This validation is done using *Next Generation Simulation* (NGSIM [174]) traffic data on the Peachtree Street network (Atlanta, Georgia). The network consists in twelve 3 lane-links with five intersections. This dataset offers very detailed trajectories of *all* the vehicles traveling on the network and thus an adequate ground truth dataset to validate against.

Experiment setup

Automatic processing of video camera data provides detailed trajectories (location every 0.1 seconds) of all the vehicles traveling on the network between 4:00 and 4:15pm on November 8, 2006 (more than 700 trajectories). The traffic conditions are undersaturated, close to saturation. The numerical results simulate probe vehicles reporting their location with different sampling frequencies and compute the time spent on each link and the locations of stops between successive measurements to serve as ground truth for the travel time allocation. For each probe measurement, the travel time is allocated to the corresponding links according to the optimization algorithms described in Section 5.4. The performances of the different travel time allocation algorithms are compared to an algorithm which allocates the travel times proportionally to the free flow travel time on each link (***Benchmark*** *algorithm*). Denoting by v_m^f the free flow speed on link m and by $|x_{m+1} - x_m|$ the distance traveled on link m, the travel time allocated to each link m of the path by the *benchmark* algorithm is given by

$$y_{x_m,x_{m+1}} = \frac{1}{Z}\frac{|x_{m+1} - x_m|}{v_m^f}, \tag{5.19}$$

where the proportionality constant Z is chosen such that the allocated travel times sum up to the path travel time as stated in (5.10). Recall the different algorithms presented in Section 5.4

1. The *Gradient* algorithm finds local optima of (5.11) using a gradient descent algorithm
2. The *EM* algorithm is an iterative algorithm. At iteration n, it computes $(\tilde{\beta}^n_{i_m,k})$ for each pair of link m and delay pattern k according to (5.12) and solves the convex optimization problem (5.13).

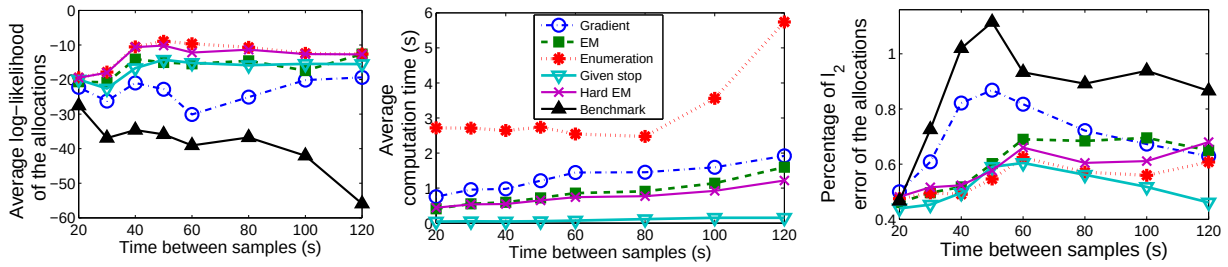


Figure 5.9: Performance analysis of the different travel time allocation algorithms as a function of the sampling frequency. (**Left**) Average log-likelihood of the travel time allocations. (**Center**) Average computation time. (**Right**) Average percentage error .

3. The *Given stop* algorithm solves the convex optimization problem (5.13) where the values of $\tilde{\beta}_{i_m,k}$ are equal to $\beta_{i_m,k} \in \{0,1\}$ and are given by the sampling scheme which detects the location of stops.

4. The *Enumeration* algorithm solves a serie of convex optimization problems (5.13) for each set of $(\beta_{i_m,k})_{m=0}^{M}$ satisfying (5.15).

5. The *Hard EM* algorithm is an iterative algorithm. At iteration n, it computes $\beta_{i_m,k}^{n} \in \{0,1\}$ according to (5.17) and solves the convex optimization problem (5.13), with $\tilde{\beta}_{i_m,k} = \beta_{i_m,k}^{n}$.

6. The *Benchmark* algorithm allocates travel times proportionally to the free flow speed on each link (Equation (5.19)).

Convergence and performance analysis

When vehicles remain on the same link between two successive location measurements, the travel time allocation is trivial and is not taken into account in these results. For each probe measurement, the algorithm computes the log-likelihood (objective function of (5.11)) of the allocations performed by the different algorithms and also reports the average computation time (Figure 5.9, left and center). The algorithms *Given stops*, *Enumeration* and *Hard EM* assume that each vehicle has a specific delay pattern on each link whereas *EM* allows for a mixture of delay patterns and *gradient* does not make any assumption. All the algorithms provide an allocation which is more likely than the *benchmark*. The algorithms which leverage the convexity property of the travel time distributions lead to better convergence results (*gradient* has the lowest likelihood of all optimizations). The algorithms *enumeration* and *hard EM* provide similar convergence results but the computation time is much better for *hard EM*. In particular, when the sampling time increases, the effect of the exponential computation time is substantial for the *enumeration* algorithm. The algorithm *given stop* provides allocations with an average log-likelihood slightly inferior to the *enumeration* and *hard EM* algorithms. Indeed, vehicles may not always experience the most likely delay

patterns. However, the small difference in log-likelihoods (in comparison to the benchmark algorithm for example) may indicate that the assumption that vehicles experience the most likely delay patterns is important to guarantee the quality of the results.

Validation of the algorithms

Let $\hat{y}_q^l$ be the q^{th} travel time allocated on link l and by y_q^l the actual travel time of the probe vehicle (computed from the detailed trajectories). The number of travel times allocated on link l is denoted Q_l. The average percentage error on link l is defined as the root mean square error of the allocation divided by the average travel time TT_l:

$$e_l = \frac{1}{TT_l}\sqrt{\frac{\sum_{q=1}^{Q_l}(\hat{y}_q^l - y_q^l)^2}{Q_l}}.$$

To have a more compact validation metric on the entire network, the network average percentage error is the average of the percentage error on the different links (Figure 5.9, right).

Confirming the expectations, *given stop* provides the best results. The information on stops is rarely available in current sampling strategies and the algorithms *enumeration* and *hard EM* provide the highest accuracy, with slightly better results but higher computation cost for *enumeration*. The *gradient* algorithm has the least accuracy of the optimization algorithms which underlines the importance of the structure imposed by the other algorithms. As a tradeoff between accuracy and computation time, the hard EM algorithm seems the best suited to solve the optimization problem. It provides an improvement of 35% to 50% compared to the benchmark method for common sampling rates (30 seconds or more between measurements).

5.6 Conclusion and discussion

This chapter derived a parametric probability distribution of travel times between arbitrary locations on an arterial link from horizontal queuing theory. The model captures the shape of the distribution, which characterizes the periodic formation and dissolution of queues. The distributions are parameterized by physical parameters (red time, cycle time, parameters of the free flow pace, queue length and queue length at saturation) which can be estimated using travel time measurements. The parameters may not all be estimated independently (Propositions 5.1 and 5.2) but it is always possible to retrieve the duration of the red time, the level of congestion and the parameters of the free flow pace distribution. The queue length can also be estimated from probe vehicles reporting partial link travel times. The modeling is designed to incorporate sparsely sampled probe data in the estimation.

- The pdf of travel time is derived between any two locations on a link to take into account the fact that location measurements from probe vehicles do not coincide with the beginning and end of the links. Travel speeds vary significantly within an arterial

link (stops are more likely close to intersections) and scaling of partial travel times may result in important errors. Moreover, a finer discretization of the road network would imply the learning of a larger number of parameters which increases the risk of over-fitting given the amount of data available today at a large scale. It also increases the potential errors introduced by the travel time allocation as there would be more segments traversed.

- The probability distributions are mixtures of log-concave distributions. The proof of this property is used to formulate the travel time allocation problem as a convex optimization problem and to incorporate travel time measurements when several links are traversed between successive location measurements.

The numerical results show the superiority of the *physical* model, derived from horizontal queuing theory. The model represents more accurately the distribution of travel times when compared to commonly used distributions (normal, log-normal, Gamma, GMM). The traffic distribution performs particularly well (in comparison with the other distributions) when little data is available. It captures the delay of vehicles due to the presence of a queue that forms and dissipates periodically because of the traffic signal. The learning and estimation rely on small optimization problems which can be run in parallel in large urban networks. In addition to the estimation capabilities, the model estimates key parameters such as the queue length or the red time which are essential information for planning purposes.

The numerical analysis shows that the assumption of uniform arrivals is the most restrictive assumption on which this work is based. The assumption does not take into account signal synchronization. It is possible to generalize the proposed approach to take into account platoon arrivals. Note that this generalization does not invalidate the methodology presented in this article. In particular, the probability distribution of stopping times will remain a mixture of discrete mass probabilities and uniform distributions.

The probability distribution of travel times are finite mixture distributions [107]. Each component of the mixture corresponds to a type of delay: stopping or not stopping for the undersaturated regime or depending on the location of the vehicle in the congested regime. The estimation of transition probabilities representing the probability of a type of delay on a link given the type of delay on the upstream link would allow to compute route travel time distributions with a Markov chain approach.

Chapter 6

Statistical dynamics of physical queuing networks

Chapter 5 presents a statistical model of urban traffic based on well-established theory of traffic flow through signalized intersections. It captures the variability of travel times among vehicles traveling on the network (vehicle to vehicle variability) because of the presence of traffic signals. The machine learning framework enables us to learn the parameters of the traffic dynamics (such as free flow velocity or traffic signal parameters) and to account for discrepancies between the model and the physical reality as well as noise in the data. Another advantage of the machine learning component of the approach is to leverage significant amounts of historical data collected to improve the estimation of the model parameters and provide better real-time estimates.

It is possible to go further in the modeling presented in Chapter 5 by taking into account the network dynamics, rather than considering each link of the network independently. In particular, congestion spreads on a network and information on a link of the network can be used to infer traffic conditions on links which receive little data. Such a hybrid model of traffic flow theory and statistical modeling provides a distinct advantage over pure statistical or pure traffic theory models in that it is robust to noisy data (due to the large volumes of historical data) and it produces forecasts using traffic flow theory principles consistent with the physics of traffic.

The chapter is organized as follows. Section 6.1 summarizes the notation used in the chapter. Section 6.2 presents the traffic model and the underlying assumptions. Section 6.3 summarizes how probability distributions of travel time between any two locations are derived from this model (results from Chapter 5 with minor notation changes). The section also models the spatio-temporal statistical dependencies between the links of the network. Section 6.4 describes the algorithm developed to learn the parameters of the network and then infer and predict traffic conditions and distributions of travel time across the network (EM Algorithm using particle filtering). Section 6.5 analyzes the estimation capabilities of the model.

6.1 Summary of the notations used in the chapter

1. **Traffic model parameters**
   The traffic model parameters represent the characteristics of the network. They are specific to a link i of the network. For notational simplicity, the subscript i is omitted when the derivations are valid for any link of the network.

$\rho^i_{\max}$	Maximum density of link i.
$q^i_{\max}$	Capacity (maximum flow) on link i.
ρ^i_c	Critical density of link i.
w^i	Backward shockwave speed of link i.
$\xi^i_{\max}$	Maximum number of vehicles that can physically be on link i. This is the number of vehicles when the density is the maximum density. For a link of length L^i, $\xi^i_{\max} = \rho^i_{\max} L^i$.
v^i_f	Free flow speed of link i.
p^i_f	Free flow pace (inverse of free flow speed) of link i. The free flow pace and free flow speeds are related as $p^i_f = 1/v^i_f$.

2. **Traffic signal parameters**
   The traffic signal parameters characterize the properties of the traffic signal that condition the traffic dynamics. This model only considers traffic signals in the form of traffic lights. As for the traffic model parameters, these variables are specific to a link i of the network. However, the subscript may be omitted.

C^i	Duration of a light cycle on link i.
R^i	Duration of the red time on link i.
ξ^i_s	Maximum number of vehicles that can exit link i during a light cycle. This variable is related to the ratio of green time and the traffic model parameters.

3. **Traffic state variables**
   The traffic state variables describe the conditions of traffic that characterize the traffic dynamics on the network. The variables are specific to a link i and a time interval t and represent the dynamic evolution of the traffic state in the different time intervals $t \in \{0 \dots T\}$. The reference to the link or to the time interval may be omitted when the derivations are not link or time specific.

$\rho^{i,t}_a$	Arrival density on link i during time interval t.
$v^{i,t}_a$	Arrival shockwave speed on link i during time interval t (speed of growth of the queue due to additional vehicles arrival).
$\tau^{i,t}$	Duration of the clearing time on link i during time interval t.
$l^{i,t}_{\max}$	Length of the triangular queue on link i during time interval t.
$\xi^{i,t}$	Number of vehicles that stop during each light cycle.

4. **Network variables and parameters**
The network variables and parameters characterize the architecture of the road network and describe the flow of vehicles at intersections.

$\mathcal{I}$	Set of the links of the network.
$\mathcal{K}$	Set of the intersections of the network.
L^i	Length of link i.
i, j	Indices of links of the network ($i, j \in \mathcal{I}$). When referring to an intersection, i refer to a link upstream of the intersection whereas j refers to a link downstream of the intersection.
k	Index of an intersection of the network.
L^k_{in}	Set of incoming links of intersection k.
L^k_{out}	Set of outgoing links of intersection k.
k_{in}	Source (if existing) of intersection k.
k_{out}	Sink (if existing) of intersection k.
$n^{i,t}_{\text{in}}$	Number of vehicles that arrive in link i during a light cycle for time interval t.
$n^{i,t}_{\text{out}}$	Number of vehicles that leave link i during a light cycle for time interval t.
$N^{i,t}_{\text{in}}$	Cumulative number of vehicles that arrive in link i during time interval t.
$N^{i,t}_{\text{out}}$	Cumulative number of vehicles that leave link i during time interval t.
$N^{i,j,t}_{\text{in}}$	Cumulative number of vehicles that leave link i and are assigned to link j during time interval t.
κ^i	Number of lanes of link i.

5. **Particle filter and E Step**
The inference of traffic states on the network given the parameters of the network, of the turn movements and given observed path travel time data is computed using an approximation (for tractability reasons). This approximation relies on particle filtering.

V	Number of particles.
v	Index of the particle.
$\xi^{i,t}_v$	State of particle v on link i during time interval t.
ω_v	Importance weight of particle v.
$a^{i,t}(\xi^{i,t})$	Expected probability that link i is in state $\xi^{i,t}$ at time interval t, computed from the approximation of the joint distribution given by the particles and their importance weight.
$b^{i,j,t}(\xi^{i,t}, N^{i,j,t}_{\text{in}} : j \in L^k_{\text{out}})$	Expected probability that link i is in state $\xi^{i,t}$ at time interval t and that $N^{i,j,t}_{\text{in}}$ vehicles get assigned to the outgoing links of the intersection. It is computed from the approximation of the joint distribution given by the particles and their importance weight.

6. **Probabilities**
The model relies on a probabilistic description of the traffic network dynamics, whose

notations are summarized in the following Table.

φ^i	Probability distribution function of the free flow pace on link i. This function is defined on $\mathbb{R}^+$ and for $p_f \in \mathbb{R}^+$, $\varphi^i(p_f)$ is the probability density that vehicles drive with a free flow pace p_f.
θ_p^i	Parameters of the probability distribution function φ^i.
$\gamma(\cdot)$	Probability distribution function of a random variable with Gamma distribution.
(α^i, β^i)	In the case of a gamma distribution on the free flow pace, the parameters of the distribution are the shape α^i and inverse scale parameter β^i. The Gamma distribution reads $\gamma(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$, where Γ is the Gamma function defined on $\mathbb{R}^+$ and with integral expression $\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t}\, dt$, when i is omitted for simplicity.
y_{x_1,x_2}	Observation of the random variable representing the travel time between locations x_1 and x_2.
$\mathbf{y}^{i,t}$	Set of travel time observations received on link i during time interval t.
$I^{i,t}$	Number of travel time observations received on link i during time interval t.
$g^{i,t}(\cdot)$	Probability distribution function of travel times on link i during time interval t. The function is parameterized by the traffic model and signalization parameters. It changes over time with the state of the link. The function also takes into account the location of the measurements x_1 and x_2 on link i such that $g^{i,t}(y_{x_1,x_2})$ is the probability density of the travel time observation y_{x_1,x_2}.
$\nu^{i,j}$	Probability that a vehicle leaving link i is assigned to link j.
λ^j	Intensity of the Poisson process of vehicles arrival on an outgoing link $j \in L^k_{\text{out}}$ of intersection k, coming from a source k_{out}.
$\pi^i(\xi)$	Probability that link i is in state ξ at the beginning of the experiment. These probabilities represent probabilistic initial conditions for the state of link i.

7. **Other variables**

t	Index of the time interval.
T	Index of the last time interval. By convention, the first interval is numbered 0 so $T+1$ is the number of time intervals.
Δ_t	Duration of a time interval.
$\mathbf{1}_S$	Indicator function of set S.

8. **Probability distributions**

$\mathcal{P}\left(\Xi \vert \mathbf{y}^{i,t}, R^i, C^i, \xi_s^i, \theta_p^i : i \in \mathcal{I},\, t \in \{0 \dots T\}\right)$	Probability of observing a state evolution Ξ given the travel time observations.
$\mathcal{P}\left(\xi, \mathbf{y}\right)$	Likelihood of the state evolution of the system, with observations $\mathbf{y}$.
$\mathcal{P}\left(\mathbf{y}^{i,t} \vert \xi^{i,t}\right)$	Conditional probability of the travel time observations $\mathbf{y}^{i,t}$, given that link i is in state $\xi^{i,t}$ during time interval t.
$\mathcal{P}(N_{\text{in}}^{i,j,t} \,:\, j \in L_{\text{out}}^k \cup k_{\text{out}})$	Probability that $(N_{\text{in}}^{i,j,t})_{j \in L_{\text{out}}^k \cup k_{\text{out}}}$ vehicles leave link i and are assigned to link j during time interval t.

6.2 Statistical model formulation

Traffic model and assumptions

On each link of the network, the model follows the same standard assumptions as the ones presented in Section 5.1. These assumptions are commonly made in the transportation engineering literature on the dynamics of traffic flow. The chapter builds on the model of Chapter 5 to model the dynamics of the network. The model assumptions are stated as follows.

1. *Macroscopic LWR model*, as introduced in Chapter 5.

2. *Characterization of the state of traffic assumption*: For each link of the network, traffic conditions are characterized by a traffic state variable. This state variable represents the number of vehicles that stop on the link per light cycle. It is denoted ξ (generically) and will appear with indices later in the text when required. It is related to the queue length l by $\xi = l\rho_{\max}$.

3. *Discrete time dynamical system*: Let t_0 and Δ_t respectively denote the initial time and the time discretization. Typically, Δ_t is in the order of *five to fifteen* minutes. During each interval, $[t_0 + t\Delta_t,\, t_0 + (t+1)\Delta_t]$ for $t \in \{0, \dots, T\}$, the state and flow entering each link are considered constant. The parameters of traffic signals (red time, R and cycle time, C) are specific to ech time of day (*e.g.* Mid-week evening rush hour), during which they do not change. During each time interval of duration Δ_t, the system exhibits a periodic behavior. The periodicity is dictated by the period of the traffic light. The queue length changes over time, but the fundamental characteristics of the queue (e.g. the maximum length reached during a cycle and thus the number of vehicles stopping in the queue per cycle) remain constant within a time interval. In particular, the number of vehicles stopping per cycle is constant for a link i and a time interval t and is denoted $\xi^{i,t}$.

4. *Transition model*: According to the time discretization assumption, the state variables $\xi^{i,t}$ are piecewise constant, with possible discontinuities at the end of each interval. These transitions model the information propagation on the road network by taking into account the spatio-temporal dependencies of the state of the links. Based on the conservation of vehicles, these transitions are modeled using an approach derived from the *Cell Transmission Model* [52]. The state of a link during a time interval depends on the state of this link and the adjacent links during the previous time interval. This dependency represents the effect of supply and demand of downstream and upstream links respectively. The dynamic evolution of the traffic state of each link is probabilistic and parameterized by *turn movement probabilities* from and to neighboring links and *arrival rates* of vehicles in the network. The parameters of the turn movements can be learned historically.

5. *Conditional independence assumptions:* The dynamics are represented using a graphical model, which characterizes the conditional independence assumptions between the state variables (representing traffic conditions) and the observations. A graphical model is a graph in which the nodes represent random variables. The edges denote the conditional independence structure between the random variables. For more background on graphical models, please refer to [127]. The random variables represented by the present graph are (i) the *state variables* $\xi^{i,t}$, number of vehicles stopping on a link per light cycle, on each link i at each time interval t and (ii) the set of *travel times* $\mathbf{y}^{i,t}$ measured on each link i at each time interval t. The conditional independence assumptions between the random variables can be formulated as follows:

    a) Travel time measurements on link i for time interval t are independent and identically distributed given the state $\xi^{i,t}$ (number of vehicles stopping on a link per light cycle) of this link at this time interval. This means that given the state $\xi^{i,t}$, a travel time on link i during time interval t does not depend on the realization of the other travel time measurements on link i during time interval t. Note that the *conditional* independence assumption is much less strong than assuming independence between travel times.

    b) Travel time measurements on link i for time interval t are independent from all the other random variables given the state $\xi^{i,t}$ of this link at this time interval. This means that given the state $\xi^{i,t}$, a travel time on link i during time interval t does not depend on the realization of the other random variables. It does not depend on the states of the other links at any time intervals nor on the state of link i during time intervals previous or posterior to time interval t nor on the realization of other travel time measurements.

    c) Conditioned on the state of the adjacent links (including itself) at the previous time interval t, the state $\xi^{i,t+1}$ of link i at time interval $t+1$ is independent from the travel time measurements from anterior time periods and all other anterior state variables. This means that given the states $\xi^{j,t}$ of the adjacent links of link

i (including link i), the state of link i during time interval $t+1$ does not depend on the realization of the anterior random variables. It does not depend on the states of the non adjacent links at time interval t nor on the state of any link at time intervals anterior to $t-1$ nor on the realization of travel time measurements during time intervals interior to t. In the following, the set of adjacent links of link i (including link i) is referred to as the *neighbors* of link i.

6. *Data availability assumption:* The data consists of point to point travel time measurements from a small subset of vehicles traveling on the network. Measurements from the past are stored and accessible in real time. The *Mobile Millennium* system [4] provides such data.

Remark 6.1 (Effect of dedicated lanes)**.** *As for Chapter 5, the derivations do not detail the effect of dedicated lanes. A potential approach would be to introduce parameters for each lane of the arterial link which can be estimated from the probe vehicles with the corresponding turn movement (known from the path of the vehicle). As for other modeling choices, the decision to take into account dedicated lanes is a trade-off between the model complexity and the level of information contained in the data. Note that, even though the different lanes of a link are not modeled as distinct queues, the number of lanes is a key characteristic of the network as it influences the capacity of each link. The number of lane of a link i is denoted κ^i.*

Arterial traffic dynamics

Chapter 5 presents the horizontal queuing theory which is used to derive the probability distributions of travel time (Section 5.3). The following recalls the results of these derivations and presents how they are extended to represent traffic states with the number of vehicles in the queue ξ instead of the queue length l. For notational simplicity, the reference to the link i and the time interval t are omitted in this section.

Let ξ_s denote the *saturation number of vehicles*. It corresponds to the maximum number of vehicles that can exit the link in the duration of a cycle. It is related to l_s as $\xi_s = \rho_{\max} l_s$. Remark that at the transition between the undersaturated and the congested regimes, $\xi = \xi_s$. As pointed out in [137], there is a smooth transition between these regimes. The distinct regimes are introduced for the mathematical derivations of the travel time distributions, in particular because of the presence of a remaining queue in the congested regime.

In the undersaturated regime, the duration between the time when the light turns green and the time when the queue fully dissipates is called the *clearing time* denoted τ, sometimes also referred to as *saturation green time*. Recalling that ξ denotes the number of vehicles which stop in the queue per cycle, the relation with the clearing time is given by

$$\tau = (C - R)\frac{\xi}{\xi_s}. \tag{6.1}$$

Notice that when $\xi = \xi_s$ the clearing time is equal to $C-R$, *i.e.* at the transition between the undersaturated and the congested regime, the queue finishes to dissipate as the signal turns red.

In the congested regime, the number of vehicles which stop in the queue per cycle is denoted ξ. It is the sum of the number of vehicles which stop in the triangular queue (ξ_s vehicles) and in the remaining queue ($l_r/\rho_{\max}$).

All notations introduced up to here are illustrated for both regimes in Figure 5.1, except ξ and ξ_s which represent number of vehicles (and are related to the corresponding queue lengths through the maximum density $\rho_{\max}$).

Network model and associated notation

The following variables are learned historically by the model. They are sufficient to characterize the travel time distribution on each link of the network, conditioned on the number of vehicles in the queue (dynamic state variable):

- Static model parameters: These parameters are learned historically and valid for a given *time of day* (representing several time intervals of duration Δ_t. They consist of the cycle time, C, red time, R, saturation number of vehicles, ξ_s, parameters of the free flow pace distribution, θ_p.
- Traffic state: It represents the number of vehicles in the queue, denoted ξ. It is estimated dynamically (in real-time).

As mentioned in Propositions 5.1 and 5.2, the aggregation of the periodic dynamics limits the number of parameters which can be estimated. In particular, the model only uses two parameters derived from the fundamental diagram (p_f and ξ_s). These two parameters allow for the computation of the critical density and the capacity but not the maximum density $\rho_{\max}$. The maximum density (effective length of the vehicles) may be estimated off-line with other means (*e.g.* The *Highway Capacity Manual* [209]). It may remain constant over time and be the same for links with similar properties.

The *time evolution* of the state of traffic depends on the probabilistic *assignment* of vehicles to the links of the network. Let L^k_{in} (resp. L^k_{out}) denote the set of incoming (resp. outgoing) links of intersection k. Each intersection may include dummy links representing sinks, k_{out} and sources, k_{in} which model vehicles arriving or leaving the network at intersection k (parking, residential roads, etc.). At time interval t, $n^{i,t}_{\text{in}}$ (resp. $n^{i,t}_{\text{out}}$) denotes the number of vehicles arriving (resp. leaving) link i during a cycle. Similarly, $N^{i,t}_{\text{in}}$ (resp $N^{i,t}_{\text{out}}$) represents the total number of vehicles arriving (resp. leaving) the link during the duration Δ_t of time interval t. In the derivations at time interval t, for two adjacent links i and j (with i upstream of j), $n^{i,j,t}_{\text{in}}$ (resp. $N^{i,j,t}_{\text{in}}$) is the number of vehicles arriving to link j from link i during a cycle (resp. during time interval t). These notations are summarized in Figure 6.1.

The dynamics of the state of traffic are fully characterized by the turn movements on the network. For an incoming link $i \in L^k_{\text{in}}$ and an outgoing link $j \in L^k_{\text{out}} \cup k_{\text{out}}$ of intersection

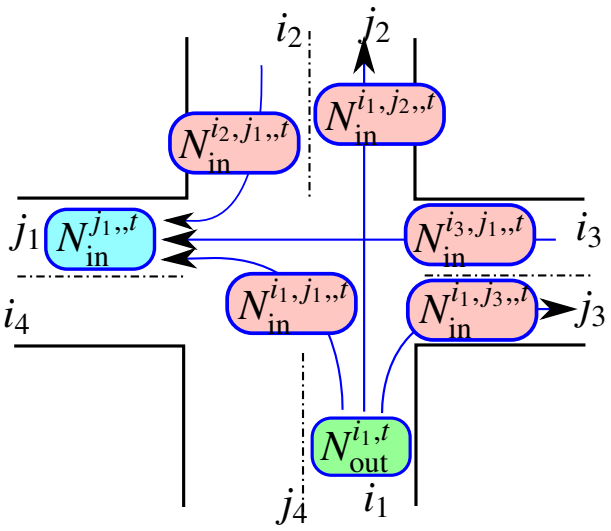


Figure 6.1: Schematic representation of an intersection k illustrating the definition of incoming links, outgoing links, turn ratios and vehicle assignment. The incoming links are denoted $L^k_{\text{in}} = \{i_1, i_2, i_3, i_4\}$ and the outgoing links are denoted $L^k_{\text{out}} = \{j_1, j_2, j_3, j_4\}$.

k, the probability of going from link i to link j is called a *turn probability* and denoted $\nu^{i,j}$. These variables are non negative and satisfy $\sum_{j \in L^k_{\text{out}} \cup k_{\text{out}}} \nu^{i,j} = 1$. The presence of a source at the intersection is modeled for each outgoing link of the intersection $j \in L^k_{\text{out}}$ via a Poisson process with intensity λ_j.

The following section summarizes the derivation of probability distributions $g^i(y_{x_1,x_2}|\xi^{i,t})$ for the travel time y_{x_1,x_2} between two locations x_1 and x_2 on a link i of the network, conditioned on its state $\xi^{i,t}$ at time interval t. The set of travel time measurements received for link i during time interval t is denoted $\mathbf{y}^{i,t}$. The section also derives transition probabilities for the number of stopped vehicles per cycle on a link i at time $t+1$ given the number of stopped vehicles of the neighboring links at time t. The full set of notation used in this article is available in Section 6.1 for convenience.

6.3 Probabilistic model of traffic dynamics

Modeling the travel time distributions between any two points on a link

The derivations of Section 5.3 provide the probability distribution of travel times between *arbitrary* locations x_1 and x_2 on a link of the network. The general method to derive these distributions is as follows:

- From horizontal queuing theory, derive the probability of delay δ_{x_1,x_2} experienced between the two locations x_1 and x_2 on the link, parameterized by the network parameters and the traffic state.
- Model the differences in driving behavior, as presented in Section 5.1. Considering a free flow pace p_f with probability distribution φ, the probability distribution of free flow travel times $y_{f;\,x_1,x_2}$ between locations x_1 and x_2 is computed by scaling φ since $y_{f;\,x_1,x_2} = p_f(x_1 - x_2)$.
- Derive the probability distribution of travel times y_{x_1,x_2} between locations x_1 and x_2 as the sum of two independent random variables: the delay δ_{x_1,x_2} and the free flow travel time $y_{f;\,x_1,x_2}$.

In the following, quantities are indexed by i (and sometimes t) to indicate that they refer to link i (and to time interval t). For a link i and a time interval t, the resulting travel time probability distribution between any two points on the link are parameterized by the network parameters and the points on the link (x_1 and x_2). The probability distribution of travel time y_{x_1,x_2} between x_1 and x_2 is conditioned on the traffic state $\xi^{i,t}$ and denoted $g^i(y_{x_1,x_2}|\xi^{i,t})$. The dependency on the network parameters is implicit and only reminded by the indexing of g by i.

Modeling the spatio-temporal dependencies: transition probabilities

The spatio-temporal dependencies between the links of the network are modeled with a transition probability on the state of each link i at time $t+1$ given the state of the neighbors at time t. For link i, this transition probability is parameterized by the turn probabilities and intensities of the Poisson processes for the arrival vehicles.

In this chapter, all the lanes of a link are assumed to follow the same dynamics. In particular, each lane of link i is in state $\xi^{i,t}$ during time interval t. The red time R^i, the cycle time C^i, the saturation number of vehicles ξ_s^i and the parameters of the free flow pace θ_p^i are the same for each lane of the link. The derivations can readily be extended to account for variable queue length and signal phases by considering different red times and queue lengths for each lane of the link.

Number of vehicles leaving a link in a cycle

The derivations in this section are valid for any link i of the network at any time interval t.

In a congested regime, there are more vehicles on the link than can exit during a cycle. The number of vehicles that exit the link during a cycle within time interval t is $n_{\text{out}}^{i,t} = \kappa^i \xi_s^i$.

In an undersaturated regime, the signal time is divided into three distinct phases: the *red phase* during which the light is red and no vehicle goes through the intersection (duration R^i), the *clearing phase* (introduced in Section 6.2, with duration $\tau^{i,t}$) and the *free-flowing*

phase during which the vehicles go through the intersection without stopping. The duration of the clearing time (and of the free flowing phase) depends on the time interval t since it depends on the state of the link $\xi^{i,t}$.

The duration of the free flowing phase is the remaining duration of the cycle after the red phase and the clearing phase, with duration $C^i - (R^i + \tau^{i,t})$. The number of vehicles exiting the link during a cycle is the sum of the vehicles exiting the link after stopping in the triangular queue ($\kappa^i \xi^{i,t}$) and the vehicles exiting during the free-flowing phase. For an arrival density $\rho_a^{i,t}$, it follows that

$$n_{\text{out}}^{i,t} = \kappa^i \left(\xi^{i,t} + \rho_a^{i,t} v_f^i (C^i - (R^i + \tau^{i,t})) \right). \tag{6.2}$$

In each lane, $\xi^{i,t}$ vehicles stop in the triangular queue. They exit during the clearing time ($\tau^{i,t}$) at the maximum flow ($q_{\max}^i = v_f^i \rho_c^i$):

$$\xi^{i,t} = v_f^i \rho_c^i \tau^{i,t}. \tag{6.3}$$

From equation (5.2), it follows that the ratio between the arrival and the critical density for each lane of the link is given by

$$\frac{\rho_a^{i,t}}{\rho_c^i} = \frac{\tau^{i,t}}{\tau^{i,t} + R^i}. \tag{6.4}$$

Combining equations (6.3) and (6.4) in equation (6.2), the number of vehicles that leave a link in a cycle C^i is

$$\begin{array}{rcll} n_{\text{out}}^{i,t} & = & \kappa^i \left(\xi^{i,t} + \rho_c^i v_f^i \frac{\tau^{i,t}}{\tau^{i,t}+R^i} (C^i - (R^i + \tau^{i,t})) \right) & \text{using equation (6.4),} \\ n_{\text{out}}^{i,t} & = & \kappa^i \, \xi^{i,t} \frac{C^i}{\tau^{i,t}+R^i} & \text{using equation (6.3).} \end{array} \tag{6.5}$$

The number of vehicles leaving the link during time interval t (of duration Δ_t) is derived from (6.5) as $N_{\text{out}}^{i,t} = n_{\text{out}}^{i,t} \frac{\Delta_t}{C^i}$. Incorporating the equation of $\tau^{i,t}$ from (6.1), $N_{\text{out}}^{i,t}$ is given by:

$$N_{\text{out}}^{i,t} = \kappa^i \, \min(\xi^{i,t}, \xi_s^i) \frac{\Delta_t}{R^i + (C^i - R^i) \frac{\min(\xi^{i,t}, \xi_s^i)}{\xi_s^i}}. \tag{6.6}$$

Dynamic evolution of the state

Each vehicle arriving from link i at an intersection k is assigned to an outgoing link $j \in L_{\text{out}}^k \cup k_{\text{out}}$ with probability $\nu^{i,j}$ (possibly leaving the network through the sink k_{out}). Each vehicle is assigned independently from the other ones. According to this model, the random vector $(N_{\text{in}}^{i,j,t})_{j \in L_{\text{out}}^k \cup k_{\text{out}}}$ of vehicles assigned to the different outgoing links of the intersection has a multinomial distribution with parameters $N_{\text{out}}^{i,t}$ and $(\nu^{i,j})_{j \in L_{\text{out}}^k \cup k_{\text{out}}}$ such that,

$$\mathcal{P}(N_{\text{in}}^{i,j,t} \,:\, j \in L_{\text{out}}^k \cup k_{\text{out}}) = \begin{cases} \dfrac{N_{\text{out}}^{i,t}!}{\prod\limits_{j \in L_{\text{out}}^k \cup k_{\text{out}}} N_{\text{in}}^{i,j,t}!} \prod\limits_{j \in L_{\text{out}}^k \cup k_{\text{out}}} (\nu^{i,j})^{N_{\text{in}}^{i,j,t}} & \text{if } \sum\limits_{j \in L_{\text{out}}^k} N_{\text{in}}^{i,j,t} = N_{\text{out}}^{i,t}, \\ 0 & \text{otherwise.} \end{cases}$$

If the intersection has a source k_{in}, vehicles arrive to the outgoing links j of the intersection according to a Poisson process of intensity λ^j. The probability that $N_{\text{in}}^{k_{\text{in}},j,t}$ vehicles arrive to link j from the source during Δ_t is

$$\mathcal{P}(N_{\text{in}}^{k_{\text{in}},j,t}) = \frac{(\Delta_t \lambda^j)^{N_{\text{in}}^{k_{\text{in}},j,t}} e^{-\Delta_t \lambda^j}}{N_{\text{in}}^{k_{\text{in}},j,t}!}.$$

The number of vehicles arriving to link j from the incoming links of intersection k ($(N_{\text{in}}^{i,j,t})_{i \in L_{\text{in}}^k \cup k_{\text{in}}}$) and the state of link j at time t ($\xi^{j,t}$) provide the state $\xi^{j,t+1}$ of link j at time $t+1$: (i) compute the balance of vehicles between the incoming and the outgoing vehicles at time t and (ii) update the state of the link for time $t+1$ accordingly. The details of this transition are as follows:

- Balance of vehicles on link j at time interval t: During a time interval Δ_t, there are $N_{\text{out}}^{j,t}$ vehicles exiting link j and $N_{\text{in}}^{j,t}$ vehicles arriving in link j, which corresponds to a balance of $\Delta N^{j,t} = N_{\text{in}}^{j,t} - N_{\text{out}}^{j,t}$ additional vehicles. Note that a negative number represents a decrease in the number of vehicles on the link. If link j has several (κ^j) lanes, the increase or decrease in the number of vehicles is the same for all lanes. This can be adapted for a model with lane-specific link and intersection parameters.

- Update of the state at time interval $t+1$:

  1. *Undersaturated regime with arrival flow inferior to the capacity*: At time t, link j is undersaturated ($\xi^{j,t} \leq \xi_s^j$) and the number of vehicles arriving per cycle is less than the maximum throughput per cycle ($n_{\text{in}}^{j,t} \leq \kappa^j \xi_s^j$). These two conditions imply undersaturated conditions for link j during time intervals t and $t+1$. The queue fully dissipates by the end of each light cycle and the outflow at time $t+1$ equals the inflow at time t ($N_{\text{out}}^{j,t+1} = N_{\text{in}}^{j,t}$). The inversion of equation (6.6) provides the expression of the state at $t+1$. Note that in this case, Equation (6.6) is simplified since the number of vehicles in the queue is less than the saturation number of vehicles ($\min(\xi^{j,t+1}, \xi_s^j) = \xi^{j,t+1}$).

$$\xi^{j,t+1} = \frac{N_{\text{out}}^{j,t+1} R^j \xi_s^j}{\kappa^j \Delta_t \xi_s^j - (C^j - R^j) N_{\text{out}}^{j,t+1}} = \frac{N_{\text{in}}^{j,t} R^j \xi_s^j}{\kappa^j \Delta_t \xi_s^j - (C^j - R^j) N_{\text{in}}^{j,t}}$$

2. Other transitions: If the regime was congested or if the number of vehicles arriving on the link per cycle is greater than the maximum throughput of the link, there is a constant increase (or decrease) in the number of vehicles on the link through the time period t. The number of vehicles stopping in the queue for time interval $t+1$ is given by the balance of vehicles:

$$\xi^{j,t+1} = \xi^{j,t} + \frac{\Delta N^{j,t}}{\kappa^j}.$$

Statistical modeling framework

Arterial traffic conditions vary dynamically over space and time. The conditional independence assumptions of Section 6.2 are represented using a probabilistic graphical model known as a *Dynamic Bayesian Network* (DBN). The DBN models the stochastic dynamics of the traffic states (number of vehicles stopping in a cycle) of each link in the arterial network. The traffic states are not observed directly; these variables are considered *hidden*. On each link, the travel time distribution is conditioned on the (hidden) state of the link. The travel time of the probe vehicles traveling through the arterial network provide sparse observations of the state variables. Figure 6.2 illustrates the model representation of link states and probe vehicle observations. Each circular node in the graph represents the state of a link in the road network. The forward arrows indicate the local spatial dependency of links from one time period to the next. Each square node in the graph represent probe vehicle observations on the link to which it is attached. The number of observations for a time interval t and a link i is denoted $I^{i,t}$ For more background on DBNs, please refer to [170].

The observations are successive GPS measurements of vehicle trajectories (approximately one per minute). The issues of filtering the noise of the GPS to estimate the most likely location of the vehicle when the measurement was generated and inferring the path taken by the vehicle are not addressed in this article. There are multiple approaches to solving this problem including using statistical filtering [120, 206]. The numerical results of this dissertation which are based on sparsely sampled probe data are based on a *Map-Matching and Path-Inference* filter which combines a model of driving behavior and GPS noise in a *Random Markov Field* to accurately map the GPS measurements on the road network and reconstruct the most likely route of the vehicle between successive locations reports [120]. In this thesis, the data is assumed to available in the following format: most likely measurement locations on the road network as well as the most likely path of the vehicle between successive GPS mappings. The estimation of the following parameters fully specifies the DBN:

- The probability of the state ξ at the start of the experiment. For each link, it is denoted $\pi^i(\xi)$. It represents the probability that link i has ξ stopping vehicles at the initial time,
- The transition probabilities, parameterized by the turn probabilities $\nu^{i,j}$ and intensities of the Poisson processes λ^j,

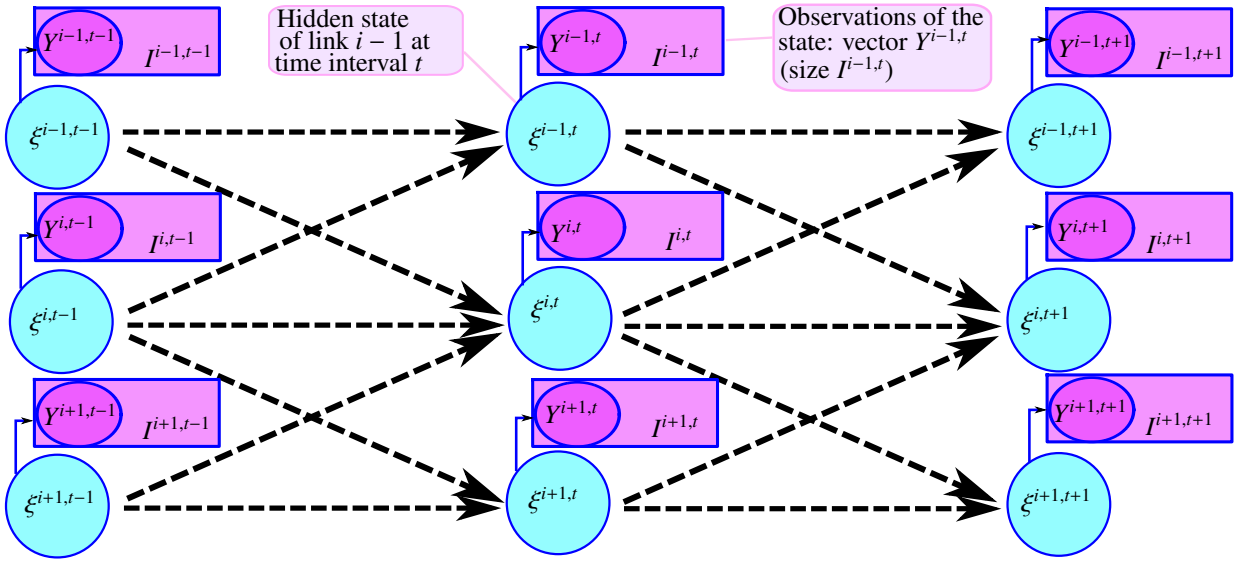


Figure 6.2: Spatio-temporal model of arterial traffic evolution represented as a Dynamic Bayesian Network. The circular nodes represent the (hidden) discrete states $\xi^{i,t}$ of traffic for each link i at each time interval t. The rectangular nodes represent the $I^{i,t}$ travel time observations (denoted $Y^{i,t}$) of each link i at each time interval t. The doted arrows represent the stochastic spatio-temporal dependencies between the states. The plain line arrows represent the dependency of the travel time distributions on the *hidden* traffic state.

- The distribution of travel time g^i on each link i of the network, parameterized by the link parameters and conditioned on the state of the link.

The traffic state is constant during each time intervals of duration Δ_t, typically chosen between 5 and 15 minutes (time discretization assumption), and the link and intersection parameters may be assumed constant for several of these time intervals representing specific *times of day* (*e.g.* morning rush hour, mid-day, afternoon rush hour, evening, night). The present chapter focuses on the estimation of the parameters for a given *time of day* and the dynamic evolution of the state within this *time of day*. Chapters 8 and 9 study data-driven algorithms which analyze the network at a global scale to automatically detect changes in the traffic dynamics.

Given the state of a subset of links, the travel time distributions on these links are assumed to be independent random variables. As mentioned in Chapter 5, travel time distributions across links are not independent (due to light synchronization, platoons, and other factors), although it is a reasonable approximation in many cases. See [188, 121] for investigations on the effect of correlated distributions. These models have the potential to capture more complex dependencies in the arterial road network. Note that it is possible to generalize the model of this chapter to take into account ideas from this research. For example, by

considering the probability of the number of stops on a link given the number of stops on the previous link of the trajectory.

6.4 Historical learning and real-time inference

There is a complex pattern of dependencies among the travel times sent by the probe vehicles. The goal of this section is to develop an algorithm to learn the stochastic dependency between these observations in order to perform estimation and prediction on the arterial network. This learning is done off-line, from historical data, and is used to perform estimation and prediction in real-time. Modeling the dependency between the observations directly is a difficult task because it does not exploit the underlying structure of the dynamical system, represented by the conditional independence assumptions. The variables $\xi^{i,t}$ are introduced to exploit the structure of the dynamical system. They represent the discrete state of each link at each time interval.

Since these variables are not observed directly, they are called *latent* or *hidden* variables. The probe vehicle travel times are noisy, sparse observations of these variables. The parameter estimation problem would be simplified if the state variables ($\xi^{i,t}$) were observed directly. Without observing ($\xi^{i,t}$), the likelihood function is a marginal probability, obtained by summing (or integrating in the continuous case) over the latent variables. Marginalization couples the parameters and obscures the underlying structure of the likelihood function.

The *Expectation Maximization* algorithm (EM algorithm) is a widely used algorithm to learn the dependencies among the observations while exploiting the structure of the stochastic dynamic evolution [62]. This choice is supported by the following two realizations: (1) given the parameters of the model and the path observations, it is possible to estimate the most likely state of each link at each time interval and (2) given the state of each link at each time interval, it is possible to compute the parameters of the model (turn probabilities, intensities of the Poisson processes and parameters of the network) which maximize the likelihood of the observations. The EM algorithm iteratively leverages these two realizations and is guaranteed to converge to a local optima of the likelihood function. More detailed information on the EM algorithm can be found in the literature [62] and a short introduction is given below.

Another challenge of the graphical model approach is that the link travel times are not observed directly. The path between two consecutive measurements can span several links of the network. This difficulty is addressed by computing the most likely link travel times that make up the path of the probe vehicle (*travel time allocation*), as introduced in Section 5.4. This section first introduces the EM algorithm and details its two iterative steps: *Expectation* step (E step) and *Maximization* step (M step) in the case of traffic estimation.

Introduction on EM algorithm

The EM algorithm allows to exploit the underlying structure of the dynamical model, even though the latent variables ($\xi^{i,t}$) are not observed. It is an iterative algorithm consisting in two steps:

- *The Expectation step (E step)* computes the joint probability distribution of the latent variables $\xi^{i,t}$ (number of vehicles in the queue for each link i and each time interval t) given the observed variables $\mathbf{y}^{i,t}$ (allocated travel times for each link i and each time interval t) and the current values of the parameters (signal parameters, turn ratios, driving behavior, saturation number of vehicles). In the Bayesian approach to dynamic state estimation, this computation is known as a *smoothing* step: at each time t, the algorithm computes the joint probability distribution of the state variables $(\xi^{i,t})_{i,t}$ given all the historical data available. In practice, the smoothing step is replaced by a filtering step for efficiency. The filtering step only uses observations received up to (and including) time t (instead of all historical measurements) to compute the joint probability distribution of the state variables $(\xi^{i,t})_i$. Such a filtering step consists of essentially two stages: prediction and update. The prediction uses the transition probabilities to predict the state probability distribution from one time interval to the next. The update operation uses the latest available measurements to modify the state probability distribution using Bayes theorem.
- *The Maximization step (M step)* optimizes the parameters (signal parameters, turn ratios, driving behavior, saturation number of vehicles) based on the estimation of the joint probability distribution of the latent variables. This step has the same complexity as if the latent variables were observed.

As illustrated Figure 6.2, a dynamic Bayesian network is a directed graphical model, in which each random variable is represented by a node of the graph. Each generic random variable x_i has a set of parents, denoted x_{π_i} such that the joint probability $p(x_1, \ldots, x_n)$ of $x_1, \ldots, x_n$ can be factored as

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | x_{\pi_i}),$$

where $p(x_i|x_{\pi_i})$ is the probability of x_i given that its parents (in the directed graph) have the realization x_{π_i}. For the application of interest, the random variables represent the traffic states $\xi^{i,t}$ and the travel time observations $\mathbf{y}^{i,t}$ on each link i of the network and at each time interval t. The conditional independence assumptions and the associated directed graphical model representation provide a compact, factored, representation of the joint distribution of these random variables:

$$\mathcal{P}(\xi, \mathbf{y}) = \left(\prod_{t=0}^{T-1} \prod_{i \in \mathcal{I}} \mathcal{P}(N_{\text{in}}^{i,j,t} : j \in L_{\text{out}}^{k_i} | \xi^{i,t})\right)$$

Probability of the assignment of the vehicles from link i to the outgoing links of the intersection, for each link and each time interval excepted the last one which corresponds to the end of the experiment.

$$\times \left(\prod_{t=0}^{T} \prod_{i \in \mathcal{I}} \mathcal{P}(\mathbf{y}^{i,t} | \xi^{i,t})\right)$$

Probability of the travel time observations $\mathbf{y}^{i,t}$ conditioned on the state of the link $\xi^{i,t}$, for each link i and each time interval t.

$$\times \left(\prod_{i \in \mathcal{I}} \pi_i(\xi^{i,0})\right)$$

Probability that link i is in state $\xi^{i,0}$ at the initial time interval, for each link i.

Note that given the state of the links at a time interval, the number of vehicles from link i assigned to the outgoing links j and the number of vehicles entering or exiting the network through the sources and sinks determine the state evolution for all the links of the network. For convenience, these probabilities are used in the expression of $\mathcal{P}(\xi, \mathbf{y})$ instead of referring directly to the probability of the number of vehicles in the queue of link i at time interval $t+1$ given the number of vehicles in the queue of the neighboring links.

If the hidden variables $\xi^{i,t}$ were observed, the likelihood optimization would amount to maximizing $\mathcal{P}(\xi, \mathbf{y})$ with respect to the link and intersection parameters. It is more common (in particular for numerical stability) to consider the logarithm of $\mathcal{P}(\xi, \mathbf{y})$, referred to as the *complete log-likelihood* because it corresponds to the log-probability of the complete set of random variables for a given value of the parameters. Given that the variables $\xi^{i,t}$ are in fact not observed, the complete log-likelihood is a random quantity, and cannot be maximized directly. Given a distribution, denoted $q(\xi|\mathbf{y})$, the *expected complete log-likelihood* is a deterministic function of the parameters, denoted $\left\langle l_c(\mathbf{y}, \xi) \right\rangle_q$ and defined as follows. It corresponds to the average of the complete log-likelihood, over the realizations of ξ, when $q(\xi|\mathbf{y})$ is chosen as the averaging distribution:

$$\left\langle l_c(\xi, \mathbf{y}) \right\rangle_q = \sum_{\xi} q(\xi|\mathbf{y}) \ln(\mathcal{P}(\xi, \mathbf{y}))$$

A proof leveraging Jensen's inequality shows that the log-likelihood can be maximized by iteratively (i) choosing the proposal distribution $q(\xi|\mathbf{y})$ as the joint distribution of the state variables computed by the E step and (ii) maximizing on the parameters of the observations (R^i, C^i, ξ_s^i, θ_p^i, $i \in \mathcal{I}$) and of the dynamics ($\nu^{i,j}$, λ^i, for $i \in \mathcal{I}$ and for j outgoing link of i).

E step: Particle filtering

The E step performs filtering given the current values of the parameters and the travel time observations collected from historical data. The Dynamic Bayesian Network used to model traffic dynamics is a multiply connected belief network (at least one pair of variables has more than one undirected path connecting them), in which probabilistic inference is NP-hard [47]. The dimension of the state space (number of possible configurations for the variables $\xi^{i,t}$) grows exponentially with the number of links in the network, making an explicit representation of the probability distributions intractable. In such networks, algorithms performing

probabilistic inference have a time complexity that, in the worst case, is exponential in the number of hidden variables in the network. Approximation algorithms are required to perform probabilistic inference. Algorithms such as Monte Carlo simulation [171], variational methods [128], and belief state simplification [30] are commonly used to approximate probabilistic inference. To maintain a compact approximation of the state probability distribution, this chapter investigates a *Monte Carlo* simulation approach called *particle filtering* and also referred to as *bootstrap filtering* or the *condensation algorithm* [192, 9]. Particle filtering is an approximation of a recursive Bayesian filter algorithm using Monte Carlo simulations which has successfully been implemented for highway traffic estimation [38]. The idea is to represent the distribution by a set of random samples with associated weights (importance weights). As the number of samples increases, this Monte Carlo approximation tends to the exact optimal Bayesian estimate.

The filter is implemented by simulating V particles. Each particle v represents an instantiation of the *time evolution of the traffic state of the network*, *i.e.* a possible succession of traffic states for each link of the network and each time interval. A particle v at time t is represented by a vector of the states of each link and each time interval up to t (denoted $(\xi_v^{i,t'})_{i\in\mathcal{I},\, t'\in\{0,\ldots,t\}}$). At t, each particle has a weight ω_v^t proportional to the probability of having this instantiation of the state evolution given the available data up to time t. The particles explore the possible state space and represent the belief state of the DBN.

Sufficient statistics to compute the expected complete log-likelihood

At time t, the spatio-temporal instantiations $\Xi_v^t = (\xi_v^{i,t'})_{i\in\mathcal{I},\, t'\in\{0,\ldots,t\}}$ of the particles and their associated importance weight ω_v^t form an approximation of the joint probability distribution of the state of the links. Let $\mathbf{y}^{i,t}$ denote the set of travel time observations received on link i during time interval t. Given the travel time observations $\left(\mathbf{y}^{i,t'}\right)_{i\in\mathcal{I},\, t\in\{0\ldots t'\}}$, the probability of observing a state $\Xi^t = (\xi^{i,t'})_{i\in\mathcal{I},\, t'\in\{0\ldots t\}}$ on the network throughout its time evolution is approximated as follows:

$$\mathcal{P}(\Xi^t|\mathbf{y}^{i,t'}; R^i, C^i, \xi_s^i, \theta_p^i : i\in\mathcal{I},\, t'\in\{0\ldots t\}) \approx \sum_{v=1}^{V} \omega_v\, \mathbf{1}_{\Xi^t}(\Xi_v^t).$$

where $\mathbf{1}_{\Xi^t}(\Xi_v^t)$ is equal to 1 if the particle has the state instantiation Ξ^t and to zero otherwise. To derive the expected complete log-likelihood, the sufficient statistics $a^{i,t}(\xi^{i,t})$ and $b^{i,t}(\mathbf{N}^{i,t})$, $c^{j,t}(N_{\text{in}}^{k_{\text{in}},j,t})$ and $d^i(\xi^{i,0})$ are defined as follows.

- The probability that link i is in state $\xi^{i,t}$ at time t, conditioned on the observations received up to time interval t is approximated using the particles and denoted $a^{i,t}(\xi^{i,t})$. It is computed by summing the weights of all the particles that represent a state

instantiation with link i in state $\xi^{i,t}$:

$$a^{i,t}(\xi^{i,t}) = \sum_{v=1}^{V} \omega_v^t \, \mathbf{1}_{\xi^{i,t}}(\xi_v^{i,t}), \qquad \forall t \in \{0, \ldots, T\}, \; \forall i \in \mathcal{I}. \tag{6.7}$$

- For an incoming link i and an outgoing link j of intersection k, $(N_{\text{in}}^{i,j,t})_v$ denotes the number of vehicles going from link i to link j during time interval t for the particle v. Using the particles, the approximation of the probability that $\mathbf{N}^{i,t} = (N_{\text{in}}^{i,j,t}, \; j \in L_{\text{out}} \cup k_{\text{out}})$ vehicles from link i are assigned to the outgoing links L_{out}^k and the sink k_{out} is denoted $b^{i,t}(\mathbf{N}^{i,t})$. It is computed by summing the weights of all the particles that represent an instantiation of the dynamics in which the assignments of the vehicles from link i to the outgoing links (and the sink) is $\mathbf{N}^{i,t}$:

$$b^{i,t}(\mathbf{N}^{i,t}) = \sum_{v=1}^{V} \omega_v^t \, \mathbf{1}_{\mathbf{N}^{i,t}} \left((N_{\text{in}}^{i,j,t})_v, \; j \in L_{\text{out}}^k \cup k_{\text{out}} \right), \qquad \forall t \in \{0, \ldots, T-1\}, \; \forall i \in \mathcal{I}. \tag{6.8}$$

- For an intersection k with a source, $(N_{\text{in}}^{k_{\text{in}},j,t})_v$ is the number of vehicles from the source assigned to each outgoing link j of the intersection. The approximation of the probability that $N_{\text{in}}^{k_{\text{in}},j,t}$ vehicles from the source are assigned to link j is denoted $c^{j,t}(N_{\text{in}}^{k_{\text{in}},j,t})$. It is computed by summing the weights of the particles for which $N_{\text{in}}^{k_{\text{in}},j,t}$ vehicles originating from the source were assigned to link j:

$$c^{j,t}(N_{\text{in}}^{k_{\text{in}},j,t}) = \sum_{v=1}^{V} \omega_v^t \, \mathbf{1}_{N_{\text{in}}^{k_{\text{in}},j,t}} \left((N_{\text{in}}^{k_{\text{in}},j,t})_v \right), \qquad \forall t \in \{0, \ldots, T-1\}, \; \forall j \in \mathcal{I}. \tag{6.9}$$

- Let $d^i(\xi^{i,0})$ be the probability of the state of link i at the initial time. Its approximation with the particles is

$$d^i(\xi^{i,0}) = \sum_{v=1}^{V} \omega_v^0 \, \mathbf{1}_{\xi^{i,0}}(\xi_v^{i,0}). \tag{6.10}$$

Filtering using a particle filter

The filtering step consists in successive prediction and update steps which lead to the computation of $\xi_v^{i,t}$ and ω_v^t for all the particles v, all the links i and all the time intervals t. The prediction and update steps are performed as follows:

- *Update of the state posterior probability distribution at time interval t.* The *posterior* state distribution is computed using the measurements available at time interval t. The weight ω_v^t of each particle is multiplied by the probability of each travel time measurement received at time interval t given the state $\xi_v^{i,t}$ of the particle. The weights of the particles are normalized so that they sum to one.

- *Prediction of the state at time interval* $t+1$. The state distribution is predicted using the parameters of the turn movements and of the Poisson processes of the sources. For each incoming link i and each particle v, the state of the particles provides the number of vehicles leaving link i (Equation (6.6)). These vehicles are randomly assigned to the outgoing links of the intersection (including the sink) according to a multinomial distribution parameterized by the turn probabilities. Similarly, a random number of vehicles (coming from the source of the intersection) is assigned to the outgoing links according to the corresponding Poisson process. This allows for the computation of $(N_{\text{in}}^{i,j,t})_v$ and for the simulation of the state of the particle at time interval $t+1$ according to the dynamic evolution described in Section 6.3. This algorithm is known a *Sequential Importance Sampling* (SIS) particle filter.

- *Improvement to prevent degeneracy: the Sequential Importance Resampling (SIR) algorithm.* A common issue of the SIS particle filter is the degeneracy phenomenon, where after a few iterations, all but one particle have negligible weights. It implies that a large computational effort is devoted to updating particles whose contribution to the posterior distribution is almost zero. To reduce the effects of degeneracy, the particles are *resampled* after the update of the importance weights for time interval t. The modified algorithm is known as *Sequential Importance Resampling* (SIR) or *Sampling Importance Resampling*. The idea of resampling is to eliminate particles that have small weights at time interval t. To resample the particles, V particles are successively chosen randomly (with replacement) from the original set of particles. Particle v is chosen with probability ω_v^t (the weights sum to 1). Each resampled particle has a weight equal to $1/V$. This set of re-sampled particles is used to perform the prediction step of the state probability distribution at time interval $t+1$. Figure 6.3 illustrates the resampling algorithm.

M step: Update of the parameters

For each link i, the travel time distribution g^i, conditioned on the state of the link, is parameterized by the red time R^i, the cycle time C^i, the number of vehicles in the queue at saturation ξ_s^i and the parameters of the driving behavior θ_p^i. The full characterization of the model requires to learn the parameters of the dynamics *i.e.* estimate the turn probabilities $\nu^{i,j}$ and the intensities of the Poisson processes λ^j. The M step uses the sufficient statistics $a^{i,t}(\xi^{i,t})$, $b^{i,t}(\mathbf{N}^{i,t})$, $c^{j,t}(N_{\text{in}}^{k_{\text{in}},j,t})$ and $d^i(\xi^{i,0})$ to update the value of these parameters by maximizing the expected complete log-likelihood, with respect to these parameters. The factored expression of the complete log-likelihood implies a similar structure for the complete log-likelihood:

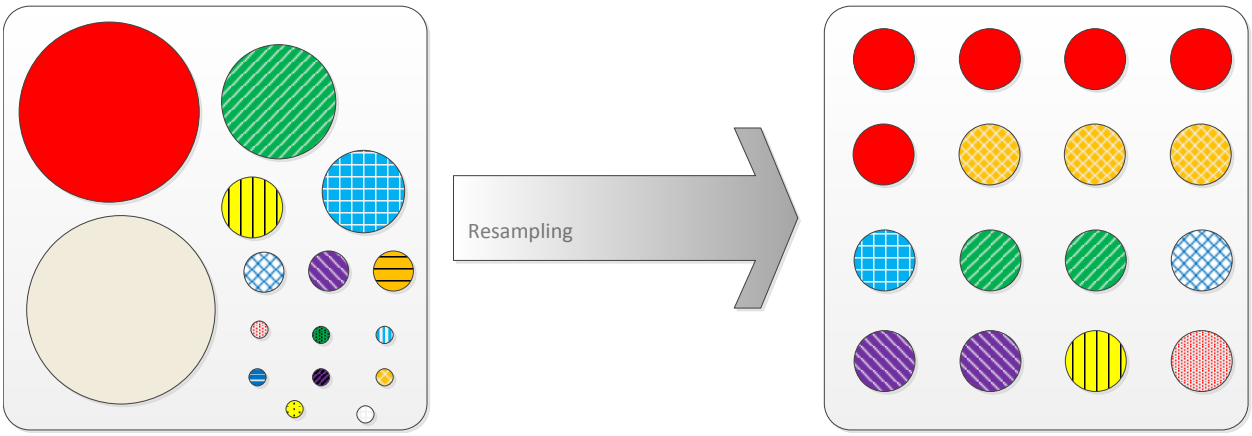


Figure 6.3: Schematic illustration of the resampling algorithm. Each particle is represented by a circle with a diameter proportional to its weight. Each particle is chosen with a probability proportional to its weight, put in the *new* set of particles with weight $1/V$ and then replaced. This process is repeated V times. The intuition is that particles with a large weight are likely to be chosen several times whereas particles with a small weight might not be present after the resampling step.

$$\left\langle l(\xi, \mathbf{y}) \right\rangle = \sum_{t=0}^{T-1} \sum_{i \in \mathcal{I}} \sum_{\mathbf{N}^{i,t}} b^{i,t}(\mathbf{N}^{i,t}) \ln(\mathcal{P}(\mathbf{N}^{i,t}))$$

Assignments of vehicles exiting link i to the outgoing links of the intersection (and the sink) for each link i and each time interval t (excepted the last one).

$$+ \sum_{t=0}^{T-1} \sum_{j \in \mathcal{I}} \sum_{N_{\text{in}}^{k_{\text{in}},j,t}} c^{j,t}(N_{\text{in}}^{k_{\text{in}},j,t}) \ln(\mathcal{P}(N_{\text{in}}^{k_{\text{in}},j,t}))$$

Arrival of vehicles in link j from the source of the intersection, for each link j and each time interval t (excepted the last one).

$$+ \sum_{t=0}^{T} \sum_{i \in \mathcal{I}} \sum_{\xi^{i,t}=0}^{\xi^i_{\max}} a^{i,t}(\xi^{i,t}) \left(\sum_{y_{x_1,x_2} \in \mathbf{y}^{i,t}} \ln\left(g^i(y_{x_1,x_2} | \xi^{i,t}) \right) \right)$$

Travel time measurements, for each travel time y_{x_1,x_2} received on each link i at each time interval.

$$+ \sum_{i \in \mathcal{I}} \sum_{\xi^{i,0}=0}^{\xi^i_{\max}} d^i(\xi^{i,0}) \ln(\pi^i(\xi^{i,0}))$$

Initial state of each link i.

where $\mathcal{P}(\mathbf{N}^{i,t})$ represents the probability (multinomial distribution) of the assignment $\mathbf{N}^{i,t}$ of the vehicles leaving link i to the outgoing links of the intersection (including the sink) and $\mathcal{P}(N_{\text{in}}^{k_{\text{in}},j,t})$ is the probability (Poisson distribution) of the arrival of $N_{\text{in}}^{k_{\text{in}},j,t}$ vehicles in link j from the source of the intersection. The factored structure of the complete log-likelihood, and thus of the expected complete log-likelihood allows the learning of the parameters to be performed independently for the turn probabilities, the intensities of the Poisson processes, the initial state probabilities and for each set of link parameters. The values of $a^{i,t}$, $b^{i,t}$, $c^{j,t}$

Algorithm 2 Maximum likelihood estimation of the parameters of the model with an Expectation Maximization algorithm.

Initialize the parameters (R^i, C^i, ξ_s^i θ_p^i, $\nu^{i,j}$ and λ^j) and the initial state probabilities $\pi_i(\xi)$.

while The algorithm has not converged **do**

E Step [Computation of $a^{i,t}(\xi^{i,t})$, $b^{i,t}(\mathbf{N}^{i,t})$, $c^{j,t}(N_{\text{in}}^{k_{\text{in}},j,t})$ and $d^i(\xi^{i,0})$]

Initialize the E Step: Simulate samples representing the state of the network at the initial time given the initial state probabilities $\pi_i(\xi)$. Each sample has initial weight $\omega_v = 1/V$.

for Time interval $t = 0 : T$ **do**

Allocate the travel times by solving (5.11) for each probe vehicle path.

Update the weight of the particles according to the observations $\mathbf{y}^{i,t}$: $\omega_v = \omega_v \prod_{y_{x_1,x_2} \in \mathbf{y}^{i,t}} g^i(y_{x_1,x_2}|\xi_v^{i,t})$.

Normalize the weights of the particles: Compute the sum Ω of the weights of the particles and normalize the weight of each particle, $\omega_v = \omega_v/\Omega$

Compute $a^{i,t}(\xi^{i,t})$, $b^{i,t}(\mathbf{N}^{i,t})$, $c^{j,t}(N_{\text{in}}^{k_{\text{in}},j,t})$ and $d^i(\xi^{i,0})$ using Equations (6.7)–(6.10).

Re-sample the particles [9]

For each link i, randomly assign the vehicles leaving link i to the outgoing links and the vehicles coming from the sources of the network according to the turn probabilities and intensities of the Poisson processes.

Update the state of the particles according to the number of vehicles that left and arrived on the link during time interval t. Each particle now represents an instantiation of the state of the network at $t + 1$.

end for

M Step [Maximization of the expected complete log-likelihood.]

Update the initial state probabilities $\pi^i(\xi)$ (6.13), the turn probabilities $\nu^{i,j}$ (6.11), the vehicle creation rates λ^i (6.12) and the link parameters $(C^i, R^i, \xi_s^i, \theta_p^i)$ (6.14).

end while

and d^i computed by the E step (Equations (6.7)–(6.10)) are necessary to update the link and intersection parameters as follows (Equations (6.11)–(6.14)).

- The update of the turn probabilities from the incoming link i of intersection k is the solution of the following optimization program:

$$\underset{\nu^{i,j}}{\textbf{maximize}}: \sum_{t=0}^{T-1} \sum_{\mathbf{N}^{i,t}} b^{i,t}(\mathbf{N}^{i,t}) \left(\sum_{j \in L_{\text{out}}^k \cup k_{\text{out}}} N^{i,j,t} \ln(\nu^{i,j}) \right)$$

$$\textbf{s.t.}: \begin{cases} \nu^{i,j} & \geq 0 \quad \forall\, j \in L_{\text{out}}^k \cup k_{\text{out}}, \\ \sum_{j \in L_{\text{out}}^k \cup k_{\text{out}}} \nu^{i,j} & = 1. \end{cases}$$

where the constant terms are ignored. The optimization problem is solved in closed form by writing the *Karush-Kuhn-Tucker* (KKT) conditions. The values of $\nu^{i,j}$ which maximize the expected complete log-likelihood are given by

$$\nu^{i,j} = \frac{\sum_{t=0}^{T-1} \sum_{\mathbf{N}^{i,t}} b^{i,t}(\mathbf{N}^{i,t}) N^{i,j,t}}{\sum_{t=0}^{T-1} \sum_{\mathbf{N}^{i,t}} b^{i,t}(\mathbf{N}^{i,t}) \sum_{j' \in L^k_{\text{out}} \cup k_{\text{out}}} N^{i,j',t}}. \tag{6.11}$$

- For each intersection k with a source k_{in}, the update of the intensities of the Poisson processes for the outgoing links $j \in L^k_{\text{out}}$ is done independently for each link j by solving the following optimization program:

$$\underset{\lambda^j \geq 0}{\textbf{maximize}} : \sum_{t=0}^{T-1} \sum_{N_{\text{in}}^{k_{\text{in}},j,t}} c^{j,t}(N_{\text{in}}^{k_{\text{in}},j,t}) \left(N_{\text{in}}^{k_{\text{in}},j,t} \ln(\Delta_t \lambda^j) - \Delta_t \lambda^j \right)$$

  This optimization problem is solved in closed form as follows:

$$\lambda^j = \frac{1}{\Delta_t} \frac{\sum_{t=0}^{T-1} \sum_{N_{\text{in}}^{k_{\text{in}},j,t}} c^{j,t}(N_{\text{in}}^{k_{\text{in}},j,t}) N_{\text{in}}^{k_{\text{in}},j,t}}{\sum_{t=0}^{T-1} \sum_{N_{\text{in}}^{k_{\text{in}},j,t}} c^{j,t}(N_{\text{in}}^{k_{\text{in}},j,t})}. \tag{6.12}$$

- For each link i, the initial state probability is updated as

$$\pi_i(\xi) = \sum_{v=1}^{V} \omega_v \mathbf{1}_{\xi_v^{i,0}}(\xi). \tag{6.13}$$

  To learn this initial state probability, it is important to run the EM algorithm on several days of data (to reduce overfitting due to fitting the initial state probabilities based on a single day of data). In general, it is advised to run the EM algorithm over several days (weeks or months) of data to improve the learning of all the parameters of the model.

- The link parameters maximize the log-likelihood of the travel time observations $\mathbf{y}^{i,t}$. The travel time allocation enables the optimization problem to decouple into smaller optimization problems, one for each link of the network. The optimization problem for link i is

$$\underset{C^i, R^i, \xi_s^i, \theta_p^i}{\textbf{maximize}} \quad \sum_{t=0}^{T} \sum_{\xi^{i,t}=0}^{\xi^i_{\max}} a^{i,t}(\xi^{i,t}) \left(\sum_{y_{x_1,x_2} \in \mathbf{y}^{i,t}} \ln(g^i(y_{x_1,x_2}|\xi^{i,t})) \right), \tag{6.14}$$

where $g^i(y_{x_1,x_2}|\xi^{i,t})$ represents the probability of observing a travel time y_{x_1,x_2} between x_1 and x_2 on link i given that the state of the link is $\xi^{i,t}$.

Decoupling the optimization problem for each link of the network (instead of solving a large optimization program over the parameters of the entire network) makes it highly scalable as each of the optimization subproblems can be performed in parallel. If the travel time allocation method is not used, then the resulting optimization problem is coupled across the entire network, resulting in a large optimization problem that may not scale well. As mentioned in Chapter 5, it is possible to share some parameters across links of the network to limit the risk of over-fitting and improve the learning and estimation capabilities by incorporating prior information. The parameter sharing couples the optimization problems for the links which share parameters, leading to fewer but larger optimization problems.

Real-time estimation and forecast

Estimating and forecasting traffic conditions in real-time can be achieved after the model parameters and turn probabilities have been learned, *i.e* once the Expectation Maximization algorithm has been run on large amounts of historical data. In real time, the parameters learned by the EM algorithm (which characterize the stochastic dynamics of traffic) are used to perform inference using data available up to the time when the estimate is produced. This is done by running the particle filter to determine the distribution of traffic states given the available data and the learned value of the parameters. Forecast is done by propagating the particle filter forward from the current time interval. Since there is no available data, the filter only performs prediction steps (no update). For both estimation and forecast, the particle filter runs in real time on medium-size networks (the numerical implementation of Section 6.5 considers a network with almost 800 links). However, the EM algorithm needs to run both the particle filter (E step) and the optimization algorithms (M step) for several iterations on large amounts of historical data. For this reason, the EM algorithm is run offline and the model parameters and turn probabilities can be updated periodically (*e.g.* every week or every month).

6.5 Experimental results

The model presented in this chapter relies on assumptions made on the dynamics of traffic flows on each link of the network (Chapter 5) to derive an analytical expression of the probability distribution of travel times, parameterized by traffic variables. The model also relies on assumptions made on the statistical dynamics of traffic flows at intersections (Section 6.2) and derives a probabilistic model of the traffic dynamics on the network.

The experimental results assess the real-time estimation and short-time prediction capabilities of the dynamical model from sparsely sampled probe data. The section describes the validation methodology of the traffic estimation algorithm and presents the results which

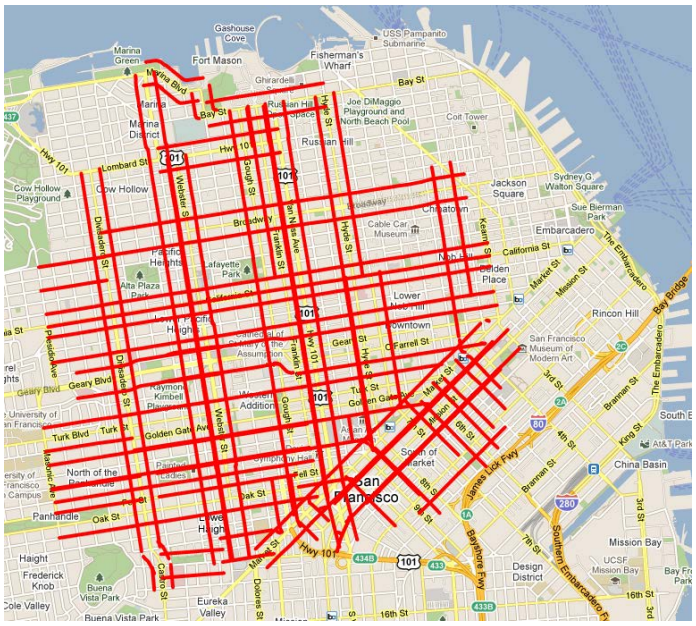


Figure 6.4: Subnetwork of San Francisco, CA used for numerical analysis of the model performance. The network consists of 769 links representing 126 kilometers of roadway.

validate the historical learning capabilities and the real-time estimation and prediction capabilities. The results are compared to a model which only estimates the mean travel time for each link. The model developed in this chapter shows a 16% improvement over this baseline model to estimate mean link travel times. The model also possesses several advantages over the baseline model. These advantages include the ability to predict traffic conditions into the short-term future, the ability to estimate probability distributions of travel times between arbitrary points on the network (instead of just mean link travel time values), as well as the ability to estimate traffic parameters including signal timing and congestions states (queue lengths).

Experiment setup

Beginning in March of 2009, data has been collected from probe vehicles in the San Francisco Bay Area, as part of the *Mobile Millennium* project. One of the available data feeds available through the *Mobile Millennium* system comes from a fleet of over 500 taxis which report their location every minute, along with an identifier and a status (carrying a passenger or not). The status flag allows for the filtering of the taxi stops to load or unload passengers. When a change of status occurs, the measurements directly anterior and posterior to this change of status are discarded. In its raw form, the data cannot be used by the algorithm. This is due to several issues.

- Between successive measurements, the vehicle may travel more than one link and the

path needs to be inferred.

- The measurements provide the location of the vehicles but no information regarding the direction of travel.
- The GPS measurement may be noisy and must be mapped onto the road network.

To overcome these difficulties, a map-matching and path-inference algorithm [120] provides accurate measurement locations and paths followed by the vehicles. The duration between two successive measurements represents the travel time of the vehicle on its path.

The study focuses on a sub-network of San Francisco shown in Figure 6.4. This network consists of 769 links representing 126 kilometers of roadway. The performance of the model is assessed using error metrics computed on previously unseen data: the *Root Mean Squared Error* (RMSE), the *Mean Absolute Error* (MAE) and the *Mean Percentage Error* (MPE) [1].

The Root Mean Squared Error is one of the most widely used metrics to quantify the difference between an estimator and the true value of the quantity being estimated. It measures the average of the squared error. As a result of the squaring of each term, Mean Squared Error heavily weights outliers. For this reason, the analysis also computes the Mean Absolute Error, a common measure of forecast error in time series analysis. Using the convexity of the square function, it is easy to prove that the RMSE is always greater than or equal to the MAE. The Mean Percentage Error computes the average of the percentage error. When the actual values of the process to be estimated vary, this metric allows an equal weighting between the terms, as it is normalized by the actual value of the process.

The model is compared to a baseline model that estimates mean link travel time. For each measurement in the training data set, the pace of the path is allocated to the links of the path with a weight equal to the proportion of the link that was traveled (1 if the full link is traveled, 0 if the link is not traveled at all). The mean pace of a link in the baseline model is computed as the weighted average of the paces on the paths of the training data set. Note that the baseline model does not provide a statistical distribution of travel times but rather a mean pace. This baseline model was chosen because standard time series statistical techniques (weighted moving average, exponential decay, ARMA) are not applicable to the data set because the measurement locations are not fixed and the time at which the measurements are received at a particular location is unknown in advance. This motivates the development of a specific comparison model adapted to the characteristics of the data. In the remainder of this section, *the traffic model* refers to the model developed in this chapter. The *baseline model* refers to the comparison model.

[1] For a vector of E estimations $\hat{\mathbf{x}} = (\hat{x}_e)_{e=1\dots E}$ of the true value $\mathbf{x} = (x_e)_{e=1\dots E}$, the error metrics are defined as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{e=1}^{E}(x_e - \hat{x}_e)^2}{E}}, \quad \text{MAE} = \frac{\sum_{e=1}^{E}|x_e - \hat{x}_e|}{E} \quad \text{and} \quad \text{MPE} = \frac{1}{E}\sum_{e=1}^{E}\frac{|x_e - \hat{x}_e|}{x_e}.$$

	RMSE	MAE	MPE
Traffic Model	25.41	20.23	37.67%
Baseline Model	31.56	25.69	46.20%
Improvement (%)	16.32	17.34	16.29

Table 6.1: Error metrics representing the estimation capabilities of the *Dynamic Bayesian Network* modeling the dynamics of traffic flow from horizontal queuing theory. The metrics are reported on a validation dataset collected during the training days, and set aside for the validation.

Both models run in a hybrid Matlab/Java environment and take advantage of the *Mobile Millennium* system infrastructure which provides simple interfaces for accessing an mathematical abstraction of the physical road network. The internal representation of the road network is built using NAVTEQ maps [2]. The network provides detailed geometry and attributes of the road network. The system also provides an interface for accessing the data feeds stored in the databases (which are map-matched and filtered in separate processes), and writing the outputs of the model to databases for future use (visualization, air quality related to traffic conditions, routing and so on). The historical learning of the parameters and the real time estimation and forecast run on a laptop for moderate size networks.

Validation of the learning capabilities

The model was trained using data collected on the three first Tuesdays of February 2010 from 3pm to 6pm. The discretization time Δ_t is set to fifteen minutes. From all the data collected on these days, the model is trained on a randomly chosen subset representing 70% of the data. The training data set is used to estimate the network parameters (cycle time C, red time R, saturation number of vehicle ξ_s, turn proportions and intensities of the Poisson processes) of each link of the network. At each time interval t, the model also estimates the a posteriori most likely state $\xi^{i,t}$ of each link i using training measurements available up to (and including) time interval t.

The performance of the *learning* capabilities is assessed using the *validation* data set of the training days. The validation data (30% of the full dataset) was previously set aside and not used to train the model. For each path in the validation dataset, the mean travel time is computed from the distribution of travel times using the estimated parameters and a posteriori states. This travel time is compared to the true value experienced by the vehicle to compute the error metrics. The results are reported in Table 6.1. The model shows an improvement of 16% in terms of RMSE compared to the baseline model. Moreover, the model learns parameters of the network (signal timing, saturation number of vehicles) for which it provides realistic estimates. For example, the duration of signal timings (cycle length) has a mean of 86 seconds over the network, with a standard deviation of 17 s, a minimum value of 45 s and a maximum value of 120 s.

Validation of the real time estimation and prediction capabilities

In real time, the model uses the network parameters and turn probabilities learned historically to estimate and predict the state $\xi^{i,t}$ of each link i at each time interval t. At time interval t, the *estimation process* is the computation of the most likely state of the network at time interval t given data received up to and including time interval t. The *prediction at q time steps* is the computation of the most likely state of the network at time interval $t+q$ given data received up to and including time interval t. The prediction at 1 time step is also known as *a-priori* state estimation. The prediction at 0 time step is identical to the *estimation process*.

The most likely state of the network is computed by performing the E step of the algorithm (particle filter) given the historical values of the network parameters (red time, cycle time, saturation number of vehicles) for each link of the network. For the prediction at time interval $t+q$, no data is available for time intervals posterior to time interval t. The filter is run forward, without weighting the particles (since future data is not observed). The prediction process is a particular case of missing data in which the data is missing for all the links and all the time intervals after t.

The prediction of the most-likely state at time $t+q$ and the historic values of the link parameters allow for the computation of the travel time distributions of each link of the network at time interval $t+q$. The travel time distributions provide various information including a mean travel time, a variance, confidence intervals and so on.

The assessment of estimation and prediction capabilities is performed on Tuesday, February 22nd 2010 (Tuesday following the training period) from 3pm to 6pm. Figure 6.5 reports the error metrics for prediction steps ranging from 1 time step (a priori estimation) to 4 time steps (1 hour). The results are compared with the estimates of the baseline model. For the baseline model, the real-time prediction is computed as the historical average of the pace for each link during the time interval of interest. This means that the prediction for Tuesday, February 22 at 3pm is the average pace observed at 3pm from the training data set (the three previous Tuesdays). Therefore, the estimates of the baseline model do not depend on the horizon of prediction.

For the *a priori* estimation (prediction at one time step), the error metrics of both the traffic model and the baseline model slightly increase compared to the results presented in Section 6.5. This increase in the error metrics accounts for the differences in traffic conditions on a new day and the loss of accuracy between the *a posteriori* and the *a priori* estimates. The improvement of the traffic model is higher and shows the capabilities of the model to adapt to slightly different traffic conditions and perform short-term prediction.

As the number of prediction steps increases, the estimation error of the traffic model increases. The modeling of the traffic dynamics ensures a certain regularity in the traffic estimates, and the prediction capabilities of the model remain accurate and represent a significant improvement to the baseline model. The Root Mean Squared Error shows the greatest improvement, which indicates that the traffic model has fewer estimates that differ in a significant way from the true values of the travel times than the baseline model does.

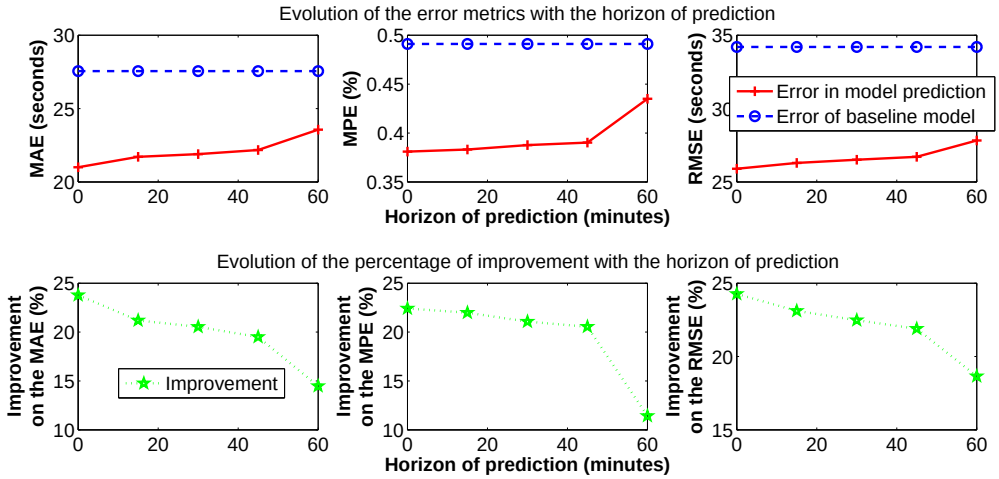


Figure 6.5: Error metrics assessing the prediction capabilities of the *Dynamic Bayesian Network* modeling the dynamics of traffic flow from horizontal queuing theory. The results show accurate prediction capabilities of the traffic model up to 45 minutes ahead. The baseline estimates are computed using historical estimates of the mean travel time, computed during the training. The baseline model does not provide prediction capabilities based on the current state of traffic and thus produces the same estimates for all horizons of prediction.

Field test experiment

The data collected during the *field test experiments* in San Francisco (see Section 5.5 for a description of the dataset) provides another validation of the capabilities of the model. Route travel times are computed from the GPS traces on four different routes of the network (Figure 5.6). The north and south end of the routes are respectively California St and Grove St. The four routes consist of Van Ness Ave. north bound, Van Ness Ave. south bound, Franklin St. and Gough St.

In order to assess the validity of the model, the GPS traces collected during the field experiment are down-sampled to mimic the kind of data generally available in real-time. The model runs over this sparsely sampled data. The validation compares the estimates of the route travel times with the actual route travel times of the drivers. The comparison of the model estimates and the ground truth route travel times are presented in Figure 6.6. This data highlights the variability of travel times experienced by vehicles. The travel time estimates closely follow the trend of traffic dynamics. The RMSE of the traffic model on the route travel times of the drivers is 74.42 seconds, the MAE is 63.62 seconds and the MPE is 33.24%. The travel times on the routes are significantly higher than the travel times used for validation in Section 6.5, hence higher values of the RMSE and MAE. In the computation of the MPE, each estimation error is normalized with the travel time on the path. The MPE is better on longer stretches, as the relative variability of travel times is comparatively smaller than on shorter stretches.

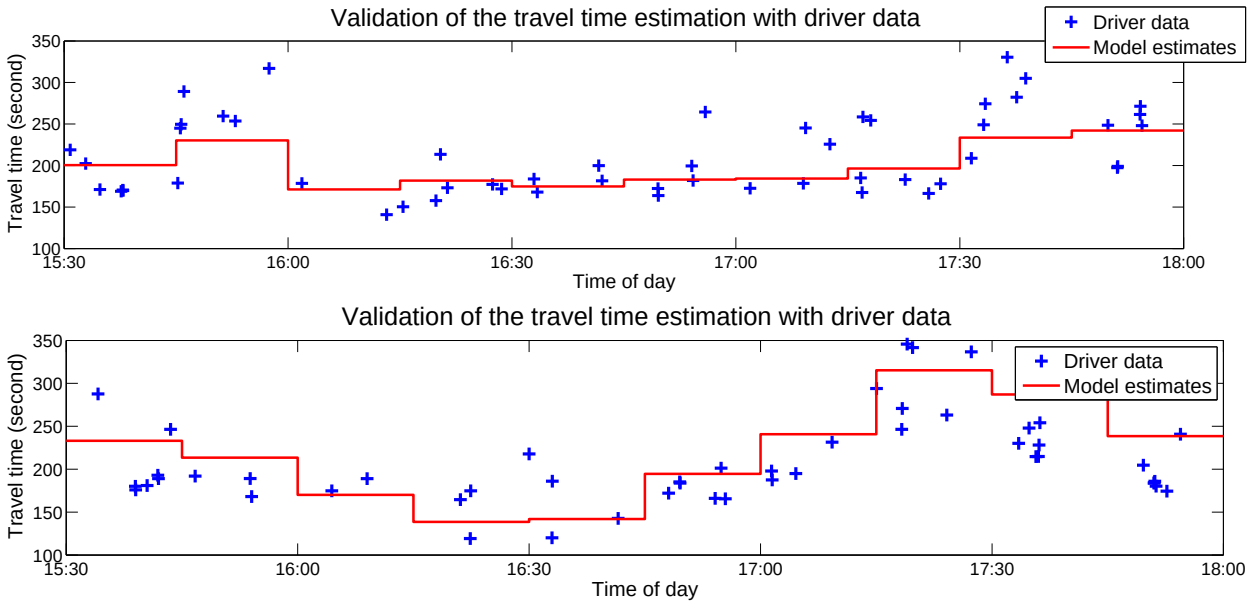


Figure 6.6: Comparison of the model estimates with the ground truth route travel times collected during a field test experiment in San Francisco, CA. The two figures compare the model estimates with the route travel times on Van Ness Avenue (North and South bound). The red curve represent the average travel time estimate of the traffic model. The blue crosses represent the driver data collected during the field test experiment.

6.6 Conclusion and discussion

This chapter presents a statistical model based on the dynamics of arterial traffic flow. The results indicate that the model provides a substantial improvement over a "simple" baseline approach. Besides the improvement of the mean travel time estimation, the model possesses several advantages over the comparison model:

- It improves the estimation of mean link travel times compared to a baseline model.
- It estimates the *probability distribution* of travel times (rather than only the mean) between any two location on the network.
- It *learns* parameters from the physical model of traffic (such as fundamental diagram and signal parameters) and also learns turn movement probabilities within the arterial network.
- It leverages historical data to estimate current traffic conditions from streaming data. The model provides estimates throughout the network even *where little or no real-time data is received*. This is due to the model's ability to accurately track flows through the network as well as the relative recurrence of arterial traffic dynamics. This is the main

improvement compared to the model of Chapter 5. However, it comes with higher computation costs which limits the size of the network to medium-sized networks.

The model can be adapted depending on the sparsity, the noise and the amount of available data. For example, the model could take into account the fact that delays are dependent upon the turn movement through the intersection by modeling each lane as a different queue. The model could also take into account the correlation between travel times on neighboring links to account for light synchronization, as done in [188, 121]. In a statistical model, one needs to find a compromise in the level of detail and number of parameters chosen for the model depending on the type and the amount of data available. Indeed, a more precise model with numerous parameters is able to fit the training data more accurately and explain more details in the dynamics of the model. However, such a model is more likely to *over-fit* the data when the amount of training data is not sufficient to learn all the parameters. Over-fitting the training data decreases the performance on testing data and thus the capabilities of real time estimation and short-term prediction of the model. The chapter considers sparsely sampled probe vehicles (vehicles report their location every minute). The level of granularity of the data does not allow for a fine recovery of the dynamics. Instead, the model focuses on estimating of trends of traffic (statistical estimation every fifteen minutes) rather than fluctuations (variations of queue length and travel time within a traffic cycle). It also motivated the decision not to estimate signal phases and lane by lane queue length (even though it can be a natural extension of the model).

The mathematical abstraction is based on traffic modeling assumptions that can limit the applicability of the model. In particular, the model assumes uniform arrivals on each link of the network. On controlled arterials, where signal synchronization is important, this hypothesis does not hold and the model does not capture travel time distributions as accurately. However, the statistical formulation of the problem provides more flexibility and robustness: it enables us to integrate small discrepancies between the mathematical model and the physical world as well as noise in the measurements. It is possible to account for platoon arrivals and generalize the traffic travel time distributions [15]. These derivations capture more accurately the travel time distributions on controlled arterials but come with the cost of more complicated analytical expressions and higher risk of over-fitting because of the additional parameters introduced in the modeling.

This chapter presents the fundamental concepts needed for performing large-scale estimation of arterial traffic conditions using only low penetration rate GPS probe data. For the next decade, only a small number of municipalities will have the financial resources to equip their entire arterial network with dedicated monitoring infrastructure. At the same time, the market of probe data remains too fragmented to this day to be used in high penetration rate models, forcing traffic engineers to design traffic information systems capable of handling sparse data.

Chapter 7

Data-driven model of congestion dynamics

Chapter 6 develops a hybrid approach of traffic flow theory and statistical modeling to estimate and predict traffic conditions in arterial networks using probe data. The analysis of the results shows that some of the assumptions of the model are sometimes too strong and have limitations when compared to the physical reality (in particular the assumption of uniform arrivals). For this reason, it is worth investigating a model which keeps some of the intuition of traffic modeling without making strong assumptions on the dynamics through a more *data-driven* approach. In particular, the data-driven model does not require to model the probability distributions of travel time using horizontal queuing theory. It uses "classical" distributions (such as Gaussian distributions) instead of the distributions derived in Chapter 5. Classical distribution tend to have mathematical properties which reduce the computational complexity of the model. The use of classical distribution also provides more flexibility to the model than the shape imposed by the traffic modeling. Moreover, the algorithm of Chapter 6 requires a travel time decomposition (travel time allocation) algorithm which has important limitations as will be underlined in the present chapter.

Chapters 5 and 6 underlined the importance to have distributions of travel times between arbitrary locations, as sparsely sampled probe vehicles may report their location at any point on the network, not only at the beginning and at the end of links. The travel time distributions derived in Section 5.3 are parameterized by the location of the measurements as they directly take into account the queue formations. For distributions which do not model queue formation, it is important to take into account the spatial heterogeneity of speeds on a link. Indeed, on a given link of the road network, speeds are on average lower close to the downstream intersections because of stops and delays induced by the signal. One possibility is to use a finer discretization of the road network and to learn parameters for each of the discretized segments. This solution has a high risk of overfitting given the current penetration rates of probe vehicles and the low level of details of the information that they send (probe vehicles report their position on average once per minute). The chapter proposes a trade-off between the risk of overfitting induced by fine spatial discretization and the necessity to take

into account the spatial heterogeneity of travel times: a scaling function which scales partial link travel times to link travel times. The scaling is based on the distribution of vehicles along each arterial road segment, illustrating the fact that travel times are on average longer close to downstream intersections because of the presence of traffic signs.

A dynamic Bayesian network represents the spatio-temporal dependencies on the network and provides a flexible framework to learn traffic dynamics from historical data and perform real-time estimation with streaming data.

As for Chapter 6, the chapter specifically investigates the estimation and short term forecast of the *probability distribution function* (pdf) of travel times in the case of noisy, sparse probe data. In particular, the model and the algorithm for traffic estimation are designed to use probe vehicle travel time measurements received at *random locations* and *random times*. Each observation, defined as two consecutive GPS measurements including the travel time between these measurements, has a probability density that depends on (i) the pdf of travel times of the links traversed and (ii) the spatial distribution of vehicles on each traversed link. The key insight is that, on average, vehicles are more likely to experience delay close to intersections because of the presence of traffic signals. According to the model, the pdf of travel times on each link of the network depends on the level of congestion (discrete *congestion state*) of this link. A *Dynamic Bayesian Network* is used to model and learn the dynamics of congestion on the network using a DBN.

The chapter is organized as follows. Section 7.1 presents a graphical model representing the dependencies between the travel time observations and congestion state of each link at each time interval and their spatio-temporal evolution, inspired from the hybrid model of traffic flow theory and statistical modeling of Chapter 6. Section 7.2 formalizes the intuition that vehicles are more likely to experience delays close to intersections. The section discusses how this information can be used to compute the pdf of travel times on any path, between arbitrary locations from the pdf of travel times of the links traversed. Leveraging the modeling of Section 7.1 and the results from Section 7.2, the DBN represents the probabilistic dynamics of traffic congestion and the probabilistic observation model of the congestion states from probe data. An *expectation maximization* (EM) algorithm (Section 7.3) is used to learn the parameters of the DBN. The *expectation step* (E step) is performed with a particle filter and the *maximization step* (M step) involves solving a large convex optimization problem and is solved with an interior point algorithm. After the historical learning of the parameters of the dynamics of the system, the section describes how to estimate the current state of the network and predict the probability of congestion and the pdf of link travel times from the probe data available in real time. Finally, Section 7.4 presents a case study in San Francisco, for which a fleet of 500 probe vehicles provides sparse location measurements [1]. This data is one of the feeds available in the *Mobile Millennium* system [4]. The numerical experiments analyze the learning and estimation capabilities on a subnetwork with more than 800 links.

7.1 Modeling assumptions

Dynamical model

The model represents the main characteristics of traffic dynamics while making assumptions necessary for the tractability of the estimation process. The validity and limitation of the model are further discussed in Section 7.5, as well as possible refinements of the modeling and generalizations. The modeling assumptions are detailed below and compared to the ones of Chapter 6.

1. *Time discretization:* As done in Chapter 6, traffic is modeled as a discrete time dynamical system, with time discretization Δ_t, chosen depending on the data available and the desired temporal scale of the estimation. This work is focused on estimating travel time distributions when measurements are sparse. In the numerical experiments, $\Delta_t = 5$min. It is chosen such that the model estimates trends rather than fluctuations. For $t \in \mathcal{T} = \{0, \dots, (T-1)\}$, time interval t is given by $[t_0 + t\Delta_t, t_0 + (t+1)\Delta_t]$.

2. *Characterization of the state of traffic*: In Chapter 6, the state of traffic on each link was represented by the number of vehicles in the queue. Here, a discrete *random variable* (r.v.) also represents the congestion state, but does not necessarily correspond to the number of vehicles in the queue. The random variable representing the discrete congestion state of link $i \in I$ during time interval t is still denoted $\xi^{i,t}$. Let $s^{i,t} \in \{0, \dots, S-1\}$ denote the realization of the r.v. $\xi^{i,t}$. The chapter details the derivations for a binary representation of traffic states ($S = 2$), characterizing an *undersaturated* and a *congested* state. The derivations are easily generalized to a larger number of discrete states.

3. *Dynamical model:* Transitions between time intervals model information propagation on the road network by taking into account the spatio-temporal dependencies of the state of the links. In Chapter 6, the dynamics was driven by the flow of vehicles at intersections. In the present data-driven model, there is no notion of flows at intersections. However, it is still possible to model the propagation of congestion by considering that the state of a link at a given time interval depends on the state of its neighboring links at the previous time interval. Formally, $\xi^{I,t}$ (with realization $s^{I,t} \in \{0, \dots, S-1\}^{|I|}$) is the state of the network at time interval t. Let π_i represent the set of links adjacent to link i, including link i: $i' \in \pi_i \Leftrightarrow i' = i$ or i' and i have a common intersection. The equation of the dynamics is given by

$$\xi^{i,t} = f_d^i(\xi^{\pi_i, t-1}) + \epsilon_d^i \; \forall i \in I,$$

where ϵ_d^i represents the state noise of the dynamical model for link i. The dynamic equation can also be defined by a set of conditional independence assumptions[1]:

$$\xi^{i,t} \perp\!\!\!\perp \xi^{i',t'} | \; \xi^{\pi_i, t-1} \text{ for } (t', i') \in X(i,t),$$

where $X(i,t) = \{t-1\} \times I \backslash \pi_i \cup \{0, \dots t-2\} \times I$ and $A \backslash B$ denotes the set A without the elements of B. The mathematical formulation expresses that, given the state of the

[1]For sets of random variables A, B and C, $A \perp\!\!\!\perp B|C$ represents the assertion "A is conditionally independent of B given C".

neighbors π_i at $t-1$, the state of link i at t is independent of the state of non-neighboring links at $t-1$ and is independent of the state of all links of the network at time intervals prior to $t-1$.

4. *Observation model:* The system is observed through noisy path travel time measurements. As for Chapter 6, the path travel times are provided by a map-matching and path inference algorithm [120] which reconstructs the path of the vehicle between successive location reports and filters out the GPS noise. The map-matching algorithm provides the family of links $j(k)$ traversed between the k^{th} pair of successive location reports as well as the distances $x_{s,k}$ and $x_{e,k}$ to the downstream intersection of the first and last link traversed. Note that the path of the probe vehicle between consecutive location reports is fully specified by $x_{s,k}$, $x_{e,k}$ and $j(k)$. The travel time between $(x_{s,k}, x_{e,k})$ is a random variable Y_k, with realization $y_k \in \mathbb{R}$. The observation equation is given by

$$Y_k = f_o(\xi^{j(k),t}, x_{s,k}, x_{e,k}) + \epsilon_o^Y(\xi^{j(k),t} x_{s,k}, x_{e,k}),$$

where ϵ_o^Y represents the observation noise, that may depend on the state of the links of the path and the distance traveled on each of these links. The observation noise is modeled as a sum of independent r.v. representing the observation noise on each link of the path. The travel time on a path is then a sum of independent r.v. representing the travel time on each link of the path. The measurements come from a small subset of vehicles traveling on the network and sending their location periodically in real-time. Measurements from the past are stored and accessible in real-time. The *Mobile Millennium* system, developed by UC Berkeley [4] provides such data.

Dynamic Bayesian Network representation

As for Chapter 6, the conditional independencies introduced by the dynamic and observation equations are represented with a DBN [61]. The structure of the model does not change over time. The structure can be fully specified by a *two-slice temporal Bayesian network* (2TBN). It is common to assume that the parameters of the 2TBN do not change, *i.e.*, the model is time-invariant. This amounts to considering *time of days*, during which the parameters of the 2TBN are constant, as done in Chapter 6. The structure of the DBN induced by the assumptions on the dynamic and observation equations is illustrated in Figure 7.1. The model is fully specified by the following conditional distributions:

- The transition probabilities: For each link i, the transition probability represents the conditional probability that $\xi^{i,t}$ has the realization $s^{i,t}$, given the state of its neighbors at the previous time interval $t-1$. The state of a link at time t may depend on the state of its neighbors in any arbitrary way. Given that both the number of states and the number of neighbors are finite, the conditional probability is represented by a matrix A^i. For each row m, $A^i(m,1)$ (resp. $A^i(m,2)$) represents the probability of being congested (resp. undersaturated) given the state m of the neighbors, so that $A^i(m,2) = 1 - A^i(m,1)$. One possible choice for A^i is to consider all the possible

state combinations of the neighbors, as done in [105], but the dimension of A^i grows exponentially with the number of neighbors and the number of parameters to estimate does not reflect the amount of data available. The chapter investigates a more scalable model in which the state of link i at time interval t depends on the total number of undersaturated links amongst neighbors. With this model, there are $|\pi_i|+1$ parameters to estimate for each link i, where $|\pi_i|$ is the cardinality of π_i. A wide variety of functions of the congestion indices of the neighbors can be used to predict the state of the link at the next time interval. Choosing the appropriate function of the congestion indices is called feature selection [94] and is not detailed in this article. The numerical analysis investigates a few other choices for this function.

- The observation conditional probabilities: For each link i and each state s, the observation conditional probability is the pdf of travel times on link i given the state s. Conditioned on the state s, the travel times on each link i are normally distributed, parameterized by a mean $\mu^{i,s}$ and a standard deviation $\sigma^{i,s}$. The normality assumption is not necessary for the derivations in the model but improves the computational efficiency, as discussed in Section 7.3. The k^{th} travel time measurement y_k is specified by the set of traversed links $j(k)$ as well as the distance to the downstream intersection on the first and last links ($x_{s,k}$ and $x_{e,k}$ respectively). Given the state of the traversed links, the travel time on this path is normally distributed and denoted $f(y_k|s^{j(k),t}, x_{s,k}, x^{e,k})$. The mean and the variance are respectively the sum of the mean and of the variance of travel times on the (partial) links of the path. Note that probe vehicles may not report their location at the beginning or at the end of a link. To overcome this difficulty, Section 7.2 develops a model to properly scale the travel time on the fraction of link traversed (partial link).

- The initial state probabilities: For each link i, $c^i(1)$ (resp $c^i(2)$) is the probability that link i is congested (resp. undersaturated) during the first time interval and have $c^i(2) = 1 - c^i(1)$. This notation can be generalized to any number of state S.

The specification of the conditional distributions leads to the following decomposition of the joint probability of the model:

$$p(s, y|\theta) = \prod_{\substack{t\in\mathcal{T}\setminus\{t_0\}\\ i\in\mathcal{I}}} A(\eta^{i,t-1}, s^{i,t}) \prod_{\substack{t\in\mathcal{T}\\ k\in\mathcal{K}(t)}} f(y_k|s^{j(k),t}) \prod_{i\in\mathcal{I}} c^i(s^{i,0}),$$

where $\eta^{i,t-1}$ represents the congestion state of the neighbors of link i at time interval $t-1$.

Modeling partial link travel time through density estimation

Since probe vehicles send their positions at any location on the network, the path can start and end at any location. The first and last links of the corresponding path are not fully traversed by the vehicle (*partial* links). The pdf of travel times on partial links, *i.e* the pdf

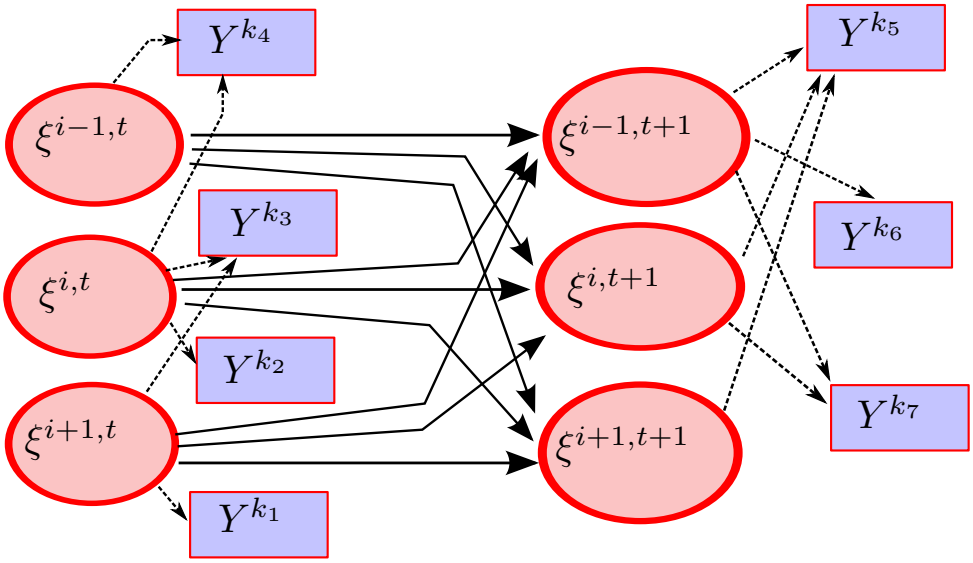


Figure 7.1: Two slice Temporal Bayesian Network (2TBN) representation of the model of arterial traffic dynamics. The circular nodes represent the (hidden) traffic states for each link at each time interval. The square nodes represent travel time observations. There is an edge from the state of link i at time t to the state of link i' at time $t+1$ if i is a neighbor of i' ($i \in \pi_{i'}$). Observation Y_k, received at time t, represents the travel time of a probe vehicle on its path, defined by the set of traversed links $j(k)$ and the distances $x_{s,k}$ and $x_{e,k}$ to the downstream intersections on the first and last links of the path. There is an edge from the state of each link in $j(k)$ to Y_k.

of travel times on link i between any offsets x_1 and x_2 (where x_m, $m = 1, 2$ represents the distance to the downstream intersection) is obtained from the pdf of link travel times with a scaling function. Let $Y^i_{x_1,x_2}$ be the r.v. representing the travel time on *partial* link i between offsets x_1 and x_2 ($x_1 \geq x_2$), then $Y^i_{L^i,0}$ represents the travel time on link i (between offsets L^i, length of link i, and 0). The scaling function $\alpha^i(\cdot,\cdot)$ is defined as $Y^i_{x_1,x_2} = \alpha^i(x_1,x_2)Y^i_{L^i,0}$. The following conditions are imposed to α^i to represent a priori information on the spatial dependency of travel times on a link:

- The travel time on a partial link is a fraction of the link travel time: $\forall(x_1,x_2) \in [0,L]^2$, $\alpha^i(x_1,x_2) \in [0,1]$. If the partial link spans the entire link, the partial travel time has the same distribution as the link travel time: $\alpha^i(L^i,0) = 1$.
- If a partial link is included in another partial link, its travel time should be smaller: $\forall x_1$, $x_2 \mapsto \alpha^i(x_1,x_2)$ is a decreasing function of x_2 and $\forall x_2$, $x_1 \mapsto \alpha^i(x_1,x_2)$ is an increasing function of x_1.
- The probability for a vehicle to experience delay increases as the location gets closer to

the downstream intersection. For the same distance traveled, travel times are longer close to the downstream intersection because of the presence of traffic signals. This physical property is written mathematically as

$$\begin{array}{lll} \forall x_1, & x_2 \mapsto \alpha^i(x_1, x_2) & \text{is a convex function} \\ \forall x_2, & x_1 \mapsto \alpha^i(x_1, x_2) & \text{is a convex function} \end{array}$$

The function defined by $\alpha^i(x_1, x_2) = (x_1 - x_2)/L^i$ satisfies these conditions. However it assumes that the travel time on a partial link is proportional to the distance traveled on the link, but does not take into account the presence of traffic signals. Section 7.2 derives a parametric model for α^i from a hydrodynamic model of traffic flow (Section 5.1) and learns the parameters from the sparse measurements of probe vehicle locations. The function α^i is defined as the *cumulative distribution function* (cdf) of a specific r.v.. For a probe vehicle sampled uniformly in time and reporting its position while traveling on link i, the r.v. represents the position of the vehicle on the link as it reports its location. Its pdf is denoted f_X. Because of the presence of traffic signals, f is a decreasing function of the distance to the downstream intersection (increasing function of the distance from the upstream intersection). The choice $\alpha^i(x_1, x_2) = \int_{x_2}^{x_1} f_X(x)\,dx$, satisfies all the above assumptions.

7.2 Spatial heterogeneity of travel times in signalized networks

Probe vehicles send periodic location measurements, which provide two sources of indirect information about the arterial traffic link parameters. (i) As the location measurements are taken uniformly over time, more densely populated areas *of the link* will have more location measurements. (ii) The time spent between two consecutive location measurements provides information on the speed at which the vehicle drove through the corresponding arterial link(s).

The first source of information provides information on the relation between the travel time on a partial link and the travel time on the entire corresponding link. It is used to derive the function $\alpha^i(\cdot, \cdot)$ introduced in Section 7.1, using the traffic flow theory presented in Section 5.1. The derivations consider a generic link i during a generic time interval. For notational simplicity, this dependency is omitted.

Arterial traffic flow model

The assumptions on arterial traffic dynamics are the same as the ones presented in Section 5.1:

- Lighthill-Whitham-Richards (LWR) model,

- Triangular fundamental diagram, see Figure 2.1 and Equation (2.2),
- Constant characteristics of the traffic light (red time R and cycle time C) and arrival rate q_a, leading to a periodic formation and dissolution of the queues.

As defined in Section 5.1, the model considers two discrete traffic regimes: *undersaturated* and *congested*, depending on the presence (resp. the absence) of a remaining queue when the signal turns red (see Figure 5.1).

Probability distribution of vehicle locations

According to the assumptions, the density at location x is time periodic with period C. The density $d(x)$ at location x is the temporal average of the density $\rho(x,t)$ at location x and time t: $d(x) = \frac{1}{C}\int_0^C \rho(x,t)\,dt$. The density at location x and time t takes one of the three following values, numbered 1 to 3 for convenience: (1) $\rho_1 = \rho_{\max}$, when vehicles are stopped, (2) $\rho_2 = \rho_c$ when vehicles are dissipating from a queue, (3) $\rho_3 = \rho_a$ when vehicles have not yet stopped in the queue. The average density at location x is $d(x) = \sum_{i=1}^{3} \beta_i(x)\rho_i$ where $\beta_i(x)$ represents the fraction of the cycle time C during which density is equal to ρ_i at location x.

When vehicles are sampled uniformly in time, the pdf $f_X^s(x)$ of observing a vehicle at location x is proportional to the average density $d(x)$ at location x, with the proportionality constant given by $Z = \int_0^L d(x)\,dx$ so that $f_X^s(x) = d(x)/Z$. The index $s \in \{u, s\}$ indicates the regime (undersaturated or congested).

Undersaturated regime

Upstream of the maximum queue length, the density is equal to ρ_a throughout the entire cycle. Using the assumption that the FD is triangular and that the arrival density is constant, the average density increases linearly from ρ_a at $x = l_{\max}$ to the value it takes at the intersection, where $x = 0$ (denoted d_0). The function d is defined by three parameters, for example ρ_a, $l_{\max}$ and d_0.

The pdf of vehicle location is proportional to the density f_X^u, with the constraint that $\int_0^L f_X^u(x)\,dx = 1$. It follows that f_X^u is fully specified by two parameters: its constant value for $x \geq l_{\max}$ and the queue length $l_{\max}$.

Congested regime

In the congested regime, the average density is constant upstream of the maximum queue length—equal to ρ_a—and increases linearly until the remaining queue. In the remaining queue, it is constant and equal to $\frac{R}{C}\rho_{\max} + (1 - \frac{R}{C})\rho_c$. The density of vehicles is specified by four parameters, for example, the constant value for $x \geq l_r + l_{\max}$, the constant value for $x \leq l_r$ and the lengths of the queues $l_{\max}$ and l_r.

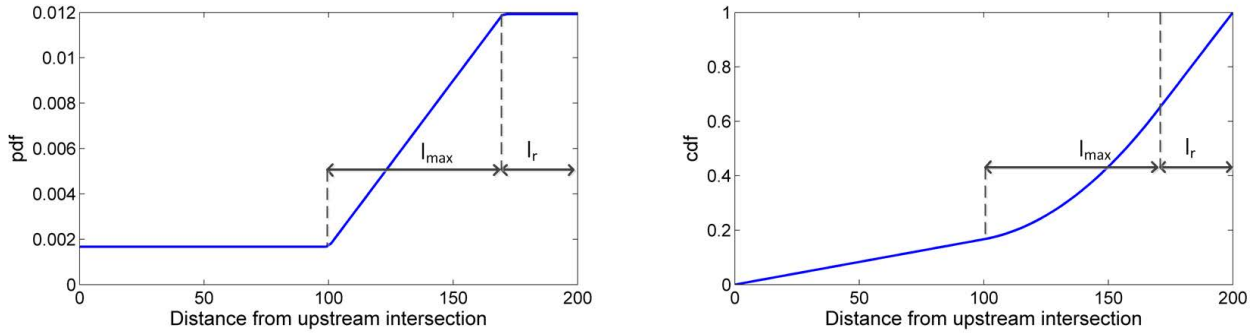


Figure 7.2: Distribution of vehicle location derived from horizontal queuing theory as a function of the distance from the upstream intersection. **Left**: Probability density function. **Right**: Cumulative density function.

Remark 7.1. *The undersaturated regime is a special case of the congested regime, in which the remaining queue length l_r is equal to zero. The congested regime is considered as the general case for the spatial distribution of vehicle location. For this reason, the distribution of vehicle location is denoted f_X, without index referring to the regime in the reminder of the chapter.*

The distribution of vehicle locations is fully determined by three independent parameters: the remaining queue length l_r, the triangular queue length $l_{\max}$ and the normalized arrival density $\tilde{\rho}_a$, which corresponds to the value of the function for $x \geq l_{\max} + l_r$. The pdf is illustrated Figure 7.2 (left: pdf, right: cdf), and reads:

$$\begin{array}{ll} f_X(x) = \tilde{\rho}_a & \text{if } x \geq l_{\max} + l_r \\ f_X(x) = \tilde{\rho}_a + \frac{(l_r + l_{\max}) - x}{l_{\max}} \Delta_{\tilde{\rho}} & \text{if } x \in [l_r,\, l_{\max} + l_r] \\ f_X(x) = \tilde{\rho}_a + \Delta_{\tilde{\rho}} & \text{if } x \leq l_r \end{array} ,$$
$$\text{with } \Delta_{\tilde{\rho}} = \frac{1 - \tilde{\rho}_a L}{l_{\max}/2 + l_r}.$$

The expression of $\Delta_{\tilde{\rho}}$ is obtained by enforcing that $\int_0^L f_X(x)\, dx = 1$.

Density estimation

The parameters of the distribution f_X are estimated by maximizing the likelihood of the set of location observations (denoted $(x_o)_{o \in O}$) provided by large amounts of historical data on each link of the network:

$$\underset{\tilde{\rho}_a, l_r, l_{\max}}{\textbf{maximize}} \sum_{o \in O} \ln(f_X(x_o)) \quad \textbf{s.t.} \begin{cases} 0 \leq \tilde{\rho}_a \leq \frac{1}{L} \\ l_r + l_{\max} \leq L \\ 0 \leq l_r,\, 0 < l_{\max} \end{cases}$$

The constraints come from the physics of the problem. The first constraint ensures that the arrival density is inferior to the average density on the link. The other constraints illustrate that the total queue cannot extend beyond the length of the link and that the triangular queue and the remaining queue are non-negative. The constraints on the queue lengths do not limit the generality of the model. Under spill-over conditions (queue length extending beyond the upstream intersection), the queue is assumed to extend up to the upstream intersection. The rest of the queue is accounted for in the upstream links. Tighter bounds on the parameters can be added to encode the physical insights.

The objective function is not concave in the optimization variables. However, the search space is limited (three bounded variable). The optimization problem is solved using a grid search followed by a local gradient ascent for the B best solutions of the grid-search. The numerical implementation uses 15 points in each dimension for the grid search and $B = 10$. A finer grid did not provide better results.

7.3 Historical learning and real-time inference

As done in Section 6.4, an Expectation-Maximization algorithm is used to learn the parameters of the dynamics. Please refer to Section 6.4 and corresponding references for more background on the Expectation-Maximization algorithm.

Let $\mathcal{Y}$ denote the observable r.v., with realization y (travel time from the probe vehicles) and ξ the latent variables, with realization s (congestion state of the links of the network). Let θ be the set of unknown parameters; *i.e.* $\theta = \{(\mu^{i,s}, \sigma^{i,s}), i \in I, s \in \{0, \dots, S-1\}\} \cup \{A^i, i \in I\}$.

Expectation step

As done in Section 6.4, the Expectation step is performed using a particle filter. The algorithm simulates V particles ($V = 2,000$ in the numerical experiments). Each particle v represents an instantiation of the *time evolution of the traffic state* of the network, *i.e.* a possible succession of traffic states for each link and each time interval. A particle v at time t is represented by a vector of the states of each link and each time interval $(s_v^{i,t'})_{i \in I, t' \in \{0 \dots t\}}$. At t, each particle has a weight ω_v^t proportional to the probability of having this instantiation of the state evolution given the available data up to time t. The particles explore the possible state space and represent the belief state of the DBN.

At time t, the spatio-temporal instantiations $s_v^t = (s_v^{i,t'})_{i \in I, t' \in \{0 \dots t\}}$ of the particles and their associated importance weight ω_v^t form an approximation $p_V(s^{1:t}|y^{1:t}, \theta)$ of the joint probability distribution $p(s^{1:t}|y^{1:t}, \theta)$ of the state of the links up to time t. The following *sufficient statistics* are computed from the particles and their corresponding weights:

- The path sufficient statistic is the joint distribution of the states of the links $j(k)$, on path k, conditioned on the observations received up to time interval t. It is denoted

$p_V(s^{j(k),t}|y^{1:t},\theta)$ and is computed by summing the weights of all the particles for which the links $j(k)$ are in state $s^{j(k),t} \in S^{|j(k)|}$ at t:

$$p_V(s^{j(k),t}|y^{1:t},\theta) = \sum_{v=1}^{V} \omega_v \, \mathbf{1}_{s^{j(k),t}}(s_v^{j(k),t}).$$

- The link sufficient statistic is the probability of the state of link i at time t, conditioned on the state of the neighbors π_i at time interval $t-1$ and the observations received up to t. It is denoted $p_V(s^{i,t}|s^{\pi_i,t-1},y^{1:t},\theta)$ and is computed by summing the weights of all the particles for which link i is in state $s^{i,t} \in S$ at t *and* for which the neighbors of link i are in state $s^{\pi_i,t-1} \in S^{|\pi_i|}$ at $t-1$. To compute the conditional probability, this sum is normalized by the sum of the weights of the particles for which the neighbors of link i are in state $s^{\pi_i,t-1}$.

For each link and each time interval, the number of sufficient statistics to compute is exponential in the number of neighbors of the link. It is possible to overcome this computational cost by assuming that the state of a link at time t depends on the total number of undersaturated neighbors at $t-1$, defined by $\eta^{i,t-1} = \sum_{i' \in \pi_i} s^{i',t-1}$. The number of sufficient statistics to compute for link i is $|\pi_i|+1$ for each time interval, which significantly limits the complexity. Other functions could be used to compactly represent the state of the neighbors. A few other choices are analyzed in the numerical experiments. These functions do not need to be linear nor one-dimensional. The sufficient statistics $p_V(s^{i,t}|\eta^{i,t-1},y,\theta)$ are computed similarly as for $p_V(s^{i,t}|s^{\pi_i,t-1},y,\theta)$: sum the weights of the particles for which link i is in state $s^{i,t}$ at time interval t and for which the sum of the congestion of the neighbors is $\eta^{i,t-1}$ at time interval $t-1$ and normalize as follows:

$$p_V(s^{i,t}|\eta^{i,t-1},y^{1:t},\theta) = \frac{\sum_{v=1}^{V} \omega_v^t \mathbf{1}_{s^{i,t},\eta^{i,t-1}}(s_v^{i,t},\eta_v^{i,t-1})}{Z(\eta^{i,t-1})},$$

The constant $Z(\eta^{i,t-1}) = p_V(\eta^{i,t-1}|y^{1:t},\theta)$ is computed from the particles or, with less computational cost, by summing the joint probabilities $p_V(s^{i,t},\eta^{i,t-1})$ over the possible states of link i at time t.

Using these sufficient statistics, the expected complete log-likelihood $\left\langle l_c(\theta;y,s)\right\rangle_{p_V}$ is given by

$$\sum_{\substack{t\in\mathcal{T}\setminus\{0\}\\ s^{i,t}\in S}} \sum_{\eta^{i,t-1}} p_V(s^{i,t}|\eta^{i,t-1},y^{1:t},\theta)\ln(A(\eta^{i,t-1},s^{i,t})) + \sum_{\substack{t\in\mathcal{T}\\ k\in K(t)}} \sum_{s^{j(k),t}} p_V(s^{j(k),t}|y^{1:t},\theta)\ln f(y_k|s^{j(k),t},\theta).$$

where $s^{j(k),t} \in \{0, \ldots, S-1\}^{|j(k)|}$, $\eta^{i,t-1} \in \{0, \ldots, |\pi_i|\}$, $K(t)$ is the set of path from probe vehicles received during time interval t and $f(y_k|s^{j(k),t}, \theta)$ is the density of probability of the travel time y_k on the links of the path $j(k)$ which are in state $s^{j(k),t}$. The mean and variance of travel times are computed by summing the mean and variance travel times of the (partial) links of the path. Recall that the mean and variance of travel times on *partial* link i are scaled according to the function α^i. In the first sum, $\{0\}$ is removed from the set $\mathcal{T}$ since there is no transition prior to t_0. To compute the sufficient statistics, the filtering step is performed with the particles as follows:

- *Update at t:* Compute the *posterior* distribution using the measurements of time interval t. For each particle, ω_v^t is multiplied by the probability of each measurement given the states $\xi_v^{i,t}$ of the particle. The weights are normalized so that they sum to one.
- *Prediction at $t+1$:* Predict the state distribution for time interval $t+1$ using the transition probabilities. For each link i and each particle v, sample the state $\xi_v^{i,t+1}$ given the states $\xi_v^{\pi_i,t}$ (or any function of the states such as the sum of the congestion states) of its neighbors at time t according to the transition probabilities, *i.e.* the state $s^{i,t+1}$ is chosen with probability $A(s^{i,t+1}|\xi_v^{\pi_i,t})$.

This algorithm is known as a *Sequential Importance Sampling* (SIS) particle filter [9]. As mentioned in Section 6.4, a common problem with the SIS particle filter is the *degeneracy phenomenon* [64, 92]. To reduce the effects of degeneracy, the particles are *resampled* after the update step. The modified algorithm is known as *Sequential Importance Resampling* (SIR) or *Sampling Importance Resampling*.

Maximization step: update of the model parameters

The M step maximizes the expected complete log-likelihood with respect to θ, representing the parameters of both the dynamics (transition probability matrices A^i, $i \in I$) and the observations (parameters of the travel time distributions, conditioned on the state of the link). Given the structure of the complete log-likelihood, this optimization can be performed independently for each transition probability matrix A^i and for the parameters of the joint Gaussian distribution. Note that because travel time observations may span several links, the estimation of the travel time distribution couples all the links of the network. This coupling of the network arises because the algorithm no longer performs a travel time decomposition step.

- *The transition probability matrices* are updated by maximizing with respect to the entries of A^i under the constraint that A^i is a stochastic matrix (all the lines have non-negative entries and sum to 1). For the line j representing the transition probability when the neighbors are in state $m \in \{0, \ldots, S-1\}^{|\pi_i|}$, it follows that

$$A^i(m, s) \propto \sum_{t \in \mathcal{T} \setminus \{0\}} p_V(s^{i,t} = s | s^{\pi_i, t-1} = m, y^{1:t}, \theta),$$

where the proportionality constant is computed for all m such that $\sum_s A^i(m,s) = 1$. A similar expression is obtained if the transitions depend on any functions of the states, such as the number of undersaturated neighbors.

- Given the discrete state of link i at time interval t, the travel time on link i, $Y^{i,t}$, is normally distributed. Remember that the pdf of a partial travel time is computed from the pdf of a link travel time using the scaling function $\alpha^i(\cdot,\cdot)$ presented in Section 7.2, even though the dependency does not appear explicitly for notational simplicity. The travel times are independent r.v.: given the state s of the network at time t, $Y^{I,t}$ is a multivariate Gaussian variable with mean $\mu^s = (\mu^{i,s^i}\, i \in I)$ and covariance $\Sigma^s = \text{diag}((\sigma^{i,s^i})^2 : i \in I)$, where s^i is the i^{th} coordinate of s and represents the state of link i. The M-step updates the mean $\mu = (\mu^{i,s}\, i \in I,\, s \in \{0,\dots,S-1\})$. Let Σ be defined by $\Sigma = \text{diag}((\sigma^{i,s})^2 : i \in I,\, s \in \{0,\dots,S-1\})$. It is the solution of the following optimization problem:

$$\underset{\mu \in \mathbb{R}^{|S|\times|I|}}{\textbf{minimize}} \sum_{\substack{t\in\mathcal{T}\\ k\in K(t)}} \sum_{s^{j(k),t}} p_V(s^{j(k),t}|y,\theta)\cdot(y_k - \mu^{s^{j(k),t}})^T \left(\Sigma^{s^{j(k),t}}\right)^{-1} (y_k - \mu^{s^{j(k),t}})$$

Given that $\Sigma^{s^{j(k),t}}$ is positive definite for all k, the objective function is convex in μ. However the objective function is not jointly convex in μ and Σ and the optimization is performed in the variable μ. The variances are estimated once at the beginning of the algorithm using a Gaussian mixture with two components. The number of variables grows linearly with the number of links. Constraints on the values of the parameter may limit the feasible set to physically relevant values. The optimization is solved using an interior point algorithm [29], which can be replaced by distributed first order algorithms if required by the size of the problem.

7.4 Experiments

The model formalizes an intuitive representation of the propagation of congestion throughout the network. The chapter proposes a learning algorithm of the dynamics of traffic on a network and a real-time estimation framework in a similar fashion as done in Chapter 6. The main difference is the use of Gaussian distributions for the probability distribution of travel times (conditioned on the traffic state) and the use of binary (although higher number of discrete states would be possible) states to describe the congestion level of each link of the network. The framework also makes it possible to by-pass the travel time allocation step.

The validation of the *density model* (Section 7.2) is detailed extensively in [112, 105] and summarized below. This section validates the learning and estimation capabilities of the DBN presented in this chapter. Cross-validation is used to test the estimation and prediction accuracy of the model for different time horizons. The results are compared to a

Algorithm 3 Maximum likelihood estimation of the parameters of the dynamic and observation models.

Initialize the parameters: $(\mu^{i,s}, \sigma^{i,s})_{i,s}$, $(A^i)_i$.
EM-algorithm for parameter estimation in DBN
while The algorithm has not converged **do**
E Step
Initialize the E Step: Simulate samples with weight $\omega_v = 1/V$ representing the state of the network at the initial time given the initial state probabilities.
for $t \in \mathcal{T}$ **do**
Update: For each travel time observation, multiply the weight of each particle with the probability of the observation given the state of the particle: $\omega_v \leftarrow \omega_v \prod_k f_{Y_k}(y_k|\xi_v^{j(k),t})$
Normalize: divide the weight of each particle by the sum of the weights.
Re-sample the particles to avoid degeneracy (Figure 6.3 and details in [138, 9]).
Predict: For each link i and each particle v sample the state at time interval $t+1$ using the transition probabilities A^i.
end for
M Step
Update the transition probabilities A^i, $i \in I$.
Update the parameters of the observation model.
end while

baseline model. The section also investigates how the use of the density model improves the results and how the imposed structure on the dynamics (*i.e.* the dependency between the state of a link at a given time interval and the state of its neighbors at the previous time interval) influences the estimation capabilities. Finally, the section analyzes the quality of the pdf of travel times learned by the model. The estimation of the travel time distribution (rather than mean values only) is crucial in arterial networks to accurately describe the variability of travel times.

Validation of the density model

The data is collected by one of the feeds of the *Mobile Millennium* system: a fleet of 500 vehicles reporting their location every minute in San Francisco, CA. The study focuses on a subnetwork of the San Francisco (Figure 6.4) as in Chapter 6). A *historical interval* is a tuple consisting of a day of the week, a start time, and an end time. For each historical interval and each link, the locations reported by the vehicles are aggregated and used to learn the parameters of the density model. The numerical results analyze data collected on 15 minutes intervals on Tuesdays from 4 to 8 pm, *i.e.* (Tuesday, 4pm, 4:15pm), ..., (Tuesday, 7:45pm, 8pm).

Table 7.1: Percentage of positive K-S tests for different values of threshold to accept the hypothesis H_0 and the two hypothesis (density model or uniform distribution).

Distribution of vehicles	α			Mean p-value
	0.1	0.05	0.01	
Density model	0.75	0.80	0.89	0.35
Uniform	0.46	0.55	0.67	0.15

For each link and each historical interval, the *Kolmogorov-Smirnov* (K-S) statistics tests if the locations of the probe vehicles are distributed according to the density model [162]. The K-S test is a standard non-parametric test to state whether samples are distributed according to a hypothetical distribution. The test is used to accept (or reject) the null hypothesis H_0: "*The measurements of probe vehicles are distributed according to the density model*".

The experiments aim at validating the capability of the density model to properly scale travel time on portions of arterial links. In particular, the density model takes into account the non-uniformity of measurements along the link as vehicles are more likely to experience delay close to the downstream intersection. To illustrate this reasoning, the K-S test is also performed with the following null hypothesis: "Measurements are uniformly distributed along the link". The results of the test on both hypotheses in Table 7.1.

The results indicate that for a majority of arterial links, the average location of vehicles is a random variable that follows the density model. The spatial distribution of vehicle location is better represented by the density model than by a uniform distribution. A graphical representation of the data provides valuable qualitative information. In particular, it is informative to represent the cumulative locations reported by the vehicles for different links of the network[2]. The Figure also displays the empirical (Kaplan-Meier) cdf [132] and the proposed cdf. Figure 7.3 represents the cumulative distributions obtained for two links of the network during the first historical interval. The first link shows a good qualitative fit. However, the p-value is only 0.091. The map discretization does not take into account the width of intersections and may be the reason why no measurements are received on the last 15 meters of the link. The second link has an average p-value. In both cases, the data follows the sharp increase in the density of measurements close to the downstream intersection, as predicted by the model because of the presence of a traffic signal. The model also provides an estimate of the historical queue length on each link of the network which can be used for planning and network congestion analysis.

The analysis of the links with low p-values is also informative and valuable. Figure 7.4 presents the result for a link with p-value equal to 6.8×10^{-4}. The model predicts that sharp increases in the density of measurements occur upstream of traffic signals. The map

[2]The cumulative locations are computed as follows: (1) order the locations reported by the probe vehicles, (2) plot the points $(x_i, i/N)$ for $i = 1 \dots N$, where N is the number of locations collected for the link and historic interval and x_i is the i^{th} location on the link (in meters from the upstream intersection).

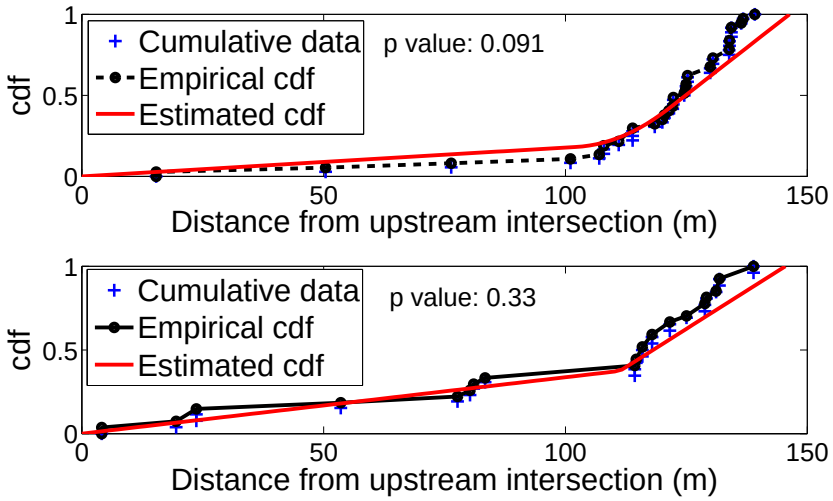


Figure 7.3: Comparison of the empirical and the learned cumulative distribution of vehicle locations. The empirical distribution is computed from points which were set aside during the training. **(Top):** Link with p-value equal to 0.09. The model predicts a sharp increase in the density of measurements towards the downstream extremity of the link but no measurements are received on the last 15 meters of the link. The digital map does not model the width of the road or the intersection, which might explain the absence of measurements on the last 15 meters. **(Bottom):** Link with p-value equal to 0.33. The model learns the characteristics of the distribution of vehicle locations. The results also provide an estimate of the historical queue length (around 30 meters) which provides information on the average congestion of the link.

database, provided by NAVTEQ, contains attributes of the transportation network, such as road characteristics, presence of traffic lights, and so on. On this link, the cumulative distribution of vehicle location exhibits two important increases, whereas only one signal was present in the map database.

The analysis of the location of the link in *Google Street View* confirms that there is a signal which is not indicated in the original database. With the corrected information, the p-value of the K-S test for the updated proposed distribution is 0.29. This approach can be generalized and developed to automate the detection of traffic signals from probe data [110], to develop and correct *Geographic Information Systems* (GIS). Other sources of poor fitting are due to specific behaviors of the taxi, such as waiting in front of major hotels, which can be filtered, when considering successive locations of a taxi.

Validation of the dynamic Bayesian modeling

As probe vehicles report their location periodically in time, the duration between two successive location reports x_s and x_e represents an observation of the travel time of the vehicle on

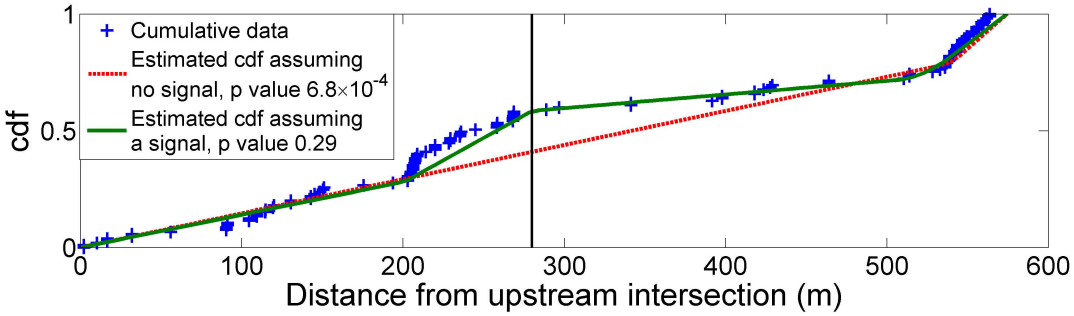


Figure 7.4: Detection of signal locations using the spatial distribution of vehicles. The figure illustrates an example of very low p-value for a link of the network. A careful analysis of the results showed that a signal was missing in the database, explaining the poor fit of the model.

its path from x_s to x_e, *i.e.* the realizations y_k of the random variables Y_k. A map-matching and path-inference algorithm [120] which combines models of GPS emissions and of drivers' behavior into a conditional random field filters out GPS noise, maps the GPS measurements to the road network and reconstructs the most likely set of links traversed by the vehicle.

This case study focuses on learning the model parameters on Tuesdays from 4pm to 8pm in the subnetwork of San Francisco depicted in Figure 6.4. The time discretization Δ_t is chosen as $\Delta_t = 5$min. As mentioned in Section 7.1, the travel times on the links of the network are considered as independent Gaussian variables, conditioned on the state of the links of the network. The choice of a Gaussian distribution may restrict the flexibility of the model to capture unique traffic characteristics, but it is more computationally efficient in practice. In particular, the model relies on travel times from probe vehicles which typically traverse several links between successive observations. The travel time on the path is a sum of independent r.v. and its pdf is computed as the convolution of the pdf of the link travel times on the path. If the link travel times are normally distributed, the computation of the convolution is straightforward whereas it is requires numerical algorithms otherwise. The density model is used to compute the pdf of partial link travel times from the pdf of link travel times.

In traffic estimation (or prediction), access to ground truth data is rare as it requires the monitoring of each vehicle on the entire network for the duration of the estimation. Instead, cross-validation [142] is commonly used in the machine learning community to assess how the results of a statistical model generalize to an independent data set, not used to develop the model but assumed to follow the same model. For each time interval, the available data (travel time measurements of the probe vehicles) is randomly partitioned into complementary subsets. One of the subsets (training set) learns the parameter of the model. The other subset (validation or testing set) validates the performance of the model. The training set constitutes 70% of the available data, the remaining 30% is used for validation.

Estimation and prediction errors

The travel times predicted by the model are compared to the travel times reported by the probe vehicles using l_1 distance. Given a set of observations $y_k,\ k \in K(t)$ received at time interval t and corresponding estimates (predictions) $\hat{y}_k$, the average l_p error e_p is given by

$$e_p = \left(\frac{\sum_{k=1}^{K(t)} |y_k - \hat{y}_k|^p}{|\mathcal{Y}|} \right)^{1/p}.$$

The error is typically normalized by the average travel time measurements $\overline{y}$ (time between successive measurements) and is then denoted percentage error $\tilde{e}_p = e_p/\overline{y}$. Without a reference, these values are hard to interpret: travel times on arterial networks have a high variance due, in particular, to the presence of traffic signals (see Chapters 5 and 6 and [114, 113]). Under similar traffic conditions, the travel times of vehicles on an arterial link vary significantly depending on the time at which the vehicle entered the link and the corresponding waiting time at the signal. To improve interpretability, the results of the model are compared to a *baseline model*: a time-series model adapted to probe vehicle data.

If probe vehicles sent their travel times between defined positions, time series could be applied to estimate the travel time between these positions. However, no two distinct vehicles report their travel time between the same locations. The baseline model adapts the traditional time series approach to probe vehicle data. Travel times are decomposed onto the links of the path and partial link travel times are scaled onto link travel times. The link travel times are estimated with a moving average algorithm.

The following two aspects of the model are also analyzed:

- *Use of the density model*: the errors of the DBN model with the density model are compared to the errors of the DBN without the density model (scaling of partial link travel times using the fraction of the link traversed)
- *Structure of the DBN*: the results of the DBN are compared to the results of a model, denoted *self only*, with no spatial dependency, In the 2TBN, the edges representing the dynamics of the *self-only* model only connect the same links. To show the generality of the spatial dependencies allowed by the framework, the results are also compared to the results of a model denoted *not self* where edges linking link i from time intervals t to $t+1$ in the 2TBN of Figure 7.1 are removed.

Figure 7.5 compares the results of the proposed model (estimation and 15 minute forecast capabilities) with the baseline model. There is a significant improvement in the percentage of error compared to the baseline model. The prediction accuracy decreases with the horizon of prediction but remains better than the baseline. Note that the baseline model does not have prediction capabilities.

Table 7.2 compares the results of the DBN with or without the density model and validates the use of the density modeling to scale partial travel times and compute the pdf of

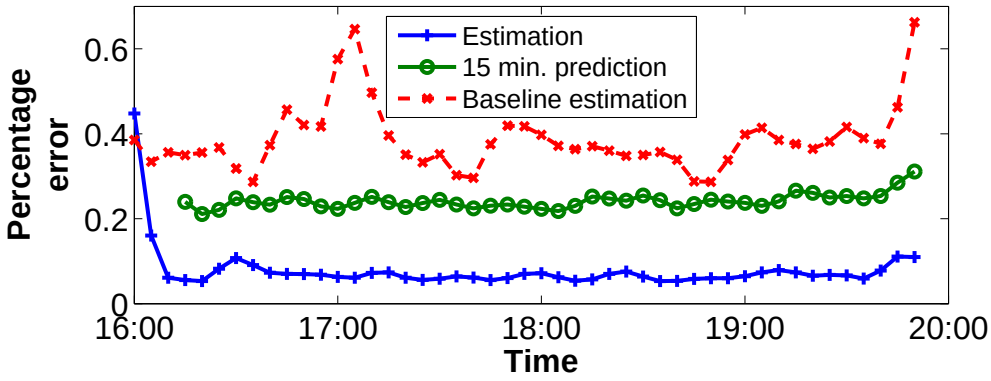


Figure 7.5: Evolution of the estimation and prediction of the percentage l_1 error on the validation dataset.

Table 7.2: Percentage of l_1 error of the model computed on a validation data set to test the estimation and prediction capabilities of the model.

	Percentage of l_1 error				
	with density	without density	not self	self only	baseline
Estimation	0.068	0.072	0.086	0.076	0.385
Prediction 15 min.	0.240	0.243	0.243	0.242	N/A

travel times on partial links. The results also validate the short-term prediction capabilities of the DBN (both with and without the density modeling) and underline the importance of the rich DBN structure, as shown by the better results of the model compared to simpler DBN structures (not self and self only).

Validation of the estimated travel time distributions

The algorithm produces more information than a single mean travel time: (i) it characterizes the pdf of travel times on the network, (ii) it estimates the probability of congestion $p^{i,t}$ of each link i and time interval t and (iii) it provides the parameters of the Gaussian distributions $(\mu^{s,i}, \sigma^{s,i})$. The distribution of travel times on any path $j(k)$ can be sampled and numerically approximated, using Algorithm 4. In the following, the distributions are approximated with 1000 samples. Let ζ_α be defined as

$$\zeta_\alpha = \left\{ y \in \mathbb{R} : \mathcal{P}(y_k \leq y) = \frac{1-\alpha}{2}, \mathcal{P}(y_k \geq y) = \frac{1+\alpha}{2} \right\}.$$

The probability that y_k is in interval ζ_α is α. For a Gaussian distribution $\zeta_{0.68}$ (resp. $\zeta_{0.95}$) is the interval centered around the median of length two (resp. four) standard deviations. If

the estimation of the travel time distribution is exact, the percentage of points in ζ_α is equal to α. The comparison of the percentage of points in ζ_α with α assesses the goodness of fit of the travel time distributions with the testing data (Figure 7.6).

Algorithm 4 Travel time sampling

```
ŷ^k = 0          // Initialize the path travel time sample
for l = 1 : j(k) do
  r = rand();          // Choose the congestion state
  if r < p^{c,l} then
    g = μ^{0,l} + σ^{0,l} randn()
    ŷ^k = ŷ^k + g          // Add the sampled link travel time to the path travel time
  else
    g = μ^{1,l} + σ^{1,l} randn()
    ŷ^k = ŷ^k + g          // Add the sampled link travel time to the path travel time
  end if
end for
```

The evolution of the percentage of points in ζ_α for different values of α over the validation period characterizes the quality of the estimation of the distribution of travel times. The percentage of points in ζ_α varies over time but remains close to its theoretical value (α) as shown in Figure 7.6 (left). The right of Figure 7.6, represents the percentage of points in ζ_α (averaged on the entire validation period) as a function of α. For all values of α, the percentage of points in ζ_α is slightly inferior to α. The difference between the theoretical and results curves is mostly due to small inaccuracies in the estimation of the mean and/or underestimation of the variance of the distribution. Note that if the curve produced by the model (dashed line with circles) was over the theoretical line, it would indicate an overestimation of the variance.

7.5 Conclusion and discussion

As underlined in the previous chapters, sparsely sampled probe vehicles come as a very promising source of data to develop ubiquitous traffic management systems on arterial networks. Chapters 6 and 7 develop models and algorithms that face specific challenges of probe vehicle data. In particular, the models and algorithms address the following issues, that emphasize the novelty of the estimation technique: (i) the location of measurements and quantity of measurements received in an area is unknown prior to receiving the measurements, (ii) the travel time measurements may span multiple links, (iii) the paths may include partial links for which pdf of travel times must be computed.

The algorithm leverages the large amount of data available historically to learn the dynamics of congestion on the network using an EM algorithm. Modeling assumptions on the observation model (independent travel times normally distributed) and the state dynamics

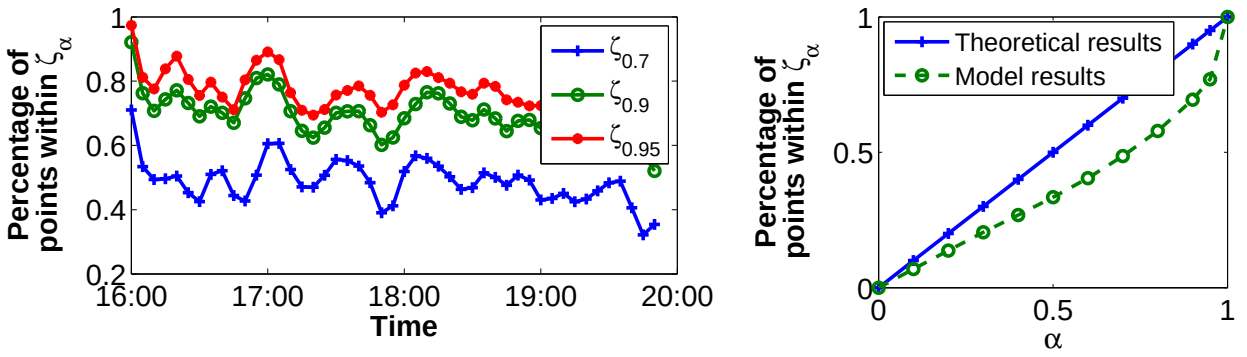


Figure 7.6: Validation of the travel time distributions computed by the model. (**Left**) Evolution of the percentage of points in ζ_α for $\alpha \in \{0.7, 0.9, 0.95\}$. (**Right**) Comparison of the percentage of points contained in ζ_α with the theoretical value.

(evolution depending on the state of the neighbors) maintain the tractability of the algorithm. After learning the physical parameters, the algorithm updates the estimates of traffic conditions from streaming data. The historical training provides robustness to the model when little or no streaming data is available and provides short term prediction capabilities. The algorithm improves significantly the estimation capabilities of a baseline time series algorithm adapted to probe vehicle data. The use of the density modeling to estimate partial link travel times from link travel times also provides an improvement compared to an approach consisting in scaling partial link travel times proportionally to the length of the partial links. Moreover, the algorithm estimates the pdf of travel times on the network rather than mean travel times only, which is a valuable information given the variability of travel times on arterial networks.

The DBN provides the flexibility to adapt to the specifics of the data received and/or the requirements of the estimation by adapting some of the assumptions:

- The *time discretization* Δ_t is chosen as a trade-off between the sparsity of the data and the information that can be reconstructed (fixed to 5 minutes in the numerical results). This time step can be adapted if more precise information is available (or increased if little information is available and traffic conditions are known to have slow dynamics in the region and time period of interest). Chapter 8 investigates a data-driven algorithm to detect changes in the dynamics and potentially improve the choice of temporal discretization.

- The state of traffic of each link is a discrete random variable, which conditions the distribution of travel times. The number of traffic states is not theoretically limited, and may not be the same for all the links of the network. Increasing the number of states implies learning a significantly larger number of parameters to represent the dynamics of traffic on the network: parameters of the travel time distribution for each

state and each link of the network, and parameters of the transition matrix representing the congestion dynamics of each link. As a tradeoff between the information provided by the probe vehicle data and the complexity of the model, the chapter presents the derivations for a binary representation of traffic states. The algorithm can be readily applied with a higher number of states.

- Conditioned on the discrete congestion state, the link travel times are random variables, chosen to be normally distributed in the article. The use of Gaussian random variables offers important model refinement possibilities (without increasing the computational complexity). Chapters 5 and 6 indicate that distributions derived from horizontal queuing theory are able to capture specific features of the underlying physics of queuing networks. However, the modeling assumptions reduce the flexibility of the models. Moreover, the resulting distributions lead to more computationally intensive calculations. The use of Gaussian random variables has the following benefits. First, the pdf of travel times on a path are computed analytically (conditioned on the state of the links on the path), as the sum of independent Gaussian variables. Second, the independence of link travel time, conditioned on the state of the corresponding links, can be interpreted as modeling link travel times on the network as a multivariate Gaussian random variable with diagonal covariance matrix. Allowing extra-diagonal entries models correlation between the travel times on different links.

As mentioned in Chapter 6, it may be desirable to model light synchronization, as done in [188, 121]. To account for light synchronization, the Gaussian distribution of travel times could be replaced by Mixture distributions. Each component of the mixture would represent a *delay pattern* such as "stopping" or "not-stopping", as motivated by the horizontal queuing theory (Chapter 5). A*Markov model* could characterize the probability of a delay pattern on a link given the delay pattern on the previously traversed link. Note however that the models developed in [188, 121] rely on stronger assumption regarding the type of data available (high sampling rate or at least individual link travel times).

Chapter 8

Using sparse modeling to learn spatio-temporal structure

Chapters 6 and 7 developed *Dynamic Bayesian Networks* to model the dynamics of queues and the propagation of congestion in a network. The models make different modeling assumptions corresponding to different level of abstraction from the physics of the queuing system. However, both models have a very similar structure: a dynamic Bayesian network for which the *hidden* state represents a level of congestion, the observed variables are point to point travel times and the parameters characterizing the dynamics are constant for predefined *times of day*. The dynamical models also rely on modeling choices to characterize the spatio-temporal structure of the dynamics (time discretization and edges between hidden variables in the graphical model). For example, both models only consider edges between neighboring links. Similarly the definition of the *time of day* or the time discretization is presented as a trade-off between the amount of data available and the desired level of temporal accuracy. There is no algorithm to automatically detect variations in traffic conditions and trigger an adapted response from the model (transition in the dynamical model, change the parameters corresponding to the time of day, discard obsolete data and so on).

The chapter develops a *novel* general algorithm which has the potential to improve the models of Chapters 6 and 7 by automatically detecting changes, either spatially or temporally. In the context of urban transportation networks, the previous chapters have emphasized the importance of estimating waiting times or travel times. Let the estimate x^n represents the average travel time on each link of the network at time t^n. A l_1-norm penalty on the variations of the estimate $||x_{n+1} - x_n||_1$ encourages the travel time on each link to remain constant unless a significant change in traffic conditions is detected. Similarly, with an appropriate choice of a matrix K, the penalization $||Kx_n||_1$ encourages sparse spatial variations of traffic conditions.

The algorithm is a general data-driven online estimation algorithm which extends existing work in sparse modeling and estimation. The algorithm performs online least-squares estimation of a system. A l_1-norm penalty on the variations of the estimate, or on an affine transformation of the estimate exhibits the spatio-temporal structure of the system. The al-

gorithm analytically computes a homotopy path to update the estimate as new observations become available. It leverages the sparsity structure of the solution to perform computationally efficient and numerically robust estimation. The chapter extends the results of [84] with the following contributions: (i) the algorithm updates the solution as a new batch of p observations is received (previous work only considered updates with one measurement at a time), (ii) the online algorithm solves the LASSO when an linear transformation of the estimate is sparse and (iii) the online algorithm solves the LASSO when the l_1 penalization is between the estimate and a reference value, which can be updated at each estimation step (to study sparse variations for example).

The chapter is organized as follows. Section 8.1 reviews existing work in sparse modeling and introduces the LASSO problem (least-square estimation with l_1-norm penalization on the estimate). Section 8.2 reviews the optimality conditions of the LASSO algorithm and introduces an existing homotopy algorithm [84] to solve the LASSO problem recursively. Section 8.3 presents a homotopy algorithm to update the solution of the LASSO to add (or remove) p observations. In Section 8.4, the algorithm is adapted to produce estimation with the l_1 penalty imposed between the estimate and a reference point which can vary after each estimation. Section 8.5 illustrates the potential of the algorithm to detect spatial or temporal changes in traffic conditions on an arterial network in San Francisco, CA.

8.1 Introduction and related work

Least-squares regression with l_1-norm regularization is known as the LASSO algorithm [207]. It has generated significant interest in the statistics [207, 63], signal processing [19, 36, 81] and machine learning [94, 173] communities, in particular for estimation problems. Adding a l_1-penalty usually leads to sparse solutions, which is a desirable property used to achieve model selection, data compression, or to obtain interpretable results.

The LASSO can be solved using interior-point methods [136], iterative thresholding algorithms [60, 80], feature-sign search [151], bound optimization methods [77], incremental methods [24] or gradient projection algorithms [78]. Homotopy algorithms compute the regularization path [177, 66]. They are particularly efficient when the solution is very sparse [65, 158]. Homotopy algorithms are also powerful to compute online updates [193, 84] when the training examples are obtained sequentially (one at a time). This method is particularly efficient when the support of the LASSO solutions at the particular penalty parameter is similar.

The chapter extends the results of [84] with the following contributions: (i) the algorithm updates the solution as a new batch of p observations is received (previous work only considered updates with one measurement at a time), (ii) the online algorithm solves the LASSO when a linear transformation of the estimate is sparse and (iii) the algorithm solves the LASSO with a l_1 penalty on the difference between the estimate and a reference point, which may change over time. This last property allows to perform estimation in dynamical

system with estimates which exhibit few "jumps" over time. In this case, the penalty is between successive estimate and the reference is updated at each estimation.

At estimation step n, a set I_n of training examples or observations $(y_i, a_i) \in \mathbb{R}\times\mathbb{R}^m$, $i \in I_n$ is available. The chapter presents how to fit a linear model to estimate the response y_i as a function of $x \in \mathbb{R}^m$. A linear function of the solution, $K_1 x$, with $K_1 \in \mathbb{R}^{k\times m}$, is expected to be sparse. The matrix K_1 represents inherent structure of the problem or trend filtering [14, 135]. To achieve this property, an l_1 penalty on $K_1 x$ is added to the least-square estimation problem. The resulting optimization problem is given by:

$$\textbf{minimize}_{x\in\mathbb{R}^m} \frac{1}{2}\sum_{i\in I_n}(a_i^T x - y_i)^2 + \mu_n||K_1 x||_1 \tag{8.1}$$

Other applications may be interested in sparse changes between the state vector and a reference vector $\overline{x}^n$. To achieve this property, an l_1 penalty on the difference between the state vector x and the reference $\overline{x}$ is added to the least-square estimation problem. The estimation problem of x^n is defined as:

$$\textbf{minimize}_{x\in\mathbb{R}^m} \frac{1}{2}\sum_{i\in I_n}(a_i^T x - y_i)^2 + \mu_n||x - \bar{x}^n||_1. \tag{8.2}$$

The reference $\bar{x}^n$ may change after each estimation. In particular, the model can encourage sparse temporal variations to regularize the estimates when measurements are noisy and the dynamics of the system is slow compared to the sampling rate. This property is achieved by choosing $\bar{x}^n = x^{n-1}$.

In applications, it is useful to add additional regularization to the optimization problems (8.1) and (8.2). In particular, for the solution of the least-squares estimation problem to be unique, the matrix $A^T A$ should be non singular, which is not always the case for some applications. Moreover, the regularization term $\mu_n||K_1 x||_1$ or $\mu_n||x - \bar{x}^n||_1$ is on the sparse structure of the estimate but there is no regularization to maintain the state estimates close to an a priori value. As done in the *Elastic Net* [226], the chapter investigates the addition of an l_2 regularization term with weighting parameter λ to Equations (8.1) and (8.2) to improve estimation capabilities. This additional term leverages prior information $\hat{x}$ on the value of the state x (from historical data for example) to improve the estimation capabilities.

The regularization parameter μ_n may depend on the number of measurements $|I_n|$. Example choices are $\mu_n = |I_n|\mu_0$ as in [84] or $\mu_n = \sqrt{|I_n|}\mu_0$ as in [140]. The parameter μ_0 is chosen via cross-validation, as a trade-off between the structure imposed by the regularization, and the fit to the data.

8.2 The LASSO problem

The LASSO problem [207] is defined as follows:

$$\textbf{minimize}_{x\in\mathbb{R}^m} \frac{1}{2}\sum_{i=1}^{n}(a_i^T x - y_i)^2 + \mu_n||x||_1. \tag{8.3}$$

This section summarizes previous work [66, 84] which uses the optimality conditions to solve this problem. The objective function of (8.3) is convex and non-smooth since the l_1-norm is not differentiable when there exists an index i such that the i^{th} element of x (denoted x_i) equals zero. There is a global minimum at x if and only if the subdifferential of the objective function at x contains the 0-vector. The subdifferential of the l_1-norm at x is the following set

$$\partial\|x\|_1 = \left\{ v \in \mathbb{R}^m \; : \; \left\{ \begin{array}{rcll} v_i & = & \text{sgn}(x_i) & \text{if} \quad |x_i| > 0 \\ v_i & \in & [-1,1] & \text{if} \quad x_i = 0 \end{array} \right\} \right\},$$

where $\text{sgn}(\cdot)$ is the sign function. Let $A \in \mathbb{R}^{|I_n| \times m}$ be the matrix whose i^{th} row is equal to a_i^T, and let $y = (y_i)_{i \in I_n}^T$ be the vector of response variables. The optimality conditions for (8.3) are given by

$$A^T(Ax - y) + \mu_n v = 0, \; v \in \partial\|x\|_1.$$

Definition 8.1 (Active set)**.** *The* active set *a is the set of indices representing non-zero elements of x. The matrix A_a is a selection of the columns of A in a. The non-zero coordinates of x are in x_a. The index a_i references the i^{th} coordinate of the active set. Since $v \in \partial\|x\|_1$, $v_{a_i} = sgn(x_{a_i})$.*

Definition 8.2 (Non active set)**.** *The* non active set *na is the set of indices representing zero elements of x. The matrix A_{na} is a selection of the columns of A in na. It follows that x_{na} is the 0-vector. The index na_i references the i^{th} coordinate of the non active set. Since $v \in \partial\|x\|_1$, $v_{na_i} \in [-1, 1]$.*

If the solution is unique, $A_a^T A_a$ is non-singular[1]. The optimality conditions read

$$\begin{array}{rl} x_a & = (A_a^T A_a)^{-1}(A_a^T y - \mu_n v_a) \\ -\mu_n v_{na} & = A_{na}^T(A_a x_a - y) \end{array}.$$

Given the active set and the signs of the coefficients of the solution (and thus the vector v_a), the solution x is computed in closed form. When observations come sequentially, a homotopy algorithm [84] solves the LASSO problem recursively by considering the following problem:

$$x(t,\mu) = \underset{x \in \mathbb{R}^m}{\arg\min} \; \frac{1}{2} \left\| \begin{pmatrix} A \\ t a_{n+1}^T \end{pmatrix} x - \begin{pmatrix} y \\ t y_{n+1} \end{pmatrix} \right\|_2^2 + \mu \|x\|_1 .$$

Adding (resp. removing) a point is equivalent to computing the homotopy path from $t = 0$ to $t = 1$ (resp. from $t = 1$ to $t = 0$). Varying the regularization parameter is equivalent to computing the path from $\mu = \mu_n$ to $\mu = \mu_{n+1}$.

[1]The Elastic Net [226] ensures the uniqueness of the solution without requiring $A^T A$ to be non-singular.

8.3 Recursive lasso with p new observations, l_2 and linear l_1 regularizations

The section studies a least square estimation problem, for which a linear transform of the solution, $K_1 x$ for $K_1 \in \mathbb{R}^{k\times m}$ is sparse. The estimate is updated as p new observations $(y^{\text{new}}, A^{\text{new}}) \in \mathbb{R}^p \times \mathbb{R}^{p\times m}$ become available[2]. The algorithm updates the solution online without having to fully recompute it at each estimation step. Let $\hat{x}$ represent a priori information on the solution, which is used as additional regularization when the matrix A is not full column rank or is ill conditioned (see the *Elastic Net* [226] for details). The matrix K_1 is assumed to be full row rank, which is the case for numerous applications including *total variation regularization*. Each row of K_1 corresponds to an information on the sparsity structure of the solution. Let $K_2 \in \mathbb{R}^{m-k\times m}$ be such that $K = (K_1^T \ K_2^T)^T$ is non singular. For example, K_2 is such that the columns of K_2^T form a basis for the null-space of K_1. The non-singular matrix K defines a change of variable $z = Kx$. It is also convenient to define new data matrices $B = AK^{-1}$, $B_{\text{new}} = A_{\text{new}}K^{-1}$ and $\hat{z} = K\hat{x}$. The section develops an algorithm which updates the solution z of

$$\underset{z\in\mathbb{R}^m}{\textbf{minimize}} \ \frac{1}{2}\left\| \begin{pmatrix} B \\ tB^{\text{new}} \end{pmatrix} z - \begin{pmatrix} y \\ ty^{\text{new}} \end{pmatrix} \right\|_2^2 + \mu||(I_k \ 0_{k\times m-k}) \ z||_1 + \frac{\lambda}{2}||z-\hat{z}||_2^2. \tag{8.4}$$

(i) as t varies to add (or remove) observations and (ii) as μ varies to change the weight of the l_1 regularization. The l_1 penalization is on the first k coordinates of z, denoted *regularized indices*. The last $m-k$ indices are in the active set and are referred to as the *non-regularized indices*.

Add p observations

At $t = 0$, the solution $z(0, \mu_n)$ is known, and so are the active set and the signs of the regularized indices of z. Let v_{a_i} be the sign of $z_{a_i}(0)$ for the *regularized indices* and define $v_{a_i} = 0$ for the *non-regularized indices*. The data matrices with the new observations are indicated with a tilde: $\tilde{B} = (B^T B^{\text{new}T})^T$ and $\tilde{y} = (y^T y^{\text{new}T})^T$. The optimality conditions of (8.4) read

$$\tilde{B}_a^T(\tilde{B}_a z_a(t) - \tilde{y}) + (t^2-1)B_a^{\text{new}^T}(B_a^{\text{new}} z_a(t) - y^{\text{new}}) + \mu_n v_a + \lambda(z_a(t) - \hat{z}_a) = 0, \tag{8.5}$$

$$\tilde{B}_{na}^T(\tilde{B}_a \, z_a(t) - \tilde{y}) + (t^2-1)B_{na}^{\text{new}^T}(B_a^{\text{new}} z_a(t) - y^{\text{new}}) + \mu_n w_{na}(t) - \lambda\hat{z}_{na} = 0. \tag{8.6}$$

where $w_{na}(t)$ is a vector with coordinates in $[-1, 1]$. Notice that, at $t = 0$, $z_a(\cdot)$ and $w_{na}(\cdot)$ are continuous in t. Let t^* to be the largest $t \in [0, 1]$ such that: (i) for all $t \in [0, t^*)$, for all

[2]The solution can also be updated when some of the observations become obsolete.

i in the regularized indices, $\text{sgn}(z_a(t)) = \text{sgn}(z_a(0))$ and (ii) for all $t \in [0, t^*)$, for all i in the non-active set, $|w_{na_i}(t)| < 1$. On this interval, v_{a_i} is the sign of $z_{a_i}(t)$ and Equations (8.5-8.6) are valid.

The matrix $Q = (\tilde{B}_a^T \tilde{B}_a + \lambda I_{|a|})^{-1}$ is computed from its previous value without the p new observations using the Woodbury matrix identity (p rank update). Let $\tilde{z}_a$ and α be defined as $\tilde{z}_a = Q(\tilde{B}_a^T \tilde{y} + \lambda \hat{z}_a - \mu v_a)$ and $\alpha = t^2 - 1$. The singular value decomposition of $B_a^{\text{new}} Q B_a^{\text{new}^T}$ is written $B_a^{\text{new}} Q B_a^{\text{new}^T} = \Gamma^T \Sigma \Gamma$. The *rotated* data is defined by $\overline{B}^{\text{new}} = \Gamma B^{\text{new}}$ and $\overline{y}^{\text{new}} = \Gamma y^{\text{new}}$. Similarly, the rotated error is $\overline{E} = \overline{B}_a^{\text{new}} \tilde{z}_a - \overline{y}^{\text{new}}$ and U is defined as $U = Q \overline{B}_a^{\text{new}^T}$.

Proposition 8.1 (Solution path to add p observations)**.** *For $t \in [0, t^*)$, $z_a(\cdot)$ is continuous in t and given by*

$$z_a(t) = \tilde{z}_a - (t^2 - 1) U \left(I + (t^2 - 1)\Sigma\right)^{-1} \overline{E}. \tag{8.7}$$

Let t^0 be the smallest[3] $t \in [0, 1]$ such that a coordinate of $z_a(t)$ equals zero, t^+ (resp. t^-) the smallest[3] $t \in [0, 1]$ which sets a coordinate of $w_{na}(t)$ to 1 (resp. to -1). The transition point t^* is defined as $t^* = \min(t^0, t^+, t^-)$ and can be computed by solving p-degree polynomial equations on a bounded interval.

Proof. For $t \in [0, t^*)$, it follows from (8.5) and from the Woodbury matrix identity that $(Q^{-1} + \alpha B_a^{\text{new}^T} B_a^{\text{new}})^{-1}$ can be written as $Q - \alpha U (I + \alpha\Sigma)^{-1} \Gamma B_a^{\text{new}} Q$. The expression of $z_a(t)$ reads

$$\begin{aligned}
z_a(t) &= \tilde{z}_a - \alpha U (I + \alpha\Sigma)^{-1} \overline{B}_a^{\text{new}} \tilde{z}_a + \alpha \left(Q - \alpha U (I + \alpha\Sigma)^{-1} \Gamma B_a^{\text{new}} Q\right) B_a^{\text{new}^T} y^{\text{new}} \\
z_a(t) &= \tilde{z}_a - \alpha U (I + \alpha\Sigma)^{-1} \overline{B}_a^{\text{new}} \tilde{z}_a + \alpha \left(U \overline{y}^{\text{new}} - \alpha U (I + \alpha\Sigma)^{-1} \Sigma \overline{y}^{\text{new}}\right) \\
z_a(t) &= \tilde{z}_a - \alpha U (I + \alpha\Sigma)^{-1} \overline{B}_a^{\text{new}} \tilde{z}_a + \alpha U (I + \alpha\Sigma)^{-1} \overline{y}^{\text{new}}
\end{aligned}$$

which proves (8.7). The computation of t^0, t^+ and t^- is given by Lemma 8.1 and 8.2. □

Let $U_{i,j}$ denote the element of U on line i and column j and by U_i the i^{th} line of U, σ_i is the i^{th} singular value of Σ and $\overline{E}_i$ is the i^{th} coordinate of $\overline{E}$.

Lemma 8.1 (Computation of t^0)**.** *Let $t^0_{a_i}$ be the smallest value of $t \in [0, 1]$ which sets the i^{th} coordinate of z_a (in the regularized indices) to zero. It is given by $t^0_{a_i} = \sqrt{\alpha^0_{a_i} + 1}$ where $\alpha^0_{a_i}$ is the smallest real valued solution in the interval [-1,0] of the following p degree polynomial equation in α:*

$$0 = \tilde{z}_{a_i} \prod_{l=1}^{p} (1 + \alpha\sigma_l) - \alpha \sum_{j=1}^{p} U_{i,j} \bar{E}_j \prod_{l \neq j} (1 + \alpha\sigma_l).$$

If the polynomial equation does not have real valued solutions in [-1,0], set $t^0_{a_i} = 1$. It follows that t^0 is the smallest value of $t^0_{a_i}$ in the interval $[0, 1]$.

[3] If no such t exists, set t^0 (resp. t^+ and t^-) to 1.

Proof. Setting the i^{th} coordinate of z_a to zero in (8.7), it follows that

$$\begin{aligned} 0 &= \tilde{z}_{a_i} - \alpha U_i (I + \alpha\Sigma)^{-1}\overline{E} \\ 0 &= \tilde{z}_{a_i} - \alpha \textstyle\sum_{j=1}^{p} \frac{U_{i,j}\bar{E}_j}{1+\alpha\sigma_j} \\ 0 &= \tilde{z}_{a_i} \textstyle\prod_{l=1}^{p}(1+\alpha\sigma_l) - \alpha \sum_{j=1}^{p} U_{i,j}\bar{E}_j \prod_{l\neq j}(1+\alpha\sigma_l). \end{aligned}$$

□

Let c_i denote the i^{th} column of $\tilde{B}_{na}$, d_i denote the i^{th} row of $\overline{B}_{na}^{\text{new}}$ and $d_{i,j}$ denote the element of $\overline{B}_{na}^{\text{new}}$ on the i^{th} row and j^{th} column. Let f_i be the i^{th} element of $\tilde{B}_{na}^T\tilde{e} - \lambda\hat{z}_{na}$ and let $\tilde{e}$ be defined as $\tilde{e} = \tilde{B}_a\tilde{z}_a - \tilde{y}$.

Lemma 8.2 (Computation of t^+ and t^-)**.** *The smallest value of t that sets the i^{th} coordinate of w_{na} to 1 (resp. to -1) is denoted $t^+_{na_i}$ (resp. $t^-_{na_i}$). It is given by $t^+_{na_i} = \sqrt{\alpha^+_{na_i}+1}$ (resp. $t^-_{na_i} = \sqrt{\alpha^-_{na_i}+1}$) where $\alpha^+_{na_i}$ (resp. $\alpha^+_{na_i}$) is the smallest real valued solution in the interval [-1,0] of the p degree polynomial equation in α^+ (resp. in α^-):*

$$(-\mu - f_i)\prod_{l=1}^{p}(1+\alpha^+\sigma_l) = \alpha^+\sum_{j=1}^{p}\overline{E}_j(d_{i,j} - c_i^T\tilde{B}_aU_j)\prod_{l\neq j}(1+\alpha^+\sigma_l),$$
$$(\mu - f_i)\prod_{l=1}^{p}(1+\alpha^-\sigma_l) = \alpha^-\sum_{j=1}^{p}\overline{E}_j(d_{i,j} - c_i^T\tilde{B}_aU_j)\prod_{l\neq j}(1+\alpha^-\sigma_l).$$

If the polynomial equation does not have real valued solutions in [-1,0], set $t^+_{na_i} = 1$ (resp. $t^-_{na_i} = 1$). It follows that t^+ (resp. t^-) is the smallest value of $t^+_{na_i}$ (resp. $t^-_{na_i}$) in the interval $[0,1]$.

Proof. It follows from (8.7) that

$$\begin{aligned} B_a^{\text{new}}z_a(t) - y^{\text{new}} &= B_a^{\text{new}}\tilde{z}_a - \alpha\Gamma^T(I+\alpha\Sigma)^{-1}\overline{E} - y^{\text{new}} \\ &= \Gamma^T\overline{E} - \alpha\Gamma^T\Sigma(I+\alpha\Sigma)^{-1}\overline{E} \\ &= \Gamma^T(I+\alpha\Sigma)^{-1}\overline{E} \end{aligned}$$

The following equality also hold: $\tilde{B}_az_a(t) - \tilde{y} = \tilde{e} - \alpha\tilde{B}_aU(I+\alpha\Sigma)^{-1}\overline{E}$ Equation (8.6) is rewritten as

$$\begin{aligned} 0 &= \tilde{B}_{na}^T(\tilde{e} - \alpha\tilde{B}_aU(I+\alpha\Sigma)^{-1}\overline{E}) + \mu w_{na} - \lambda\hat{z}_{na} + \alpha B_{na}^{\text{new}^T}\Gamma^T(I+\alpha\Sigma)^{-1}\overline{E} \\ -\mu w_{na}(t) &= \tilde{B}_{na}^T\tilde{e} - \lambda\hat{z}_{na} + \alpha(\overline{B}_{na}^{\text{new}^T} - \tilde{B}_{na}^T\tilde{B}_aU)(I+\alpha\Sigma)^{-1}\overline{E} \end{aligned}$$

The values of $t^+_{na_i}$ (resp. $t^-_{na_i}$) are obtained by solving the p degree polynomial equation in α^+ (resp. α^-) on the interval [-1,0]:

$$(-\mu - f_i)\prod_{l=1}^{p}(1+\alpha^+\sigma_l) = \alpha^+\sum_{j=1}^{p}\overline{E}_j(d_{i,j} - c_i^T\tilde{B}_aU_j)\prod_{l\neq j}(1+\alpha^+\sigma_l),$$
$$(\mu - f_i)\prod_{l=1}^{p}(1+\alpha^-\sigma_l) = \alpha^-\sum_{j=1}^{p}\overline{E}_j(d_{i,j} - c_i^T\tilde{B}_aU_j)\prod_{l\neq j}(1+\alpha^-\sigma_l).$$

□

Lemma 8.3 (Update of the active set)**.** *When t reaches a transition point, the active set and signs of the regularized indices are updated as follows: (i) if* $t^* = t^0$*, remove the corresponding coordinate from the active set, (ii) if* $t^* = t^+$ *(resp.* $t^* = t^-$*), add the coordinate to the active set and set its sign to positive (resp. to negative).*

Proof. If $t^* = t^0$, let a_i be such that $z_{a_i}(t^*) = 0$. The subgradient of $||(I_k\ 0_{k\times m-k})z||_1$ with respect to the coordinate a_i is in the interval [-1,1], The coordinate is removed from the active set.

If $t^* = t^+$, let na_i be such that $w_{na_i}(t^*) = 1$. For $t > t^*$, the optimality condition for the coordinate na_i cannot be satisfied with the current active set because w_{na_i} is bounded by 1. If one lets the coordinate na_i of the solution take non-zero values, the optimality condition can be rewritten as $f(z_a(t)) + \beta z_{na_i}(t) = 0$, where $f(z_a(t)) < 0$ and β is a positive term which depends on λ and on the norm of the column na_i of B and B^{new}. This proves that z_{na_i} is positive. Adding the index na_i to the active set provides a solution, thus *the* solution (strict concavity). □

Algorithm 5 updates the solution when t varies from $t = 0$ to $t = 1$. The same algorithm is relevant to remove p observations by finding the transition points as t decreases from 1 to 0.

Update the regularization parameter

The computation of the regularization path is detailed in [66] and in [226] for the Elastic Net. As done in the previous step of the algorithm (add p observations), it is necessary to define the *non-regularized indices* and set $v_{a_i} = 0$ for these indices to solve (8.4). The end of the section details how the algorithms developed in [66] and [226] are adapted to solve (8.4).

At $\mu = \mu_n$, the solution $z(0, \mu_n)$ is known, and so are the active set, non active set and signs of the coordinates of z which are in the active set. The optimality conditions read

$$B_a^T(B_a z_a(\mu) - y) + \mu v_a(\mu) + \lambda(z_a(\mu) - \hat{z}_a) = 0, \tag{8.8}$$

$$B_{na}^T(B_a\, z_a(\mu) - y) + \mu w_{na}(\mu) - \lambda \hat{z}_{na} = 0. \tag{8.9}$$

where $v_a(\mu)$ is the partial derivative of the l_1 norm for the indices in the set a with entries $v_{a_i}(\mu) = \text{sgn}(z_{a_i}(\mu))$ for the regularized indices, $v_{a_i} = 0$ for the non regularized indices and $w_{na}(\mu)$ is a vector with coordinates in $[-1, 1]$. Let Q be defined by $Q = (B_a^T B_a + \lambda I_{|a|})^{-1}$.

Proposition 8.2 (Linear dependence in μ)**.** *There exists a transition point* $\mu^* \in [\mu_n,\ \mu_{n+1}]$ *such that the active set, non active set and signs of the regularized indices of the solution remain constant for* $\mu \in [\mu_n, \mu^*)$*. Let* μ^0 *be the smallest*[4] $\mu \in [\mu_n,\ \mu_{n+1}]$ *such that a coordinate*

[4] If no such μ exists, set μ^0 (resp. μ^+ and μ^-) to μ_{n+1}.

Algorithm 5 Update of the solution to add p observations

Initialize the active set a, non active set na and signs of the regularized indices v_a.
$t = 0$
while $t < 1$ **do**
  Compute t^0, t^+ and t^- as the smallest value of $t^0_{a,i}$, $t^+_{na,i}$ and $t^-_{na,i}$ in $(t, 1]$ (Lemma 8.1-8.2).
  $t = \min(t^0, t^+, t^-)$
  if $t > 1$ **then**
    break;
  else if $t = t^0$ **then**
    Add the corresponding index to na and remove it from a and v_a.
  else if $t = t^+$ **then**
    Add the corresponding index to a and remove it from na, set its sign to positive and add it to v_a.
  else
    Add the corresponding index to a and remove it from na, set its sign to negative and add it to v_a
  end if
  Update the matrix Q to account for the updated active set (rank 1 update).
end while
Compute the solution at $t = 1$.

of $z_a(\mu)$ equals zero, μ^+ (resp. μ^-) the smallest[4] *$\mu \in [\mu_n, \mu_{n+1}]$ which sets a coordinate of $w_{na}(\mu)$ to 1 (resp. to -1). The transition point μ^* is defined as $\mu^* = \min(\mu^0, \mu^+, \mu^-)$. On the interval $[\mu_n, \mu^*)$, v_{a_i} denotes the (constant) sign of $z_{a_i}(\mu)$ for the regularized indices. The estimate $z_a(\mu)$ is affine in μ and given by*

$$z_a(\mu) = Q(B_a^T y + \lambda \hat{z}_a) - \mu Q v_a. \tag{8.10}$$

Proof. From (8.8), write $z_a(\mu)$ as

$$z_a(\mu) = Q(B_a^T y + \lambda \hat{z}_a) - \mu Q v_a(\mu). \tag{8.11}$$

At $\mu = \mu_n$, the solution and thus the value of $v_a(\mu_n)$ are known. Equation (8.11) shows an affine dependency of $z_a(\mu)$ with μ as long as the active set and $v_a(\mu)$ remain constant, *i.e.* as long as the regularized indices of z_a have constant sign. Let μ^0 denote the smallest value of $\mu \in [\mu_n, \mu_{n+1}]$ such that a regularized index of z_a reaches zero in Equation (8.11). If no such μ exists, set $\mu^0 = \mu_{n+1}$. The signs of the regularized indices and thus the value of $v_a(\mu)$ are constant on $[\mu_n, \mu^0]$. Let v_a denote the (constant) value of $v_a(\mu)$ on this interval. Equation (8.10) follows directly.

Using (8.10), rewrite (8.9) as

$$-\mu w_{na}(\mu) = B_{na}^T \Big((B_a Q B_a^T - I_m) y + B_a Q (\lambda \hat{z}_a - \mu v_a) \Big) - \lambda \hat{z}_{na}. \tag{8.12}$$

The expression shows that $w_{na}(\cdot)$ is a continuous function of μ. A coordinate of the non-active set joins the active set as the corresponding coordinate of w_{na} reaches one in absolute value. Let μ^+ (resp. μ^-) denote the smallest value of $\mu \in [\mu_n,\, \mu_{n+1}]$ such that a coordinate of w_{na} reaches 1 (resp. -1). The non-active set is constant on $[\mu_n, \min(\mu^+, \mu^-)]$.

The active set, signs of the regularized indices and non-active sets are constant on the interval $[\mu_n, \mu^*)$ where $\mu^* = \min(\mu^0, \mu^+, \mu^-)$. □

As long as the active set and signs of the regularized indices remain constant, the expression of $z_a(\mu)$ is given by (8.10).

Lemma 8.4 (Expression of μ^0)**.** *Let $\mu^0_{a_i}$ denote the value of μ that sets the i^{th} coordinate of z_a (in the regularized indices) to zero and have $\mu^0_{a_i} = [Q(B_a^T y + \lambda \hat{z}_a)]_i / [Q v_a]_i$, where $[V]_i$ denotes the i^{th} coordinate of generic vector V. The first possible transition point μ^0 is the smallest value of $\mu^0_{a_i}$ in the interval $[\mu_n, \mu_{n+1}]$, or μ_{n+1} if, for all the regularized indices, $\mu^0_{a_i} \notin [\mu_n, \mu_{n+1}]$.*

Proof. The expression is readily derived from (8.10) by setting the i^{th} coordinate of z_a to zero. □

Lemma 8.5 (Expression of μ^+ and μ^-)**.** *The values of μ that set the i^{th} coordinate of w_{na} to 1 and -1 are denoted by $\mu^+_{na_i}$ and $\mu^-_{na_i}$ respectively. They are given by*

$$\mu^+_{na_i} = \frac{\Big[B_{na}^T \big((B_a Q B_a^T - I_n)y\big) + \lambda(B_{na}^T B_a Q \hat{z}_a - \hat{z}_{na})\Big]_i}{-1 + [B_{na}^T B_a Q v_a]_i},$$

$$\mu^-_{na_i} = \frac{\Big[B_{na}^T \big((B_a Q B_a^T - I_m)y\big) + \lambda(B_{na}^T B_a Q \hat{z}_a - \hat{z}_{na})\Big]_i}{1 + [B_{na}^T B_a Q v_a]_i}.$$

The first possible transition points μ^+ (resp. μ^-) is the smallest value of $\mu^+_{na_i}$ (resp. $\mu^-_{na_i}$) in the interval $[\mu_n, \mu_{n+1}]$, or μ_{n+1} if, for all i, $\mu^+_{na_i} \notin [\mu_n, \mu_{n+1}]$ and $\mu^-_{na_i} \notin [\mu_n, \mu_{n+1}]$.

Proof. The expressions are readily derived from (8.12) by setting the i^{th} coordinate of w_{na} to 1 (resp. to -1). □

Leveraging Proposition 8.2 and Lemma 8.4 and 8.5, Algorithm 6 updates the solution z when μ varies from $\mu = \mu_n$ to $\mu = \mu_{n+1}$. Note that the derivations assume that $\mu_n \leq \mu_{n+1}$. The same algorithm is relevant if $\mu_n \geq \mu_{n+1}$ by finding the transition point as the regularization parameter decreases (instead of increases).

Remark 8.1 (Leveraging the sparsity structure)**.** *The matrix Q is efficiently updated when the active or non active set change or when observations are added/removed using low rank updates. The numerical implementation updates the Cholesky factorization of Q which provides better numerical stability to the algorithm than updating Q directly [91].*

Algorithm 6 Update of the solution as μ increases from μ_n to μ_{n+1}

Initialize the active set a, non active set na and sign of the regularized indices v_a.
$\mu = \mu_n$
while $\mu < \mu_{n+1}$ **do**
  Compute μ^0, μ^+ and μ^- as the smallest values of $\mu^0_{a,i}$, $\mu^+_{na,i}$ and $\mu^-_{na,i}$ in $(\mu, \mu_{n+1}]$ (Lemma 8.4 and 8.5).
  $\mu = \min(\mu^0, \mu^+, \mu^-)$
  if $\mu > \mu_{n+1}$ **then**
    break;
  else if $\mu = \mu^0$ **then**
    Add the corresponding index to na and remove it from a and v_a.
  else if $\mu = \mu^+$ **then**
    Add the corresponding index to a and remove it from na, set its sign to positive and add it to v_a.
  else
    Add the corresponding index to a and remove it from na, set its sign to negative and add it to v_a
  end if
  Update the matrix Q to account for the added (or removed) index in the active set (rank 1 update).
end while
Compute the solution at $\mu = \mu_{n+1}$

Remark 8.2 (Complexity)**.** *The complexity of the algorithm depends on the number of transitions and the size of the active set. The theoretical bound on the number of transitions is 3^k, where k is the number of rows of K_1. In practice, it is much smaller because successive estimates are expected to have a similar support. Experience with data suggests that the number of transition is linear in the problem size [190]. A theoretical analysis of the number of transitions is performed in [161].*

8.4 Recursive lasso with varying reference parameter

This section considers the linear regression problem introduced in (8.2). The problem encourages the vector $x^n - \bar{x}^n$ to be sparse. The reference $\bar{x}^n$ may change at each iteration. For example, the choice $\bar{x}^n = x^{n-1}$ leads to sparse variations of the estimate. The estimate is updated when observations are added (or removed), when the l_1 regularization parameter changes or when the reference parameter $\bar{x}^n$ changes. In order to update the solution from previous estimates, the algorithm computes a homotopy regularization path, as done in Section 8.3. After computing the solution x^n to Equation (8.2), p new observations $(y^{\text{new}}, A^{\text{new}}) \in \mathbb{R}^p \times \mathbb{R}^{p \times m}$, a new penalty coefficient μ_{n+1} and a new reference parameter

$\bar{x}^{n+1}$ (*e.g.* $\bar{x}^{n+1} = x^n$)are received[5]. As for Section 8.3, an additional l_2 penalization is added to the objective function of the LASSO to improve the estimation capabilities [226]. The homotopy algorithm is derived by introducing the following optimization problem:

$$\begin{aligned} x(t,u,\mu) = \underset{x\in\mathbb{R}^m}{\arg\min} \quad & \frac{1}{2}\left\| \begin{pmatrix} A \\ tA^{\text{new}} \end{pmatrix} x - \begin{pmatrix} y \\ ty^{\text{new}} \end{pmatrix} \right\|_2^2 \\ & +\mu \left\| x - \left((1-u)\bar{x}^n + u\bar{x}^{n+1}\right) \right\|_1 + \frac{\lambda}{2}||x-\hat{x}||_2^2. \end{aligned} \tag{8.13}$$

The definition of Equation (8.13) leads to $x(0,0,\mu_n) = x^n$ and $x(1,1,\mu_{n+1}) = x^{n+1}$. The section develops an algorithm that computes a path from x^n to x^{n+1} in three steps: (i) vary μ from μ_n to μ_{n+1} to change the weight of the l_1 regularization, (ii) vary t from 0 to 1 to add observations and (iii) vary u from 0 to 1 to update the reference parameter. Note that the different steps of the algorithm (variation of μ, t and u) do not need to be performed in a pre-specified order.

The change of the weight of the l_1 regularization and the variation of t from 0 to 1 are readily adapted from the computations of Section 8.3. The section succinctly presents the required changes for these steps and details the algorithm to update the reference parameter from $\bar{x}^n$ to $\bar{x}^{n+1}$ (increase u from 0 to 1).

Update the regularization parameter and add observations

During the update of the regularization parameter and the addition of observations, the parameter u remains constant. Assume without loss of generality that the variation of u is chosen to be performed last and thus $u = 0$ as the regularization parameter is updated and the observations added. If the variation of u has started before these steps occur, replace $\bar{x}^n$ by $(1-u)\bar{x}^n + u\bar{x}^{n+1}$ in the following derivations.

To leverage the algorithm developed in Section 8.3, it is convenient to introduce the following change of variables: $z = x - \bar{x}^n$, $y_r = y - A\bar{x}^n$, $y_r^{\text{new}} = y - A^{\text{new}}\bar{x}^n$ and $\hat{z} = \hat{x} - \bar{x}^n$. For notation consistency, the matrices A and A^{new} are denoted B and B^{new} respectively (same as for Section 8.3 with K being the identity matrix). With this notation, updating the regularization parameter (vary μ) and adding new observations (vary t) correspond to updating the solution of

$$\underset{z\in\mathbb{R}^m}{\textbf{minimize}}\ \frac{1}{2}\left\| \begin{pmatrix} B \\ tB^{\text{new}} \end{pmatrix} z - \begin{pmatrix} y_r \\ ty_r^{\text{new}} \end{pmatrix} \right\|_2^2 + \mu||I_m z||_1 + \lambda||z-\hat{z}||_2^2, \tag{8.14}$$

as μ varies from μ_n to μ_{n+1} and t from 0 to 1.

[5]Note that not all parameters are required to change at each iteration.

Update the reference parameter

The last step of the algorithm updates the reference parameter from $\bar{x}^n$ to $\bar{x}^{n+1}$. Let $x_r(u)$ be defined by $x_r(u) = x - [(1-u)\bar{x}^n + u\bar{x}^{n+1}]$. It represents the vector which is expected to be sparse because of the l_1-norm penalization. As done in the previous section, assume without loss of generality that the variation of u is chosen to be performed last. At this step of the algorithm, the regularization parameter has been updated and the new observations have been added. In particular, since the observations have been added, the matrix A and the vector y contain the recently added data.

Define $y_r = y - A\bar{x}^n$, $\Delta x = \bar{x}^n - \bar{x}^{n+1}$ and $Q = (A_a^T A_a + \lambda I)^{-1}$. Let c_j denote the vector defined by $c_j = A_j^T y_r + \lambda[\hat{x} - \bar{x}^n]_j$, where j represents the set of indices a or na. With this notation, $x_r(u)$ is the minimizer of the optimization problem

$$\underset{x_r \in \mathbb{R}^m}{\textbf{minimize}} \ \frac{1}{2}||Ax_r - y_r - u\Delta x||_2^2 + \mu_{n+1}||x_r||_1 + \frac{\lambda}{2}||x_r - (\hat{x} - \bar{x}^n) - u\Delta x||_2^2.$$

The optimality conditions read

$$(A_a^T A_a + \lambda I)x_{r,a}(u) - c_a + \mu v_a + u\left(A_a^T(A\Delta x) + \lambda(\Delta x)_a\right) = 0, \tag{8.15}$$

$$A_{na}^T A_a x_{r,a}(u) - c_{na} + \mu w_{na}(u) + u\left(A_{na}^T(A\Delta x) + \lambda(\Delta x)_{na}\right) = 0. \tag{8.16}$$

Proposition 8.3 (Linear dependence in u)**.** *There exists a transition point $u^* \in [0,1]$ such that the active set, non active set and signs of the regularized indices of the solution remain constant for $u \in [0, u^*)$. On this interval, v_{a_i} is the (constant) sign of $x_{r,a_i}(u)$. The estimate $x_{r,a}(u)$ is affine in u and given by $x_{r,a}(u) = \xi + u\chi$, with $\xi = Q(c_a - \mu v_a)$ and $\chi = Q\left(A_a^T(A\Delta x) + \lambda(\Delta x)_a\right)$.*

Proof. From the optimality conditions, it follows that the function $u \mapsto x_{r,a}(u)$ is affine as long as $u \mapsto v_a(u)$ is constant *i.e.* as long as the coordinates of $u \mapsto x_{r,a}(u)$ have constant signs and as long as the active set remains constant. Denote by u^0 the smallest value of $u \in [0,1]$ such that a coordinate of $x_{r,a}(u)$ equals zero in Equation (8.15). The signs of the entries of $x^{r,a}(u)$, and thus the value of $v_a(u)$, are constant on $[0, u^0]$. The (constant) value of $v_a(u)$ on this interval is denoted v_a. The optimality condition given in Equation (8.16) also shows that $u \mapsto w_{na}(u)$ is continuous. A coordinate of the non-active set joins the active set when the corresponding coordinate of $u \mapsto w_{na}(u)$ reaches one in absolute value. Let u^+ (resp. u^-) be the smallest value of $u \in [0,1]$ such that a coordinate of $w_{na}(u)$ equals 1 (resp. -1). The non-active set is constant on $[0, \min(u^+, u^-)]$. The active set and signs of the coordinates in the active set remain constant on the interval $[0, u^*]$ where $u^* = \min(u, u^+, u^-)$. □

Lemma 8.6 (Expression of u^0)**.** *Let $u_{a_i}^0$ be the value of u that sets the i^{th} coordinate of $x_{r,a}(u)$ to zero. It is given by $u_{a_i}^0 = -\xi_i/\chi_i$. The first possible transition point u^0 is the smallest value of $u_{a_i}^0$ in the interval $[0,1]$, or 1 if, for all i, $u_{a_i}^0 \ni [0,1]$.*

Proof. The proof is derived from the expression of $x_{r,a}(u)$ given by Equation (8.15) for $u \in [0, u^*]$. □

Lemma 8.7 (Expression of u^+ and u^-)**.** *Let $u^+_{na_i}$ (resp. $u^+_{na_i}$) be the value of u that sets the i^{th} coordinate of $w_{na}(u)$ to 1 (resp. -1),* i.e. *the value of u for which the i^{th} coordinate of $x_{r,na}$ enters the active set and becomes positive (resp. negative). They are given by:*

$$u^+_{na_i} = -\frac{A^T_{na_i} A_a \xi - c_{na_i} + \mu}{A^T_{na_i}(A_a \chi + A\Delta x - A_c(\Delta x)_c) + \lambda(\Delta x)_{na_i}},$$
$$u^-_{na_i} = -\frac{A^T_{na_i} A_a \xi - c_{na_i} - \mu}{A^T_{na_i}(A_a \chi + A\Delta x - A_c(\Delta x)_c) + \lambda(\Delta x)_{na_i}}.$$

The first possible transition point u^+ (resp. u^-) is the smallest value of $u^+_{na_i}$ (resp. $u^-_{na_i}$) in the interval $[0, 1]$, or 1 if, for all i, $u^+_{na_i} \ni [0, 1]$ (resp. $u^-_{na_i} \ni [0, 1]$).

Proof. The proof is derived from the optimality condition given in Equation (8.16) and the expression of $x_{r,a}(u)$ for $u \in [0, u^*]$. □

A transition occurs for the smallest $u^* \in [0, 1]$ such that one component of $x_{r,na}$ enters the active set or one component of $x_{r,a}$ enters the non-active set. At $u = u^*$, update the active and non active sets and search for the next transition point until $u = 1$ and the update of the reference parameter is completed.

8.5 Numerical results

The potential of the algorithm is illustrated through an application for traffic estimation in a subnetwork of San Francisco, CA (Figure 6.4). For this application, the estimate x^n represents the average travel time on each link of the network at time t^n. As done in Chapters 6 and 7, the numerical results use data provided by a fleet of 500 probe vehicles which report their location every minute, representative of the data available in the *Mobile Millennium* system [4].

The duration between two successive location reports[6] ξ_1 and ξ_2 is an observation of the travel time y_i on the path from ξ_1 to ξ_2. After using the map-matching and path-inference algorithm to reconstruct the path of each vehicle [120], each trajectory (path) is converted in a vector $a_i \in [0, 1]^m$, where m is the number of links in the network. The j^{th} coordinate of a_i, denoted $a_{i,j}$, is the fraction of the link traveled by the probe vehicle. It is computed as the distance traveled on the link divided by the length of the link[7]. In particular, $a_{i,j} = 0$

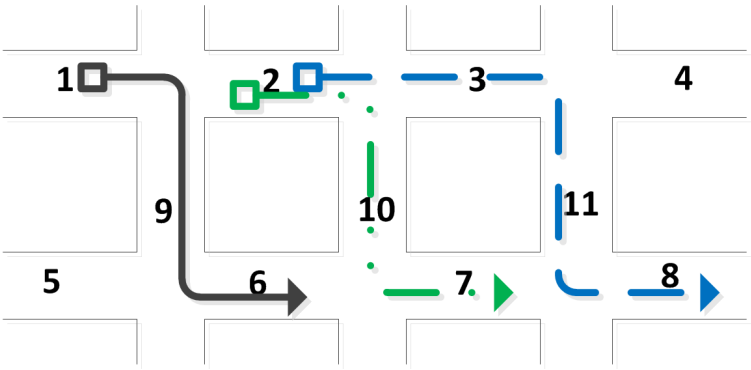


Figure 8.1: Example paths of three probe vehicles on a network. The network has eleven links. The path of a probe is represented as a vector $a_i \in [0,1]^{11}$ where the j^{th} coordinate of a_i represent the fraction of link j traveled by the probe. The path represented with a solid line is represented with a sparse vector with non zero coordinates 1, 6 and 9, respectively equal to 0.4, 0.7 and 1 considering that the probe traveled 40% of link 1 and 70% of link 6. The vector representing the dashed path has non zero coordinates 2, 3, 8 and 11, respectively equal to 0.3, 1, 0.8 and 1 considering that the probe traveled 30% of link 2 and 80% of link 8.

if the vehicle did not travel on link j and $a_{i,j} = 1$ if the vehicle fully traversed link j (see Figure 8.1).

Spatial regularization

The numerical results first investigate the addition of an l_1 regularization on the spatial variations of the travel times on the graph. The application of this regularization to arterial traffic estimation is interesting for several reasons. First, it exploits an intuitive idea that traffic conditions should be similar in neighboring links of the network and improves the estimation capabilities when little and/or noisy data is available. Traffic signals cause important variation on the travel time experienced on a link of the network and regularization is important to prevent overfitting. Second, it exhibits the inherent spatial structure of traffic by noticing the area where traffic conditions actually change. Finally, by exploiting the sparse structure of the solution, the algorithm can update efficiently the traffic estimates as soon as new measurements become available.

[6]Compared to the previous chapters, the locations on the network are denoted ξ_1 and ξ_2 (instead of x_1 and x_2).

[7]The coefficients $a_{i,j}$ can account for the fact that travel time on a fraction of the link does not vary proportionally with the distance traveled as vehicles are more likely to experience delays close to signalized intersections as demonstrated in Chapter 7.

The average travel time $x^n = K^{-1}z^n$ at time t^n is computed by solving (8.4). The additional l_2 regularization leverages the historical mean travel times $\hat{x}$. At each estimation time t^n, the regularization parameter is updated (from $|I_n|\mu_0$ to $|I_{n+1}|\mu_0$) and the new observation added. The parameters of the l_1 and l_2 regularizations (respectively μ_0 and λ) are chosen via cross-validation as described in the following paragraphs. Observations may remain relevant only for a *limited period of time*[8], denoted T. When observations become obsolete, the algorithm updates the regularization parameter and removes the old observations.

The choice of the matrix K_1 represents the prior information on the spatial dependencies between the estimates. The numerical analysis investigates different choices for the matrix K_1 and studies their respective performance. The first choice of K_1 encourages all the incoming links of an intersection to have the same pace (inverse of velocity). Let $\mathcal{J}$ be the set of junctions j with n_j incoming links ($n_j \geq 1$) and let $\mathcal{I}_j = \{i_j^1, \ldots, i_j^{n_j}\}$ be the set of incoming links of junction j. To each junction j corresponds $n_j - 1$ rows in K_1. The k^{th} row has non zero entries for the incoming links k and $k+1$ of junction j (denoted i_j^k and i_j^{k+1}). These entries are respectively $1/L(i_j^k)$ and $-1/L(i_j^{k+1})$ where $L(i)$ is the length of link i. Another choice of K_1 encourages the outgoing links of each junction to have the same pace. The results for both choices are compared in Figure 8.2 (bottom).

At time t_n the estimate x_n is computed using the observations in I_n. The prediction error e_n is defined using the current estimate to predict the future travel time as $e_n = |a_{n+1}x_n - y_{n+1}|$. Figure 8.2 analyzes the effect of the choice of the parameters λ and μ_0 as well as the choice of matrix K_1. The numerical results indicate that both the l_1 and l_2 regularizations improve the results for a wide range of λ and μ_0. As the error is not very sensitive to the choice of these parameters, they can be calibrated off-line using cross-validation. Figure 8.2 (bottom) also indicates that the choice of the regularization matrix K_1 influences the accuracy of the estimation. The regularization on the outgoing links always provides better results than the choice of regularization on the incoming links.

The results can also be represented as a traffic map with colors representing the pace of the vehicles: green for smallest pace *i.e* fastest speed, red for largest pace. White pins indicate the intersections for which the algorithm detects spatial variation of the pace. The pins tend to cluster in a few regions of the network, indicating regions with important spatial variations in the traffic conditions.

Imposing and exploiting a sparsity structure on the solution limits the computational cost of traffic estimation on large networks as the algorithm leverages the sparsity of the solution in the algorithm. The number of transition points and active indices remain small throughout the algorithm with an average of 0.5 transition points per estimate update (addition of new data points and variation of the regularization parameter) and 20 active regularized indices for a network with 815 links.

[8] Typically, T is in the order of five to fifteen minutes

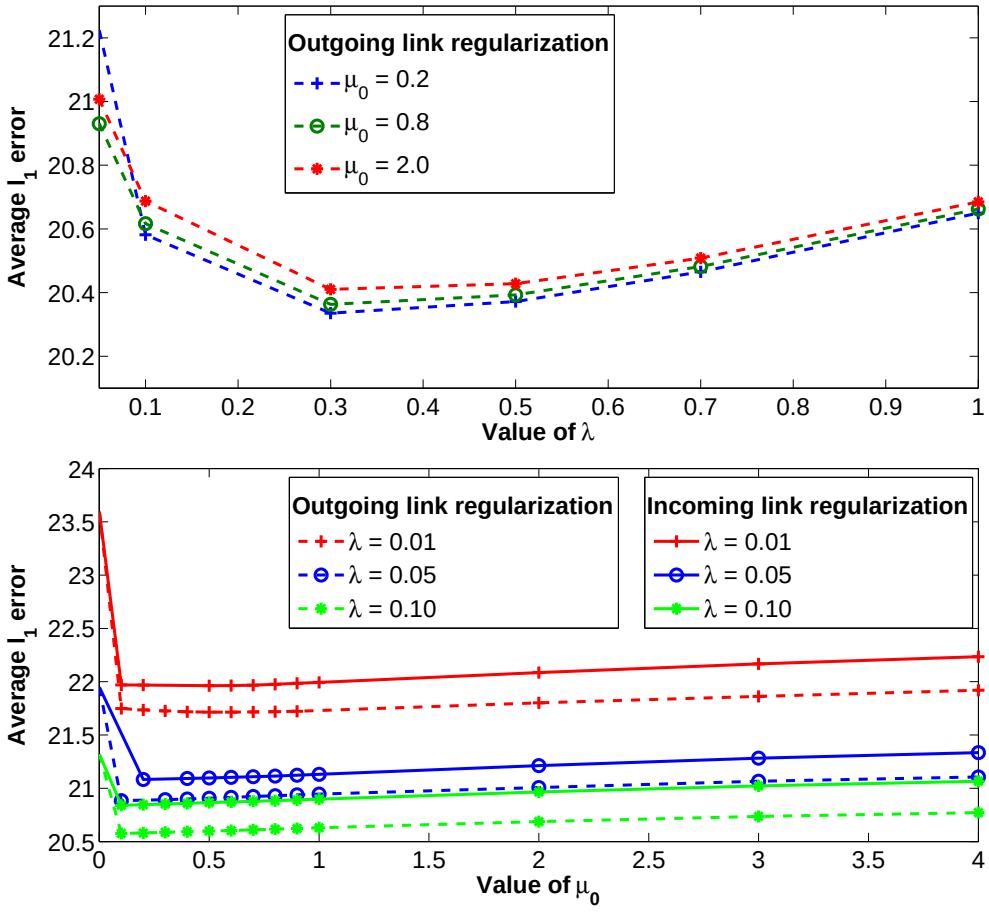


Figure 8.2: Variation of the l_1 error in function of the regularization parameters for the l_1 and l_2 penalization when encouraging sparsity on the spatial variations of traffic conditions. The figures represent the variation of the error for different values of λ (top) and μ_0 (bottom). The figures indicate the importance of the additional l_2 regularization to improve the accuracy of the estimation.

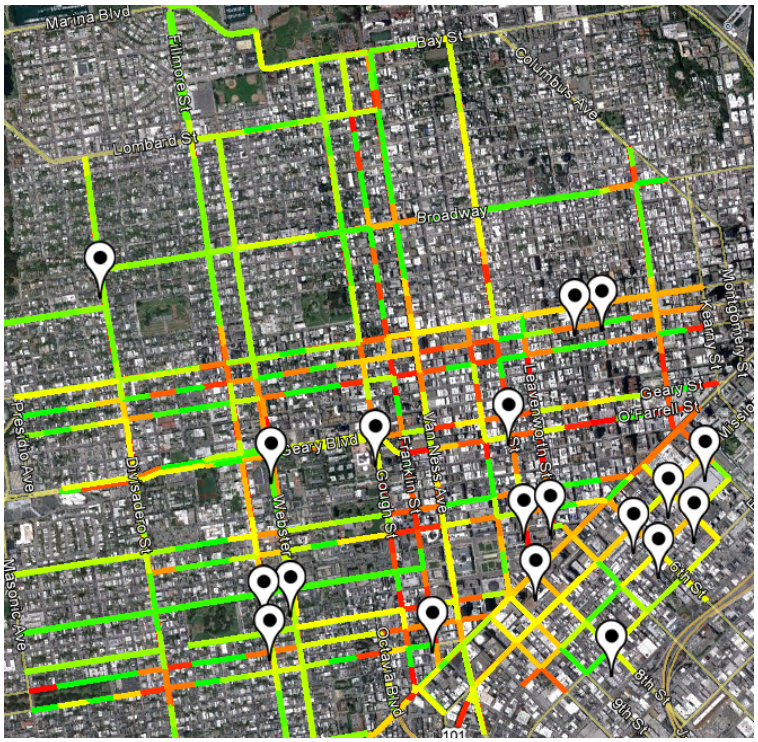


Figure 8.3: Geographical representation of the traffic estimation results with detection of the spatial variation of the pace in the network. The color of each link varies with the pace (green for small paces, red for large paces). The pins indicate the intersections for which not all outgoing links have the same estimated pace (inverse of the speed).

Temporal regularization

Besides spatial variation, arterial traffic is subject to important temporal variations. Some of these variations are due to changes in traffic conditions (level of congestion) whereas other follow the periodic dynamic of the signalized network. As underlined before, traffic data on arterial networks is mainly provided from probe vehicles sending their location at a given sampling frequency (common sampling frequencies are around 1 minute). The proportion of sampled vehicles (penetration rate) remains limited and rarely exceeds a few percent of the vehicles traveling on the network. Moreover, traffic signals cause important variation on the travel time experienced on a link of the network within very short periods of time (depending on whether the vehicle stopped at the signal or not), while the actual changes in traffic conditions have slower dynamics. Given the penetration rate of probe vehicles, the algorithm seeks to estimate trends in traffic conditions rather than fluctuations around a mean value. For these reasons, arterial traffic estimation is a good application for the algorithm. The parameter x^n represents the average travel time on each link at t^n. The algorithm updates the solution online as new measurements are available (or old one are

obsolete) while encouraging sparsity on its temporal evolution.

The parameter x^n is computed at time t^n by solving equation (8.13). Recall that x^n represents the mean travel time on each link of the network at time t^n. The algorithm is initialized using a previous estimate of the mean travel times given by least-squares regression. As for the spatial regularization the historical mean travel times $\hat{x}$ is used to add a l_2 regularization term $||x - \hat{x}||$. At each estimation time, the regularization parameter is updated (from $|I_n|\mu_0$ to $|I_{n+1}|\mu_0$), the new data is added and the reference parameter is updated (from $\bar{x}^n = x^{n-1}$ to $\bar{x}^{n+1} = x^n$).

The performance of the model is assessed using cross-validation, randomly splitting the observations sent by the probe vehicles between a training set and a validation set. After learning the travel time estimates on the training set, the validation set is used to compare the estimates to the travel time observations. The performance of the model is compared with a *baseline model*, which uses the historical value of the link travel times $\hat{x}$ as the estimate of the state. Three metrics quantify the quality of the estimation: the root mean squared error (RMSE), the mean absolute error (MAE) and the mean percentage error (MPE)[9]. Note that the variability of arterial travel times (due to traffic signals, pedestians, etc.) leads to important fluctuations of travel times. This inherent variability in the state of the system makes the estimation model robust with sparse variations, but is also responsible for relatively high values of the error metrics.

The numerical analysis assesses the performance of the model and quantifies the effect of the regularization parameters λ and μ_0. The first parameter penalizes solutions which are far (in the l_2-norm sense) from the historical estimate of travel times $\hat{x}$. The second parameter imposes sparsity on the variation of the estimate. The choice of these parameters leads to a compromise between (i) fitting the data, with risks of overfitting and lack of physical interpretation and (ii) putting too much weight on the regularization and not estimating accurately the current state of the system.

The results indicate that both the l_1 and the l_2 regularization (Figure 8.4) are important to improve the estimation capabilities. For a wide range of parameters, the results are significantly better than the baseline model. The results also underline the importance of the additional l_2 regularization to improve the robustness of the algorithm. Figure 8.5 illustrates that in addition to improving the estimation capabilities, the algorithm produces results that are easier to interpret. Arterial traffic is highly variable and the variability often prevents the interpretation of the results. This model estimates the trends in travel times on the links of the network, while filtering the variability due to the signal dynamics.

[9] $\text{RMSE} = \sqrt{\frac{\sum_{o=1}^{0}(y_o - \hat{y}_o)^2}{O}}$, $\text{MAE} = \frac{\sum_{o=1}^{0}|y_o - \hat{y}_o|}{O}$, $\text{MPE} = \frac{1}{O}\sum_{o=1}^{0}\frac{|y_o - \hat{y}_o|}{y_o}$.

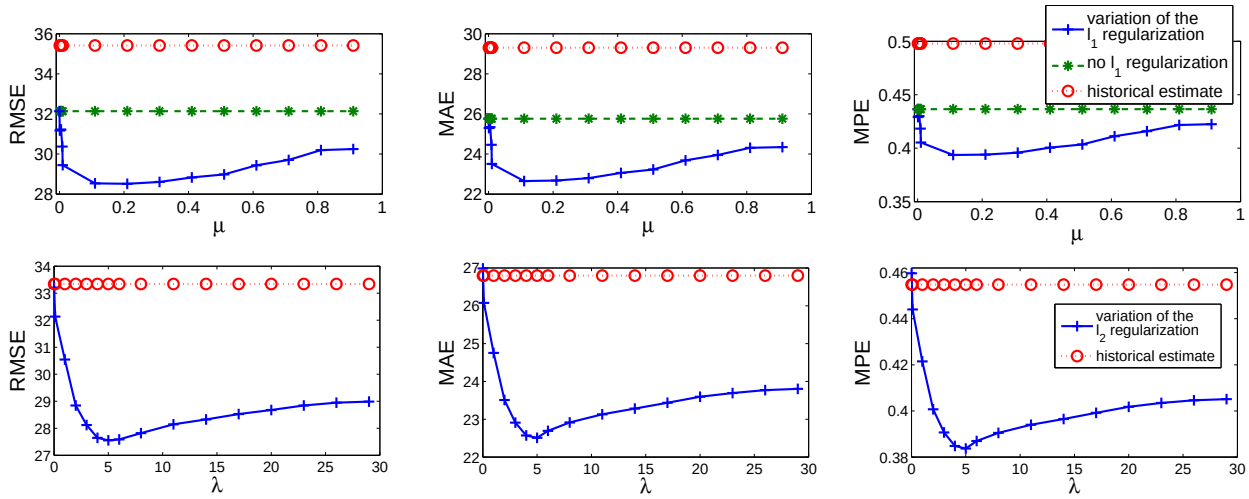


Figure 8.4: Variation of the error metrics in function of the regularization parameters for the l_1 and l_2 penalization when encouraging sparsity on the temporal variations of traffic conditions. Both the l_1 and l_2 regularizations improve the estimation accuracy and the regularization parameters can be chosen optimally. The three top figures represent the effect of the l_1 regularization for the estimation accuracy. The three bottom figures show the importance of the additional l_2 regularization introduced in Section 8.3 for the robustness of the estimation.

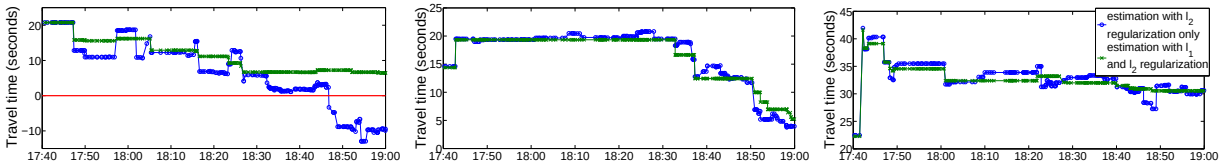


Figure 8.5: Qualitative evolution of the travel time estimates on different links of the network. The l_1 regularization provides more stable estimates that represent the dynamics of traffic more accurately and increase the physical interpretation. The left figure shows that the estimation with l_2 regularization leads to estimates that are not physically possible (negative travel times), while the estimate with l_1 regularization remains within feasible bounds. On all figures, the l_2 estimate is noisy while the additional l_1 regularization remains constant between each temporal transitions in traffic conditions.

8.6 Conclusion and discussion

The chapter derives an online-algorithm to update the solution of linear regression problems with a large class of l_1 and l_2 regularizations as new observations become available. The l_1-norm regularization improves the estimation capabilities and the interpretability of the results by exhibiting and exploiting the underlying sparsity structure of the problem. The additional l_2-norm regularization increases the robustness of the estimator and limits numerical issues. Compared to previous work on the LASSO, the algorithm provides the ability to (i) impose sparsity on a linear function of the estimate, (ii) update the solution online by computing a homotopy as new measurements become available (or as old measurements become obsolete) and (iii) impose sparsity on the variations of the state with respect to a reference parameter which can be updated at any time, for example to impose sparsity on successive estimates.

The homotopy algorithm leverages the sparsity of the solution to reduce the computational complexity and is thus particularly efficient when the solution is sparse. The computational costs at each transition point is limited by updating the matrix inverses with low-rank updates. The number of transition points and active indices varies with the parameter μ. As μ increases, the number of transition points and active indices decreases, improving the computational efficiency of the algorithm.

This generalized LASSO algorithm has the potential to improve real-time traffic estimation capabilities from streaming probe vehicle data in large urban networks. Besides providing significant improvement of the estimation accuracy, the algorithm improves the understanding and potentially the modeling of the spatial and temporal variations of traffic across the network. For example, the detection of temporal changes can be used to trigger an update in the data used for the estimation as old data may be outdated.

Chapter 9

Large scale pattern analysis

Estimating and analyzing dependencies and trends between variables at a large scale is an inherently difficult task. Chapter 8 develops an online algorithm to detect spatio-temporal changes in the state of a network. The algorithm is valuable to improve estimation capabilities of existing models and algorithms, such as the ones presented in Chapters 6 and 7. This chapter suggests a different data-driven approach to provide a global network-level analysis of patterns using dimensionality reduction (matrix factorization) and clustering methods. These techniques allow us to characterize spatial traffic patterns in the network and to analyze traffic dynamics at a network scale. The chapter identifies patterns that indicate intrinsic spatio-temporal characteristics over the entire network and give insight into the dynamics at a large scale.

Most of previous research in traffic data analysis focus on temporal dynamics of individual links (either on arterial or highways). Very little progress has been made in analyzing the temporal dynamics of *global* traffic states of an *entire large-scale road network*. As underlined in previous chapters, traffic states of neighboring individual roads are often highly correlated (both spatially and temporally) and the identification of specific traffic patterns or traffic configurations is very informative. They can be used to better understand global network-level traffic dynamics and serve as prior knowledge or constraints for the design of traffic estimation and prediction platforms. The analysis of traffic patterns is also useful for traffic management centers and public entities to plan infrastructure developments and to improve the performances of the available network using large-scale control strategies.

This chapter proposes an algorithm to identify spatial configurations of traffic states over the entire network and analyze large-scale traffic dynamics from traffic state estimates produced and collected over long periods of time. A *network-level* traffic state is defined as the aggregation of the congestion states of all the links of the network at a given time. It is represented in the form of multi-variate data, where its dimension is proportional to the number of links in the transportation network. As the size of the network increases, it becomes difficult to have an overview of the network and to notice patterns in the dynamics. In machine learning, this issue is commonly addressed using *dimensionality reduction* techniques to simplify the representation of the data, remove redundancies and improve the efficiency

of analysis techniques such as classification [126, 99]. Important applications of these algorithms include image processing and natural language processing [94, 76]. This chapter investigates the use of a dimensionality reduction matrix factorization technique known as *Non-negative Matrix Factorization* (NMF) [34, 119] to obtain a low dimensional representation of network-level traffic states. *Principal Component Analysis* (PCA) and *Locality Preserving Projection* (LPP) are other examples of matrix factorization methods [126, 99].

However, in contrast to PCA and LPP, the NMF algorithm imposes strict non-negativity constraints on the decomposition result. This allows NMF to approximate the n-dimensional data vector by an *additive* combination of a set of learned bases. This property also leads to a part-based representation of the original data. The learned bases correspond to *latent components* of the original data so that the original data is approximated by a linear *positive* superposition of the latent components. The properties of the NMF have already been exploited for various applications. In text analysis, the learned bases are used to label different latent topics contained in text documents. In face image representation, the NMF bases indicate important localized components of the face, such as the eyes, the mouth or the cheeks. The positive superposition of components gives NMF a lot of potential to model the dynamics of a physical system. The low-dimensional representation of network-level traffic states should exhibit global configurations of local traffic states and reflect intrinsic traffic patterns of network-level traffic states.

The chapter is organized as follows. Section 9.1 introduces the NMF algorithm. The algorithm is used to perform large scale analysis of the dynamics of traffic. Sections 9.2 and 9.3 provide a detailed analysis of typical spatial configuration patterns of network-level traffic states found by NMF projections. Section 9.4 further analyzes temporal dynamic patterns of the network-level traffic state, which describe evolutions of traffic states in the whole network.

9.1 Learning patterns with Non-negative matrix factorization (NMF)

This section presents the *Non-negative Matrix Factorization* (NMF) dimensionality reduction algorithm. It is used to approximate network-level traffic states as positive sums of a limited number of global traffic *configurations*. NMF [150, 34, 156, 119, 76] is a particular type of matrix factorization, in the same domain as the well-known *Principal Component Analysis* (PCA) method and *Locality Preserving Projection* (LPP).

In all cases, given a set of multivariate n-dimensional data vectors placed in m columns of a $n \times m$ matrix X, matrix factorization decomposes the matrix into a product of a $n \times s$ loading matrix M and a $s \times m$ score matrix V, where s represents the dimensionality of the subspace on which the original data is projected. Through this matrix decomposition, each n-dimensional data vector is approximated by a linear combination of the s columns

of M, weighted by the components in the corresponding column of V. The s column vectors of M represent a group of projection bases that are learned to optimally represent the original data. The variable s is typically chosen to be significantly smaller than both n and m so that the obtained score matrix V forms a low-dimensional subspace projection of the network-level traffic states, on which further data analysis is performed. The specificity of NMF is the enforced *positivity* of both the weights in V, and of the columns of M forming the NMF decomposition basis. This non-negativity therefore provides an approximation of the n-dimensional data vector by an *additive* combination of a set of learned bases. Furthermore, the NMF components forming the basis tend to be sparse, which leads leads to a part-based representation of the original data.

As introduced at the beginning of the chapter, a network-level traffic state is a vector of size equal to the number of links in the network, where the i^{th} entry corresponds to the traffic state on the i^{th} link of the network. Arterial networks are typically dense (numerous links and intersections) and the number of links in any decent size network is often over a thousand links. Assuming that k samples of n-dimensional network-level traffic states are stored as an $n \times k$ matrix X, NMF factorizes X as the product of a non-negative $n \times s$ matrix M and a non-negative $s \times k$ matrix V. The matrices M and V are chosen to minimize the Frobenius norm of the reconstruction error between X and its factorized approximation MV. The Frobenius norm of a matrix $A \in \mathbb{R}^{n \times m}$ with entry on column i and line j denoted $A_{i,j}$ is defined as

$$||A||_F = \sqrt{\sum_{j=1}^{n} \sum_{i=1}^{m} |a_{i,j}|^2}.$$

It is equal to the sum of the singular values of A. The matrix factorization problem reads:

$$\arg \underset{(M,V)}{\textbf{minimize}} \|X - MV\|_F \text{ s. t. } \quad M \geq 0,\ V \geq 0, \tag{9.1}$$

where the inequalities $M \geq 0,\ V \geq 0$ represent the non-negativity constraints (each element of the matrices are non-negative). The optimization problem (9.1) is not convex in (M, V). However, the objective function is convex in M (when fixing V) and in V (when fixing M). Equation (9.1) is solved using multiplicative updates [150]. The algorithm alternatively fixes the matrices M and V and updates the non-fixed matrix. Multiplicative updates and other gradient based optimization procedure do not guarantee the global optimum of the NMF solution. Nevertheless, a local minimum provides a possible factorization of the original data which is useful for further data analysis. The NMF projects the high-dimensional network-level traffic states on a s-dimensional subspace, which is spanned by the columns of M. According to equation (9.1), the column space of V corresponds to coordinates of network-level traffic states with respect to the learned set of bases in M. The column space of V forms a low-dimensional representation of the network-level traffic states. As mentioned in the introduction, each network-level traffic state $X_j \in \mathbb{R}^n$ is approximated by an additive

linear superposition of the column space of M due to the non-negative constraint. The approximation of X_j is written as

$$X_j \approx \sum_{i=1}^{k} M_i V_{i,j}, \tag{9.2}$$

where M_i denotes the i^{th} column of M and $V_{i,j}$ is the element at the i^{th} column and j^{th} row of V. It is valuable to interpret what the matrices M and V represent to conduct further data analysis in the low dimensional space. The column space of M represents typical elements of the spatial configuration patterns with respect to the network-level traffic states. The columns of M represent complex spatial arrangements of local traffic states over the entire network. As for V, equation (9.1) indicates that each element $V_{i,j}$ represents to which degree the j^{th} network-level traffic state observation is associated with the i^{th} expanding basis in matrix M (i^{th} spatial configuration). For example, if the spatial configuration formed by the i^{th} column of M is the best representation of the j^{th} network-level traffic state, then $V_{i,j}$ will take the largest value in the j^{th} row of V [34]. As a result, the derived low-dimensional representation formed by the columns space of V are intuitively consistent with information about spatial distribution patterns of local traffic states. By contrast, the PCA and LPP based projections aim at best reconstruction of the original data by either maximizing data variances or preserving neighboring structures. The projection results of PCA and LPP are thus less likely to be associated with interpretable latent traffic configuration patterns than the NMF. Therefore, NMF appears as an appropriate choice to analyze the network-level traffic states.

In the analysis, the traffic states used for the clustering analysis are *fluidity indices*. A fluidity index is a value in $[0, 1]$ computed as the ratio between the free flow travel time and the estimated travel time. They are provided by an estimation algorithm described in [104, 115]. The model estimates travel times and fluidity indices from the streaming data and leverages the historical data using a Bayesian update. The estimates are updated on each link of the network every five minutes. The numerical study focuses on a network consisting of 2626 links for a duration of 184 days, from 00:00 May 1st 2010 to 23:55 October 31st 2010, totaling 52292 estimates per link ($12 \times 24 \times 184$). The fluidity index of each link at each time sampling step is stored in a matrix X containing 2626 rows and 52292 columns. The clustering results include two parts. First, a clustering of network-level traffic states discovers typical spatial configurations of network-level traffic states (Section 9.2). Second, a clustering of temporal trajectories of network-level traffic states provides an analysis of traffic dynamics (Section 9.4).

9.2 Congestion patterns: spatial configurations of global traffic states

An important outcome of the dimensionality reduction is to identify typical spatial congestion patterns (i.e. spatial configurations of congestion). NMF has one essential parameter: the number s of components over which decomposition is done. The parameter s also corresponds to the dimension of the target subspace in which clustering is performed. The choice of s is empirical (s is called a *meta parameter*) and is done by analyzing results obtained for increasing values of s. In this chapter, the choice of s is made based on a trade-off between the reconstruction error (value of the objective function (9.1) at optimum) and the quality of the clustering results. The reconstruction error continually decreases as the dimension s increases. This result is expected as the optimization problem (9.1) is performed on a larger set and thus the factorization models with higher complexity always leads to better fitting to the original data. However, the rate of improvement of the objective function (with respect to the dimension s) is a good indicator to choose the meta-parameter s. The clustering of global traffic states consists of clustering the traffic data projected in the s-dimension subspace using a *k-means* algorithm [159, 130]. The k-means algorithm is a widely used unsupervised clustering algorithm. It partitions observations into k clusters in which each observation belongs to the cluster with the nearest mean. The clusters obtained in the s-dimensional space are displayed in three dimensions. This means that the number of NMF components displayed is three but that the clustering results were obtained in the s-dimensional space. The parameter s defines the importance of the dimensionality reduction as a trade-off. Higher values of s conserve more information contained in the original data. However, lower values of s filter more redundancy and noise in the data and lead to much more computationally efficient algorithms. A preliminary analysis showed that values of s inferior to eleven lead to clustering results which seem visually inadequate: the 3D representation of the clusters shows important overlap between the clusters. The clusters become separated for values of s greater than fifteen. Increasing s over 15 does not seem to bring any improvement in the clustering results, while it significantly increases the NMF computation and memory usage costs. Therefore, the number of NMF components is set to $s = 15$ for all subsequent analysis. This value achieves a balance between the descriptive power of NMF projection and the computational efficiency.

The number of clusters k arises as another meta-parameter. The choice of k does not influence the computational costs significantly but changes the interpretability of the results. The number of clusters represents the number of *global congestion patterns* that may arise. Too low values of k may not represent the different congestion patterns whereas too high values of k may decrease the possibilities of interpretation by separating similar congestion states into different clusters. After analyzing the results obtained for increasing values of k, it seems that the most insightful clustering is obtained with $k = 5$ clusters. The average fluidity index value (obtained by averaging index values on all links) are shown for each of

the five clusters in the table at the top of Figure 9.1. It appears that two clusters (cyan squares and green stars) correspond to different types of "mostly fluid" states, whereas the remaining three clusters (blue circles, yellow diamonds and red stars) represent "congested states". The physical meaning of each cluster is analyzed by constructing histograms of the fluidity index values. Fluidity index values in the *night and early morning Free-Flow* (NFF) and *Evening Free-Flow* (EFF) cluster are higher as a whole than those in the clusters corresponding to occurrences of congestion (*Morning Increasing Congestion*, *Mid-Day Congestion*, and *Afternoon Decreasing Congestion* clusters).

As done in the primary analysis for the choice of s and for visualization purposes, the spatial layouts of the global traffic state distribution is illustrated in 3D-NMF space (obtained by requesting 3 components only instead of 15), while the k-means clustering algorithm is run in the larger 15-D NMF space. The physical interpretation of the five clusters is clear in Figure 9.1. The figure shows all global traffic states projected in 3D-NMF, together with a typical temporal evolution trajectory of a single day. The global traffic state trajectory is indicated by the blue line in Figure 9.1. The green star and red circle are the starting point and ending point of the trajectory, corresponding to global traffic states at 00:00 and 23:55 respectively. The temporal arrangements of the network-level traffic states along the trajectory underline the temporal interpretation of the five clusters: the green-star cluster corresponds to night and early morning free-flowing, from which typical day evolution goes into morning intermediate states (before 10:00) corresponding to the blue-circle cluster; mid-day congestion (red-star cluster) generally occurs between 10:05 and 15:00, and represents a clearly different congestion state in 3D-NMF space, with a sudden jump of traffic states from the blue-circle cluster to the red-star one, and sudden jump back into the afternoon intermediate state (yellow-diamond cluster) around 15:00. The traffic settles to a specific evening near-free-flow state from 18:00 to 23:55 (cyan-square cluster). Interestingly, both the projection of the global congestion states in 3D-NMF space and the clustering results in 15D-NMF show a clear distinction between morning and afternoon intermediate congestion states, and also between late evening and night/early-morning near-free-flow states.

Figure 9.2 shows traffic patterns corresponding to spatial configurations of congestion for the centers of each cluster. Each cluster center is computed by averaging all elements of the corresponding cluster, so as to indicate a representative spatial configuration of traffic states of each cluster. The figure displays the links with fluidity index values less than 0.7 (congested links) on the Google Map screen-shots. For the five clusters, most of the congestion occurs within the regions highlighted by the dashed circle in Figure 9.2. This region corresponds to the downtown region of San Francisco. Compared with the downtown region, the western and southern region of San Francisco are less likely to suffer from congestion (left and bottom part in the San Francisco road network). This analysis is very useful for traffic management centers and public entities to understand the most important bottlenecks that cause heavy traffic conditions. Moreover, the results show that some of the major bottlenecks remain constant throughout the day whereas others evolve with the different traffic patterns of

Marker symbol	Average fluidity	Cluster name
Green stars	0.7757	Night + early morning Free-Flow (NFF)
Blue circles	0.7185	Morning Increasing Congestion (MIC)
Red stars	0.6393	Mid-Day Congestion (MDC)
Yellow diamonds	0.6730	Afternoon Decreasing Congestion (ADC)
Cyan squares	0.7420	Evening Free-Flow (EFF)

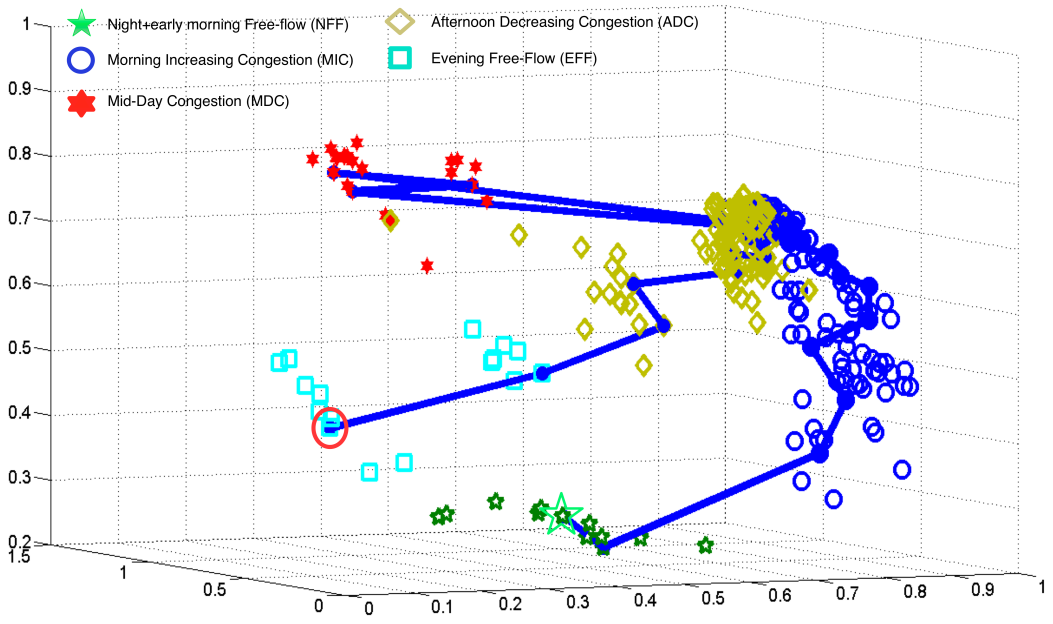
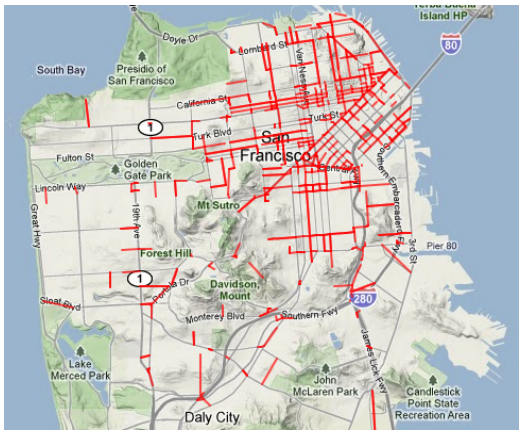
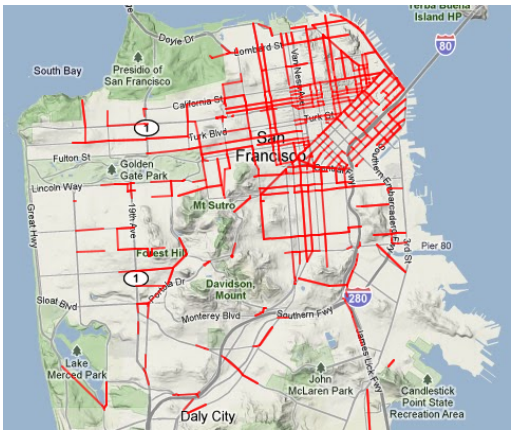


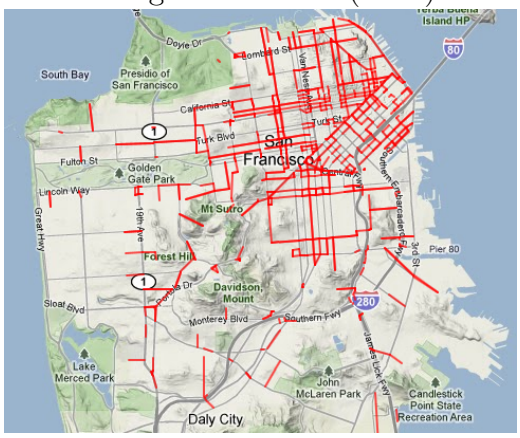
Figure 9.1: Results of the k-means algorithms on the low-dimensional projection of network-level congestion states. The clustering exhibits different times of the day corresponding to different configurations of network-level congestion states. The table shows the average fluidity values of each of the global state clusters. The figure shows the temporal evolution of global congestion states, projected in the 3D-NMF space using different colors and symbols to represent the five different clusters. The first and the last network estimates of the day are represented with a large star and a large circle respectively.



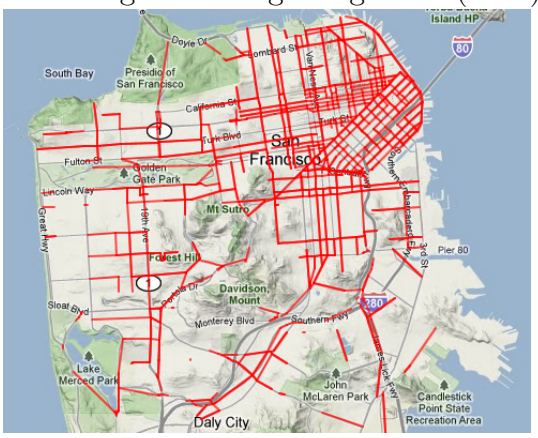
Night Free-Flow (NFF)



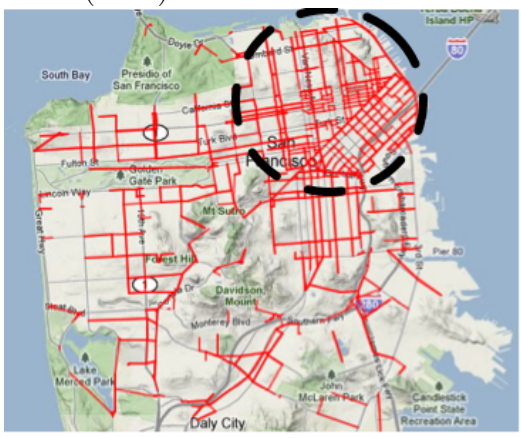
Morning Increasing Congestion (MIC)



Evening Free-Flow cluster (EFF)



Afternoon Decreasing Congestion (ADC)



Mid Day Congestion (MDC)

Figure 9.2: Typical spatial configurations of traffic states for the five cluster centers of network-level traffic states. The figures display the links with fluid index values less than 0.7 (congested links). Most of the congestion occurs within the dashed circle, which is the downtown region of San Francisco. The NFF and EFF clusters have a smaller number of links highlighted than the MIC, ADC and MDC clusters indicating the difference in congestion levels.

the day. This dynamical analysis can lead to specific management strategies to address this recurring congestion. For example, a regression model in the lower dimensional space has a lot of potential to predict the global traffic dynamics [96]. This work indicates the promising potentials of spatial congestion patterns in forecasting congestion and improving traffic management.

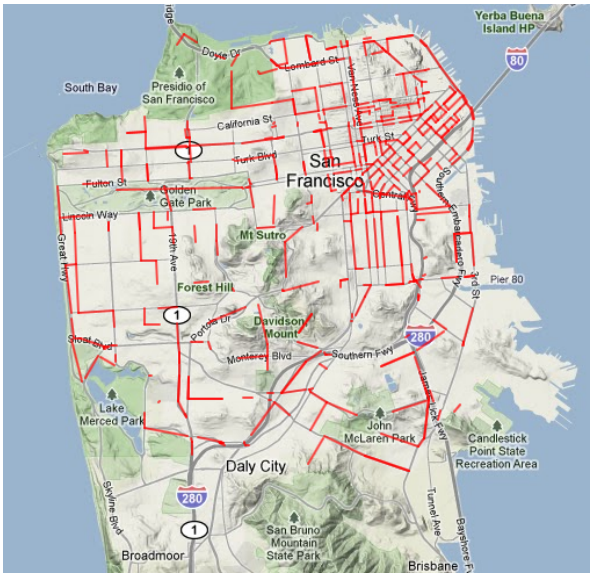
9.3 Spatial decomposition of the road network

Another motivation for using NMF is its property to approximate original data by an additive linear combination of a limited set of "components" (often denoted *NMF basis*). Due to the non-negativity constraints, spatial arrangements of the components are usually sparse, which means that values in most regions of each basis are (close to) zero except several localized regions. These localized regions with large values correspond to typical patterns or representative components of the original signals (the global congestion states). They typically correspond to independent "parts" of the data. This property has lead to successful applications of NMF to extract part-based representation or latent semantic topics from the data in image processing or text classification. For example, when NMF is applied to image datasets, it automatically extracts some part-based representation of the type of objects present in the images [150, 119, 76]. The section studies this "part-based" representation of global congestion states to analyze the physical significance of NMF components obtained on traffic data and improve the understanding of congestion patterns.

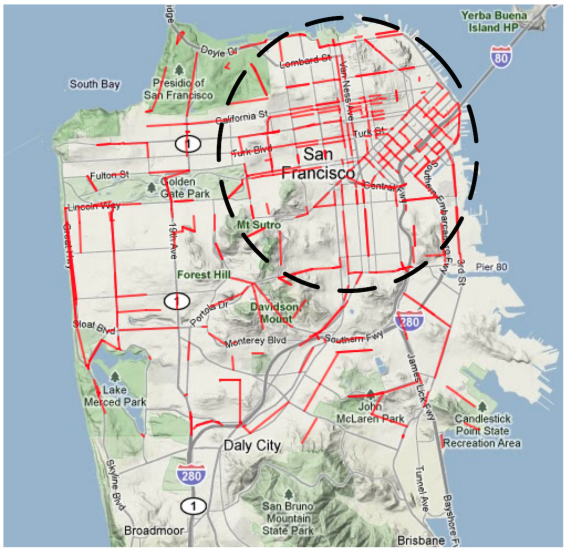
For arterial traffic, the regions with distinctively large values in each NMF basis correspond to a group of links with highly correlated traffic states. This section constructs the components by selecting the links which represent the top 20% largest values in each basis. The components are represented by displaying the selected links (red line) of the road network. Figure 9.3 shows several typical spatial arrangements of localized components, out of the fifteen components learned during the NMF training. Out of the different components, some are very informative regarding typical congestion patterns. One of the components corresponds to streets in a localized West region ("West Part" NMF component in Figure 9.3), and another to streets in the central region ("Central" NMF component in Figure 9.3), which could indicate that the traffic on the links within each of these regions is highly correlated whereas the traffic between distinct regions exhibits relatively independent behaviors. Such a characterization of independent regions of traffic dynamics is important to significantly reduce the computational costs of a large variety of estimation models, in particular estimation models based on graphical models, such as the ones presented in Chapters 6 and 9 or in [82, 105]. The characterization of different regions with limited dependency has a lot of potential to design approximate inference algorithms to reduce the computational costs while maintaining an accurate representation of traffic dynamics and limiting the estimation error [30].

Other NMF components highlight correlations of traffic in parallel directions. On Figure 9.3, a majority of the links of the "East-West transit" NMF component are horizontally-

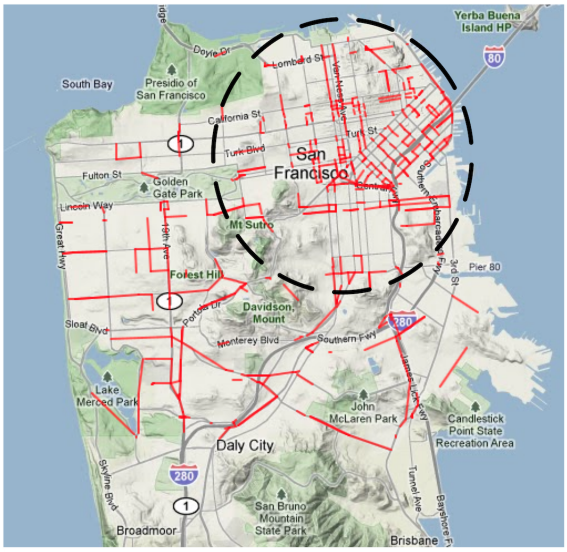
"West Part" NMF component



"Central" NMF component



"East-West transit" NMF component



"North-South transit" NMF component

Figure 9.3: Examples of NMF basis, either highlighting localized correlations (top figures), or flow-direction correlations (bottom figures).

oriented, wheras a majority of the links of the "North-South transit" NMF component are vertically-oriented. As highlighted in the figure, the links of these components which are close to the downtown tend to be more consistent with the orientation pattern. These links with similar orientations are likely to have correlated dynamics, whereas links with orthogonal orientations have less influence on each other. These correlations properties can be used to learn the structure of the graphical model representing conditional independences between traffic states on the network (both spatially and temporarily).

According to the physical representation of the NMF components, it seems that different NMF bases focus on different localized connected regions of the network. This could imply that NMF detects both strong correlation of traffic dynamics within each localized region and relative independence between these regions. However, this connectivity and localization of the components could be improved. Standard NMF does not guarantee connected nor localized components and the above promising results motivate further investigation. A possible approach is to modify the NMF algorithm in order to favor *localized* sparsity, which should help to unveil more distinct part-based network decomposition.

9.4 Temporal analysis of global traffic states

This section analyzes the daily dynamics of network-level congestion states projected in the NMF space. This analysis is important to understand how congestion forms and dissipates throughout the day. For each day in the studied period, the trajectory of the network-level traffic states in the NMF space is represented using the projection on the lower dimensional space. The projections are connected to form a solid curve representing the trajectory. Notice that trajectories are nearly closed in the NMF space. Note that for visualization purposes, the projection is done on the 3D-NMF space. Figure 9.4 (top) shows a typical day trajectory with successive temporal intervals along the trajectory plotted using different colors, to give an idea of the dynamics along the curve.

It is noteworthy that over the 184 days of reconstructed traffic data, there are only, in 3D-NMF projection, exactly seven different typical trajectories, as shown in Figure 9.4 (center). Furthermore, each one of these seven typical trajectories corresponds to a particular day of the week and are thus called *day trajectory patterns*. Note that individual day trajectories for same day-of-week, although superposed in 3D-NMF, are slightly different from another in 15D-NMF space, in which clustering is performed.

Differences between the different day trajectory patterns concentrate within the time interval corresponding to transitions between congestion states, in particular between the morning increasing congestion and the mid-day congestion and between the mid-day congestion and the evening decreasing congestion. Characterizing these specific time intervals that represent the differences in daily dynamics allows us to identify and/or predict different evo-

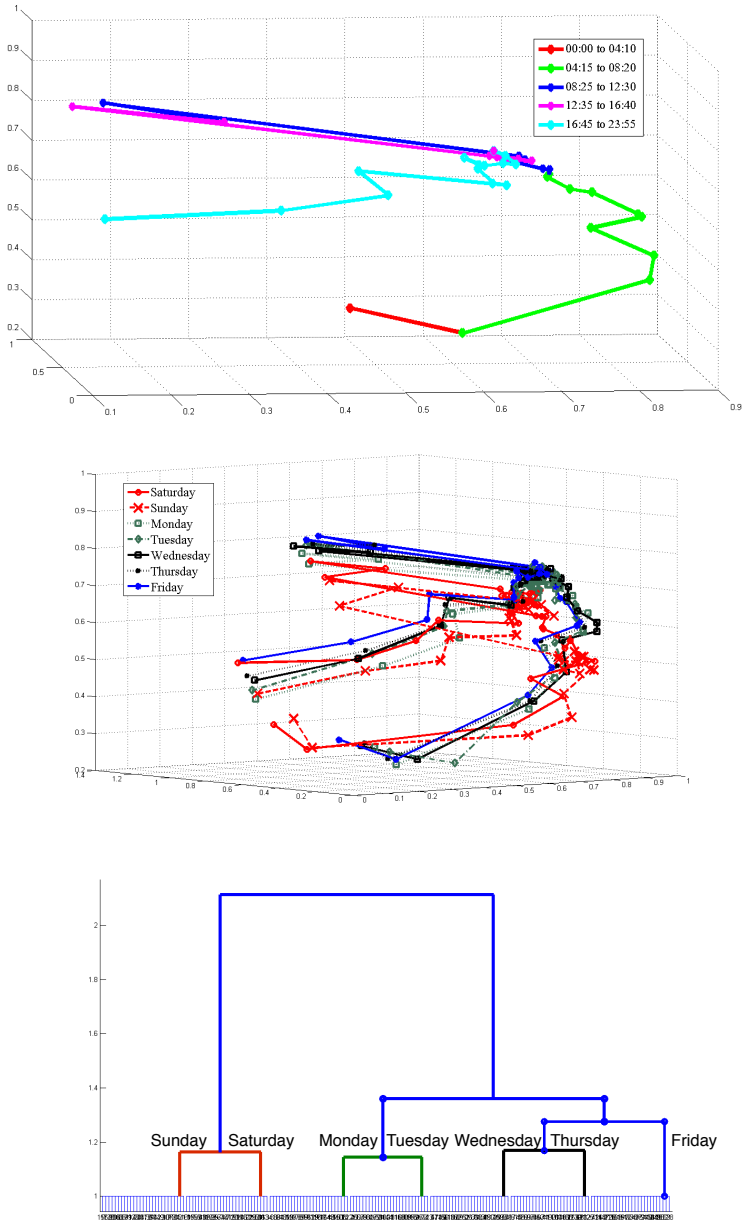


Figure 9.4: Daily trajectories of network fluidity indices projected in 3D-NMF space exhibit seven different typical trajectories, representing the days of the week. **Top**: Example of a daily trajectory with coloring representing the five different times of the day. **Center**: The seven different trajectories, representing a typical daily dynamic for each day of the week. **Bottom**: Dendrogram representing the hierarchical clustering analysis of the daily trajectories.

lution patterns of traffic states and to develop mid-term or long-term traffic forecast [97, 96].

In this data set, one complete evolution trajectory contains 288 sampling steps (estimations are performed every five minutes), which is represented by a 2626×288 matrix (the network has 2626 links). As for the previous sections, the analysis is performed in 15-D NMF space (3-D space is only used for visualization purposes). Each trajectory is represented by a sequence of 288 network-level traffic state projected on the 15-D NMF space and denoted $\{h_1, h_2, \cdots, h_{288}\}$, where $h_i \in \mathbb{R}^{15}$. The similarity between trajectories $\{h_1^a, h_2^a, \ldots, h_{288}^a\}$ and $\{h_1^b, h_2^b, \ldots, h_{288}^b\}$, representing days a and b respectively, is computed as the sum over the different estimation times $k = 1 \ldots 288$ of *cosine distances* between the NMF projections at the corresponding estimation times:

$$D = \sum_{k=1}^{288} \text{cosdis}(h_k^a, h_k^b), \tag{9.3}$$

$$\text{where cosdis}(h_k^a, h_k^b) = 1 - \frac{h_k^a \cdot h_k^b}{\| h_k^a \| \, \| h_k^b \|}. \tag{9.4}$$

The function *cosdis* is the cosine distance between two vectors and is defined in (9.4). It evaluates the cosine value of the angle between the two data vectors h_k^a and h_k^a in the 15-D NMF projection space. Larger cosine distance values indicate more important differences between the two vectors. Due to the mathematical definition of the cosine function, the derived cosine distance is normalized into the range [0,1]. The defined measure between sequences of global traffic states is used to perform hierarchical clustering of daily global traffic states sequences in 15D-NMF space [130, 203]. The successive similarity-based groupings are shown on the dendrogram in Figure 9.4 (bottom) following the same color legends as in the middle figure. In the dendrogram, daily sequences of network-level traffic states are grouped gradually into clusters in the form of U-shaped trees. The height of each U-shaped tree (vertical axis) represents the distance between the sets of daily sequences being connected. Leaf nodes along the horizontal axis correspond to all daily sequences of network-level traffic states. At the bottom level of the hierarchical tree, daily sequences are first aggregated with respect to each day of the week. It underlines the intuition that each day of the week has a particular temporal dynamic pattern in terms of network-level traffic states. By increasing the distance thresholds, clusters merge until only one cluster remains. The seven days of the week are clustered into four different groups indicating the days that tend to follow similar patterns. Weekend days (Saturday and Sunday) are clustered together. As for the week days, Monday and Tuesday, representing the beginning of a week, appear to have a different temporal dynamic pattern from Wednesday and Thursday (middle of the week). Traffic dynamics on Fridays also tend to deviate slightly from that of the other days and is assigned to a separate group. As the distance threshold increases, Friday is added to the Wednesday and Thursday cluster. The results indicate the there are generally three kinds of temporal dynamic patterns of network-level traffic states in the data, corresponding to the beginning of the week (Monday and Tuesday), the end of the week (Wednesday, Thursday and Friday)

and the weekends (Saturday and Sunday). As the threshold increases even more, the two clusters of week days merge leading to two clusters representing the weekend days on one side and the week days on the other side. The distance thresholds need to be increased significantly more for these two clusters to merge, which indicates the importance in the differences in daily dynamics between week days and weekends. It is expected for Monday and Friday to have different dynamics (coming back or leaving for the week-end). However, it is slightly surprising that Monday and Tuesday are clustered together while Wednesday and Thursday (and then Friday) form another cluster. The data seems to indicate a beginning of the week vs. end of the week clustering, with Friday being the most different of the other days.

9.5 Conclusion and discussion

The chapter investigates how dimensionality reduction and more specifically *Non-negative Matrix Factorization* can be used to provide valuable insight regarding the large scale properties of a queuing network, both temporally and spatially. The data mining algorithm unveils spatial and temporal patterns which can be incorporated in existing estimation algorithms such as the *Dynamic Bayesian Networks* presented in Chapters 6 and 7. The integration of this information has the potential to both improve the accuracy and the scalability of the estimation algorithms. The principle is to perform dimensionality reduction, which allows for clustering of spatial congestion patterns, and easy analysis/categorization of temporal daily dynamics. Furthermore, the part-based decomposition feature of *Non-negative Matrix Factorization* automatically unveils areas of the road network with strong correlations.

Chapter 10

Conclusion

This dissertation presents modeling frameworks and estimation algorithms with a focus on systems for which a constitutive equation is available as a mathematical abstraction of the physical world. The dissertation investigates how a careful modeling of the physical world and insights provided by large amounts of data can improve the estimation capabilities of constitutive equations or data-driven estimation alone. It also discusses the choice of model complexity depending on the application of interest or the data available. The thesis takes the example of signalized queuing networks for which these considerations are essential and have not been studied extensively. The integration of physical and data insights has been acknowledged in numerous fields such as natural language processing, computer vision, weather forecasts but remains an emerging notion for which few contributions exist when considering signalized queuing networks.

Advantages and disadvantages of constitutive models

A constitutive model has the potential to accurately represent specific characteristics of a given system. For example, the fields of distributed parameter systems, estimation and control theory provide powerful frameworks to develop accurate estimation algorithms. In particular, Chapter 3 shows how the precise characteristics of time varying service rates in a queuing network can be estimated given a model of queue dynamics which is formulated as a *Hamilton Jacobi partial differential equation*. The potential of physical models to produce estimates of the state of a given system is also illustrated Chapter 5. The estimation capabilities of travel time distributions are significantly improved when first principles are used to model the dynamics of horizontal queues, instead of relying on classic distributions.

Besides a better representation of some of the characteristics of the system, models based on first principles also guarantee that the estimates are compatible with the physics of the problem. This property is all the more essential that little data is available. The constraints imposed by the physical model improve the robustness and the quality of the results. Chapter 5 underlines the importance of using a constitutive model of horizontal queues to represent delay distributions when little data is available. Chapter 6 illustrates how

the addition of a conservation law in the queuing network is able to capture the propagation of congestion in the network and provide estimation capabilities in regions of the network where little data is available as well as short term prediction capabilities.

However, the physical reality cannot be fully abstracted with a mathematical representation and assumptions on the physics are necessary to derive constitutive models. This may lead to limitations and inaccuracies when some of the assumptions of the modeling are not met. For example, Chapter 5 illustrates the limitation of some of the assumptions of the horizontal queuing model. Even though the distribution derived from this model outperforms classic distribution, the results show that one of the assumption is not always met in practice (uniform arrivals). The physical model may not be able to adapt automatically and detect that one of the assumptions is not valid. A more data-driven approach may help to assess the validity of different assumptions and automatically adapt to refine assumptions based on the available data. Similarly, a data-driven approach can help detect when a physical model tries to account for characteristics of the system which cannot be retrieved because of the granularity of the data.

The accurate representation of the physics may not be compatible with the desire to have computationally efficient models. For example, the distributions of delay and travel times derived in Chapter 5 are able to characterize accurately some specific characteristics of signalized queuing networks (in particular the difference between delayed and non-delayed customers). However, this comes to the detriment of the mathematical properties of the distributions. The thesis proves that the distributions have interesting mathematical properties (mixture of log-concave distributions) which are used to improve estimation capabilities. However, some essential properties of most classic distributions are not met by the distributions derived from queuing theory. In particular, the distributions are not convex in the parameters. This limitation is overcome by considering estimation problems which decouple for each link of the network (through travel time decomposition) and lead to a series of small scale optimization problems for which convexity properties are not as essential. However, additional assumptions or constraints are required to allow for this decoupling (independence of travel times and travel time allocation), whereas a model based on classic distributions (Chapter 7) do not require this additional step in the estimation algorithm.

Advantages and disadvantages of data-driven models

Data-driven models have the capability to model systems for which constitutive equations are not available. For example, the effect of individual behavior (*e.g.* driving behavior in transportation networks) or other exogenous features (*e.g.* pedestrians, bad parking or weather condition in transportation networks) are hard to model with constitutive equations. Instead, modeling their effect as a random variable has the potential to improve the robustness of the estimation and to accurately differentiate the signal and the noise, as done in Chapter 5. Similarly, a statistical approach can account for a lack of information on the system of interest. For example, Chapter 5 models the arrival time of customers in the queue (with respect to the service cycle) as a random variable. The signal timing is not

available and the arrival time of vehicles in the queue cannot be estimated directly (because the vehicles only report their position every minute rather than at the beginning and end of links). These two arguments motivate the modeling of the arrival time in the queue as a random variable, which can be integrated in a marginal distribution, instead of trying to estimate it.

Data-driven models also have the potential to study systems at a larger scale and to discover patterns which do not appear at a lower level. The dynamics of queuing networks is accurately modeled using partial differential equations but their large scale dynamics is not well understood today and no accurate constitutive model exists. The data-driven approach allows us to underline specific patterns and aspects of the dynamics which are difficult to model by a human. These aspects of the dynamics can be integrated in estimation algorithms (based on constitutive equations for example) to improve the estimation capabilities and/or simplify the model to make it more computationally efficient. For example, Chapter 8 detects the intersections with important variations in the traffic conditions. Such information can be integrated in the structure of the graphical models of Chapters 6 and 7 to model the propagation of congestion on the network.

The results of a data-driven approach may also be more reliable as they are based on the data rather than on arbitrary assumptions which are hard to validate. For example, Chapter 9 detects different *times of the day* in the congestion dynamics. The notion of *time of day* was first introduced, somewhat arbitrarily, in Chapters 6 and 7 for the necessity of the estimation. In these chapters, the times of day are chosen somewhat arbitrarily as a trade-off between the number of parameters to estimate and the amount of data available. Besides validating the intuition that congestion has specific characteristics depending on the time of the day, the clustering algorithm defines the beginning and the end of each *time of the day* based on the data. The definition of time of day by a human-being is more likely to be subjective, based on individual perception and experience. The data-driven approach is also more general as it does not require context specific knowledge and automatically detects the characteristics of each region. For example, the definition of the times of the day could be significantly different if run in a different city with different commute patterns). The data-driven approach also allows us to define regions of the graph with high level of correlations and regions of the graphs which have independent dynamics. A human being could model these dependencies using existing division of a city (ZIP codes, neighborhoods and so on) but the data-driven approach underlines a much more complex pattern. Again, the algorithm is not context specific and does not require a-priori knowledge of the area of interest. Similarly, the dendogram clustering similar trajectories and exhibiting similar days of the week can be leveraged in a hierarchical model without requiring someone to make arbitrary assumptions. Some people may cluster the days of the week as Saturday-Sunday for the weekend, Tuesday-Wednesday-Thursday for the mid-week and Monday-Friday for the "close to weekend" days. This clustering may be appropriate in some areas whereas others see different similarities between the days of the week.

Data-driven models tend to have better computational performance. They rely on existing distributions and algorithms which have been developed over the years and have nu-

merous mathematical properties which can be leveraged to optimize the computation. For example, Gaussian random variables have a long list of properties which make them very powerful in terms of computation (closed under convolution, closed under translation and linear transformation, convexity with respect to the parameters, and so on). These distributions may appear as a trade-off between the capacity to represent the reality accurately through a physical model and the capacity to run fast and robust estimation using data-driven models.

However, as pointed out earlier, data-driven models may be limited in their representation of the reality. Chapter 5 underlines this point. This limited ability to capture specific phenomena is all the more important that little data is available. Indeed, if large amounts of data are available, appropriately increasing the complexity of data-driven models with a large number of parameters or non-parametric models will be able to approximate the system accurately. For example, any continuous probability distribution can be approximated with a Gaussian Mixture Model with a sufficiently large number of components. The accurate representation of the physical system with data-driven models may be to the detriment of interpretability. For the distribution of travel times in Chapter 5 for example, the different components of the Gaussian Mixture Model may no longer represent identifiable patterns (*e.g.* delay patterns as in Chapter 5).

Data-driven models rely on a limited set of assumptions regarding the properties of the system. They have the potential to accurately represent any process as long as enough data is available.The trends of data collection today strongly encourage to have flexible models which improve as more data is collected.

Summary of contributions

As underlined in the previous paragraphs, the thesis points out the potentials and limitations of both physical and data-driven models. The thesis goes beyond stating the strengths and weaknesses of both approaches by presenting how a hybrid approach of physical and data-driven models can outperform either approach when considering the goal of building a scalable, robust, reliable and understandable estimation platform.

- Chapter 4 builds on the deterministic model and algorithm of Chapter 3. It shows how to integrate noisy measurements in the physical representation of a dynamical system. Instead of computing the deterministic solution of the *Hamilton-Jacobi* partial differential equation, the algorithm provides results in terms of a probability distribution function. This expands the estimation capabilities and make them more robust to inaccuracies of both the data and the modeling.

- Chapter 5 takes into account another form of variability which has a very different origin from the variability taken into account in Chapter 4. In this case some of the randomness accounts for information which cannot be reconstructed from the data. The level of details provided by sparsely sampled probe vehicles does not enable the

reconstruction of precise signal timing nor arrival time at the beginning of the link. For this reason, the arrival time with respect to the beginning of the cycle (sometimes referred to as *offset*) is considered as a random variable. The distribution of delay is then computed as a marginal distribution, by integrating the delay distribution for each arrival time within a cycle. This model represents a higher level of abstraction and aggregation compared to the derivations of Chapters 3 and 4. This abstraction is possible by leveraging physical properties of the system (periodicity of signalized networks) even though the original constitutive equation does not appear as directly in the estimation algorithm of Chapter 5 as it does in Chapters 3 and 4.

- Chapter 5 also leverages a hybrid approach of physical and data-driven models to take into account the variability of travel times due to driver behavior as well as other factors such as pedestrians, bad parking, lane changes and so on. These different factors are hard to model accurately using physical models. However, the physics provides the intuition that they should be taken into account. The effect of these factors on the distribution of travel times can thus be learned from the data.

- Chapter 6 goes one step further in the integration of physical and data-driven models. The chapter builds on Chapter 5 to represent the probability distribution of travel times. It adds a physical property of the system (conservation of flows at intersections) while improving the learning capabilities of the model through a data-driven approach (*Dynamic Bayesian Network*). The model is then able to account for the propagation of congestion in the network and perform short term prediction, in coherence with both the physics and the data.

- Chapter 7 is comparable to Chapter 6 in many aspects. They both represent the dynamics of signalized queuing networks using a *Dynamic Bayesian Network*. They differ in the level of abstraction from the physics that each of them adopts. In Chapter 6, both the observation and the dynamical model are derived from first principles of horizontal queues and flow conservation. Chapter 7 has the same structure for the *Dynamic Bayesian Network* but the observation and dynamical model are more general, requiring limited assumption on the physics. In the model of Chapter 7, the number of states for each link can be as large as desired. Similarly, the structure of the transition model is general. The thesis presents the model with assumptions on the transition model: there is only edges in the *Dynamic Bayesian Network* between the state of a link at a given time interval and the state of its physical neighbors at the following time interval. This could be generalized to allow for any structure of the transition probability, *i.e.* allowing edges between the state of any link at a given time interval with the state of any other link at any time interval. Any transition model is theoretically possible and not limited by the framework of Dynamic Bayesian Networks. The choice of the number of states for each link and the choice of the structure for the transition model defines the model complexity. The choice depends on two factors: (i) a more complex model has the potential to represent the system more accurately but

is more likely to over-fit the training data and (ii) a more complex model is likely to be more computationally intensive and the chosen level of complexity may depend on requirements regarding scalability, real-time capabilities and so on. Chapter 7 shows the importance to leverage ideas from the physics while letting the model have as much flexibility as possible so that the data can improve the model.

- Chapters 8 and 9 present data-mining algorithms which provide insights to improve models derived from the physics, such as the *Dynamic Bayesian Networks* of Chapters 6 and 7. The algorithms operate at a higher scale and are not able to retrieve (nor do they aim at retrieving) as precise information as the estimation algorithms presented in the previous chapters. However, the data-driven approaches have the potential to improve some of the modeling and algorithm design decisions by providing a large-scale understanding of the dynamics which is hard to model directly. The data-driven models also have the advantage to be general. They can be applied to a large variety of systems without requiring site-specific or application-specific knowledge. For example, the algorithms have a lot of potential to improve the algorithms in the following ways.

  - The LASSO algorithm is able to detect change in each queue. This information can be integrated in the estimation algorithm to decide when old data should be discarded because conditions have changed significantly.
  - The LASSO algorithm analyzes local differences in traffic conditions. The outcome of the algorithms could improve and make less arbitrary the design of transition probabilities in the *Dynamic Bayesian Networks* of Chapters 6 and 7.
  - The *Non-negative Matrix Factorization* algorithm clusters the graph in different area with high intraclass correlation and low interclass correlation. This type of analysis is essential to scale estimation algorithms such as the *Dynamic Bayesian Networks* of Chapters 6 and 7. Indeed, inference in these graphs is computationally demanding. Approximations using particle filters require the number of particles to grow. A partition of the graph in areas with high intraclass correlation and low interclass correlation may face this limitation by using approximate inference algorithms (such as the *Boyen-Koller* algorithm).
  - The estimation algorithms usually define *times of the day* during which the parameters of the dynamics are assumed to be constant. This is important to limit the number of parameters to estimate and avoid over-fitting. However, the use of *times of the day* requires the choice of the beginning and end of each of these time periods. The *Non-negative Matrix Factorization* algorithm provides this information. The definition of the *times of the day* is based solely on the data and does not require specific knowledge or subjective modeling assumptions which makes it more general and less prone to human judgment.
  - Estimation algorithms can usually be improved by considering hierarchical models. The ides of hierarchical model is to improve the robustness of the model

when little data is available using Bayesian statistics. For example, in the graphical model of Chapters 6 and 7, the parameters are defined by *time of the day* and then by day of the week. Considering that some parameters are drawn from the same distribution may improve the robustness of the estimation. The *Non-negative Matrix Factorization* algorithm provides a dendogram representing the similarities between the days of the week. This information could be incorporated in the design of the graphical models of Chapters 6 and 7 without site-specific or application-specific knowledge. The hierarchy exhibited in Chapter 9 in the Bay Area of San Francisco, CA could be very different in other areas of the world. For example, some regions of the world have weekends on Thursday and Friday. Others may have very different mobility patterns (commuting times shifted early, commute through home at lunch time and so on).

Bibliography

[1] Cabspotting, http://www.cabspotting.org.

[2] NAVTEQ Inc., http://www.navteq.com.

[3] Sensys Networks, http://www.sensysnetworks.com.

[4] A. Bayen, J. Butler, A. Patire. et al, Mobile Millennium Final Report, Tech. rep., University of California, Berkeley, UCB-ITS-CWP-2011-6, 2011.

[5] R. E. Allsop, *An analysis of delays to vehicle platoons at traffic signals*, University College, 1968.

[6] R. E. Allsop, Delay at fixed Time Traffic Signals-I: Theoretical analysis, in *Transportation Science*, vol. 6, pp. 280–285, 1972.

[7] L. Alvarez-Icaza, L. Munoz, X. Sun and R. Horowitz, Adaptive observer for traffic density estimation, in *American Control Conference*, vol. 3, pp. 2705–2710, IEEE, 2004.

[8] A. Ambrosetti and G. Prodi, *A primer of nonlinear analysis*, 34, Cambridge University Press, 1995.

[9] M. Arulampalam, S. Maskell, N. Gordon and T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, in *IEEE Transactions on Signal Processing*, vol. 50 (2), pp. 174–188, 2002.

[10] A. Aswani and C. Tomlin, Game-theoretic routing of GPS-assisted vehicles for energy efficiency, in *American Control Conference*, pp. 3375–3380, IEEE, 2011.

[11] J. Aubin, A. Bayen and P. Saint-Pierre, Dirichlet Problems for some Hamilton–Jacobi equations with inequality constraints, in *SIAM Journal on Control and Optimization*, vol. 47 (5), pp. 2348–2380, 2008.

[12] J. Aubin, A. Bayen and P. Saint-Pierre, *Viability Theory: New Directions, 2nd edition*, Springer, 2011.

[13] A. Aw and M. Rascle, Resurrection of 'Second Order' Models of Traffic Flow, in *SIAM Journal on Applied Mathematics*, vol. 60 (3), pp. 916–938, 2000.

[14] F. Bach, R. Jenatton, J. Mairal and G. Obozinski, Convex optimization with sparsity-inducing norms, in *Optimization for Machine Learning*, pp. 19–53, 2011.

[15] C. Bails, A. Hofleitner, Y. Xuan and A. Bayen, A Three-Stream Model for Arterial Traffic, in *91st Transportation Research Board Annual Meeting*, 12-1212, Washington, D.C., 2012.

[16] X. J. Ban, R. Herring, P. Hao and A. M. Bayen, Delay pattern estimation for signalized intersections using sampled travel times, in *Transportation Research Record*, vol. 2130 (1), pp. 109–119, 2009.

[17] H. T. Banks and K. Kunisch, *Estimation Techniques for Distributed Parameter Systems*, Birkhauser Boston, 1989.

[18] J. Banks, Freeway speed-flow-concentration relationships: more evidence and interpretations (with discussion and closure), in *Transportation Research Record*, vol. 1225, 1989.

[19] R. Baraniuk, Compressive sensing, in *Signal Processing Magazine*, vol. 24 (4), p. 118, 2007.

[20] M. Bardi and I. Capuzzo-Dolcetta, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*, Springer, 2008.

[21] E. Barron and R. Jensen, Semicontinuous viscosity solutions for Hamilton–Jacobi equations with convex Hamiltonians, in *Communications in Partial Differential Equations*, vol. 15 (12), pp. 293–309, 1990.

[22] M. G. Bell and Y. Iida, *Transportation network analysis*, Wiley, 1997.

[23] A. Bennett, *Inverse methods in physical oceanography*, Cambridge university press, 1992.

[24] D. P. Bertsekas, Incremental gradient, subgradient, and proximal methods for convex optimization: A survey, in *Optimization for Machine Learning*, p. 85, 2011.

[25] P. Bickel, C. Chen, J. Kwon, J. Rice, E. V. Zwet and P. Varaiya, Measuring Traffic, in *Statistical Science*, vol. 22 (4), pp. 581–597, 2007.

[26] S. Blandin, A. Couque, A. Bayen and D. Work, On sequential data assimilation for scalar macroscopic traffic flow models, in *Physica D: Nonlinear Phenomena*, vol. 241 (17), pp. 1421–1440, 2012.

[27] S. Blandin, D. Work, P. Goatin, B. Piccoli and A. Bayen, A General Phase Transition Model for Vehicular Traffic, in *SIAM Journal on Applied Mathematics*, vol. 71 (1), pp. 107–127, 2011.

[28] J. P. Boyd, *Chebyshev and Fourier spectral methods*, Dover publications, 2001.

[29] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, 2004.

[30] X. Boyen and D. Koller, Tractable inference for complex stochastic processes, in *14th Conference on Uncertainty in Artificial Intelligence*, pp. 33–42, 1998.

[31] R. Bracewell, *The fourier transform & its applications 3rd Ed*, McGraw-Hill Science/Engineering/Math, 2000.

[32] A. M. Bratseth, Statistical interpolation by means of successive corrections, in *Tellus A*, vol. 38 (5), pp. 439–447, 1986.

[33] A. Bressan, *Hyperbolic Systems of Conservation Laws: The One-Dimensional Cuachy Problem*, vol. 20, Oxford University Press, USA, 2000.

[34] D. Cai, X. He, X. Wu and J. Han, Non-negative Matrix Factorization on Manifold, in *International Conference on Data Mining*, pp. 63–72, IEEE, 2008.

[35] G. Cameron and G. Duncan, PARAMICS - Parallel microscopic simulation of road traffic, in *Journal of Supercomputing*, vol. 10 (1), pp. 25–53, 1996.

[36] E. Candès, Compressive sampling, in *Congress of Mathematicians*, vol. 3, pp. 1433–1452, 2006.

[37] K. Chelst and J. Jarvis, Estimating the Probability Distribution of Travel Times for Urban Emergency Service Systems, in *Operations Research*, vol. 27 (1), pp. 199–204, 1979.

[38] P. Cheng, Z. Qiu and B. Ran, Particle filter based traffic state estimation using cell phone network data, in *9th Intelligent Transportation Systems Conference*, pp. 1047 –1052, 2006.

[39] N. Chiabaut, L. Leclercq and C. Buisson, From heterogeneous drivers to macroscopic patterns in congestion, in *Transportation Research Part B*, vol. 44 (2), pp. 299–308, 2010.

[40] G. Christodoulou and E. Koutsoupias, The price of anarchy of finite congestion games, in *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pp. 67–73, ACM, 2005.

[41] C. Claudel and A. Bayen, Lax-Hopf based incorporation of internal boundary conditions into Hamilton-Jacobi equation. Part I: theory, in *IEEE Transactions on Automatic Control*, vol. 55 (5), pp. 1142–1157, 2010.

[42] C. Claudel and A. Bayen, Lax-Hopf based incorporation of internal boundary conditions into Hamilton-Jacobi equation. Part II: Computational methods, in *IEEE Transactions on Automatic Control*, vol. 55 (5), pp. 1158–1174, 2010.

[43] C. Claudel and A. Bayen, Convex Formulations of Data Assimilation Problems for a Class of Hamilton-Jacobi Equations, in *SIAM Journal on Control and Optimization*, vol. 49 (2), pp. 383–402, 2011.

[44] C. Claudel, A. Hofleitner, N. Mignerey and A. Bayen, Guaranteed bounds on highway travel times using probe and fixed data, in *88th Transportation Research Board Annual Meeting*, 2009.

[45] R. M. Colombo, P. Goatin and F. S. Priuli, Global well posedness of traffic flow models with phase transitions, in *Nonlinear Analysis: Theory, Methods & Applications*, vol. 66 (11), pp. 2413–2426, 2007.

[46] G. Comert and M. Cetin, Analytical Evaluation of the Error in Queue Length Estimation at Traffic Signals From Probe Vehicle Data, in *IEEE Transactions on Intelligent Transportation Systems*, vol. 12 (2), pp. 563–573, 2011.

[47] G. Cooper, The computational complexity of probabilistic inference using bayesian belief networks, in *Artificial Intelligence*, vol. 42 (2-3), pp. 393–405, 1990.

[48] R. B. Cooper, *Introduction to Queuing Theory, 2nd Ed.*, North Holland Publishing Company, 1981.

[49] M. Crandall, L. Evans and P. Lions, Some properties of viscosity solutions of Hamilton-Jacobi equations, in *Transactions of the American Mathematical Society*, vol. 282 (2), pp. 487–502, 1984.

[50] M. Crandall and P. Lions, Viscosity solutions of Hamilton-Jacobi equations, in *Transactions of the American Mathematical Society*, vol. 277 (1), pp. 1–42, 1983.

[51] C. M. Dafermos, Polygonal approximations of solutions of the initial value problem for a conservation law, in *Journal of Mathematical Analysis and Applications*, vol. 38 (33), p. 41, 1972.

[52] C. Daganzo, The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory, in *Transportation Research Part B*, vol. 28 (4), pp. 269 – 287, 1994.

[53] C. Daganzo, The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory, in *Transportation Research Part B*, vol. 28 (4), pp. 269 – 287, 1994.

[54] C. Daganzo, The cell transmission model, part II: network traffic, in *Transportation Research Part B*, vol. 29 (2), pp. 79–93, 1995.

[55] C. Daganzo, Requiem for second-order fluid approximations of traffic flow, in *Transportation Research Part B*, vol. 29 (4), pp. 277–286, 1995.

[56] C. Daganzo, A variational formulation of kinematic waves: basic theory and complex boundary conditions, in *Transportation Research Part B*, vol. 39 (2), pp. 187–196, 2005.

[57] C. Daganzo, On the variational theory of traffic flow: well-posedness, duality and applications, in *Networks and Heterogeneous Media*, vol. 1 (4), p. 601, 2006.

[58] C. Daganzo and N. Geroliminis, An analytical approximation for the macroscopic fundamental diagram of urban traffic, in *Transportation Research Part B*, vol. 42 (9), pp. 771–781, 2008.

[59] R. D'Andrea and G. E. Dullerud, Distributed control design for spatially interconnected systems, in *IEEE Transactions on Automatic Control*, vol. 48 (9), pp. 1478–1495, 2003.

[60] I. Daubechies, M. Defrise and C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, in *Communications on Pure and Applied Mathematics*, vol. 57 (11), pp. 1413–1457, 2004.

[61] T. Dean and K. Kanazawa, A model for reasoning about persistence and causation, in *Computational Intelligence*, vol. 5 (2), pp. 142–150, 1989.

[62] A. Dempster, N. Laird and D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, in *Journal of the Royal Statistical Society: Series B*, pp. 1–38, 1977.

[63] Y. Dodge, *Statistical data analysis based on the l_1-norm and related methods*, Birkhauser, 2002.

[64] A. Doucet and N. de Freitas, *Sequential Monte Carlo methods in practice*, Springer, 2001.

[65] I. Drori and D. Donoho, Solution of l_1 Minimization Problems by LARS/Homotopy Methods, in *International Conference on Acoustics, Speech and Signal Processing*, vol. 3, IEEE, 2006.

[66] B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, Least Angle Regression, in *Annals of Statistics*, vol. 32 (2), pp. 407–451, 2004.

[67] N. El Karoui, S. Peng and M. C. Quenez, Backward Stochastic Differential Equations in Finance, in *Mathematical Finance*, vol. 7 (1), pp. 1–71, 1997.

[68] A. K. Erlang, The theory of probabilities and telephone conversations, in *Nyt Tidsskrift for Matematik B*, vol. 20 (6), pp. 87–98, 1909.

[69] A. K. Erlang, Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges, in *The Post Office Electrical Engineers Journal*, vol. 10, pp. 189–197, 1917.

[70] D. L. Estrin, Participatory sensing: applications and architecture, in *8th international conference on Mobile systems, applications, and services*, pp. 3–4, ACM, 2010.

[71] L. C. Evans, *Partial Differential Equations*, vol. 19 of *Graduate Studies in Mathematics*, American Mathematical Society, Providence, RI, 1998.

[72] G. Evensen, Using the extended Kalman filter with a multilayer quasi-geostrophic ocean model, in *Journal of Geophysical Research*, vol. 97 (C11), pp. 17905–17924, 1992.

[73] G. Evensen, Advanced data assimilation for strongly nonlinear dynamics, in *Monthly Weather Review*, vol. 125 (6), pp. 1342–1354, 1997.

[74] G. Evensen, *Data assimilation: the ensemble Kalman filter*, Springer, 2009.

[75] M. Fellendorf, VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority, in *64th Institute of Transportation Engineers Annual Meeting*, 1994.

[76] T. Feng, S. Li, H. Shum and H. Zhang, Local Nonnegative Matrix Factorization as a Visual Representation, in *2nd International Conference On Development and Learning*, pp. 178–183, IEEE, 2002.

[77] M. Figueiredo and R. Nowak, A bound optimization approach to wavelet-based image deconvolution, in *International Conference on Image Processing*, vol. 2, IEEE, 2005.

[78] M. Figueiredo, R. Nowak and S. Wright, Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, in *Journal of Selected Topics in Signal Processing*, vol. 1 (4), pp. 586–597, 2008.

[79] H. Frankowska, Lower semicontinuous solutions of Hamilton-Jacobi-Bellman equations, in *SIAM Journal on Control and Optimization*, vol. 31 (1), pp. 257–272, 1993.

[80] J. Friedman, T. Hastie, H. Höfling and R. Tibshirani, Pathwise coordinate optimization, in *Annals of Applied Statistics*, vol. 1 (2), pp. 302–332, 2007.

[81] J. Fuchs, On sparse representations in arbitrary redundant bases, in *IEEE Transactions on Information Theory*, vol. 50 (6), pp. 1341–1344, 2004.

[82] C. Furtlehner, J. Lasgouttes and A. de la Fortelle, A belief propagation approach to traffic prediction using probe vehicles, in *10th Intelligent Transportation Systems Conference*, pp. 1022–1027, IEEE, 2007.

[83] M. Garavello and B. Piccoli, *Traffic flow on networks*, vol. 1, American Institute of Mathematical Sciences, 2006.

[84] P. Garrigues and L. El Ghaoui, An homotopy algorithm for the Lasso with online observations, in *Advances on Neural Information Processing Systems*, vol. 21, 2008.

[85] D. L. Gerlough and M. J. Huber, Traffic flow theory, in *Transportation Research Board Annual Meeting*, pp. 199–201, Trans Res Board, 1975.

[86] N. Geroliminis and C. Daganzo, Macroscopic modeling of traffic in cities, in *86th Transportation Research Board Annual Meeting*, 07-0413, Washington, D.C., 2007.

[87] N. Geroliminis and A. Skabardonis, Prediction of arrival profiles and queue lengths along signalized arterials by using a Markov decision process, in *Transportation Research Record*, vol. 1934 (1), pp. 116–124, 2006.

[88] N. Geroliminis and A. Skabardonis, Queue spillovers in city street networks with signal-controlled Intersections, in *89th Transportation Research Board Annual Meeting*, Washington, D.C., 2010.

[89] P. Gipps, A behavioural car-following model for computer simulation, in *Transportation Research Part B*, vol. 15 (2), pp. 105–111, 1981.

[90] S. K. Godunov, A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics, in *Matematicheskii Sbornik*, vol. 89 (3), pp. 271–306, 1959.

[91] G. Golub and C. Van Loan, *Matrix computations*, vol. 3, Johns Hopkins Univ Press, 1996.

[92] N. Gordon, D. Salmond and A. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, in *IEE Proceedings F, Radar and Signal Processing*, vol. 140 (2), pp. 107–113, 1993.

[93] D. Gross, J. F. Shortle, J. M. Thompson and C. M. Harris, *Fundamentals of queueing theory*, vol. 627, Wiley-Interscience, 2011.

[94] I. Guyon and A. Elisseeff, An introduction to variable and feature selection, in *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[95] A. Halati, H. Lieu and S. Walker, CORSIM-corridor traffic simulation model, in *Traffic Congestion and Traffic Safety in the 21st Century: Challenges, Innovations, and Opportunities*, 1997.

[96] Y. Han and F. Moutarde, Analysis of Network-level Traffic States using Locality Preservative Non-negative Matrix Factorization, in *14th Intelligent Transport Systems Conference*, IEEE, 2011.

[97] Y. Han and F. Moutarde, Clustering and modeling of Network-level Traffic States based on Locality Preservative Non-negative Matrix Factorization, in *8th Intelligent Transport Systems (ITS) European Congress*, 2011.

[98] A. Haoui, R. Kavaler and P. Varaiya, Wireless magnetic sensors for traffic surveillance, in *Transportation Research Part C*, vol. 16 (3), pp. 294 – 306, 2008.

[99] X. He and P. Niyogi, Locality Preserving Projections, in *Advances in Neural Information Processing Systems*, vol. 16, 2003.

[100] D. Helbing, Improved fluid-dynamic model for vehicular traffic, in *Physical Review E*, vol. 51 (4), p. 3164, 1995.

[101] B. Hellinga, P. Izadpanah, H. Takada and L. Fu, Decomposing travel times measured by probe-based traffic monitoring systems to individual road segments, in *Transportation Research Part C*, vol. 16 (6), pp. 768 – 782, 2008.

[102] J. Herrera, D. Work, R. Herring, X. Ban, Q. Jacobson and A. Bayen, Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment, in *Transportation Research Part C*, vol. 18 (4), pp. 568–583, 2010.

[103] J. C. Herrera and A. M. Bayen, Incorporation of Lagrangian measurements in freeway traffic state estimation, in *Transportation Research Part B: Methodological*, vol. 44 (4), pp. 460–481, 2010.

[104] R. Herring, *Real-Time Traffic Modeling and Estimation with Streaming Probe Data using Machine Learning*, Ph.D. thesis, University of California, Berkeley, 2010.

[105] R. Herring, A. Hofleitner, P. Abbeel and A. Bayen, Estimating arterial traffic conditions using sparse probe data, in *13th Intelligent Transportation Systems Conference*, pp. 929–936, IEEE, Madeira, Portugal, 2010.

[106] R. Herring, A. Hofleitner, S. Amin, T. A. Nasr, A. Abdel Khalek, P. Abbeel and A. Bayen, Using Mobile Phones to Forecast Arterial Traffic through Statistical Learning, in *89th Transportation Research Board Annual Meeting*, 10-2493, Washington, D.C., 2010.

[107] A. Hofleitner and A. Bayen, Optimal decomposition of travel times measured by probe vehicles using a statistical traffic flow model, in *14th Intelligent Transportation Systems Conference*, pp. 815 –821, IEEE, 2011.

[108] A. Hofleitner, C. Claudel and A. Bayen, Probabilistic formulation of estimation problems for a class of Hamilton-Jacobi equations, in *51st Conference on Decision and Control*, pp. 3531–3537, IEEE, 2012.

[109] A. Hofleitner, C. Claudel and A. Bayen, Reconstruction of boundary conditions from internal conditions using viability theory, in *American Control Conference (ACC), 2012*, pp. 640–645, IEEE, 2012.

[110] A. Hofleitner, E. Come, L. Oukhellou, J.-P. Lebacque and A. Bayen, Automatic inference of map attributes from mobile data, in *15th Intelligent Transportation Systems Conference*, pp. 1687–1692, IEEE, Anchorage, Alaska, 2012.

[111] A. Hofleitner, L. E. Ghaoui and A. Bayen, Online least-squares estimation of time varying systems with sparse temporal evolution and application to traffic estimation, in *50th Conference on Decision and Control and European Control Conference*, pp. 2595–2601, IEEE, 2011.

[112] A. Hofleitner, R. Herring, P. Abbeel and A. Bayen, Learning the dynamics of arterial traffic from probe data using a Dynamic Bayesian Network, in *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, pp. 1679–1693, 2012.

[113] A. Hofleitner, R. Herring and A. Bayen, Arterial travel time forecast with streaming data: a hybrid flow model - machine learning approach, in *Transportation Research Part B*, vol. 46 (9), pp. 1097–1122, 2012.

[114] A. Hofleitner, R. Herring and A. Bayen, Probability distributions of travel times on arterial networks: a traffic flow and horizontal queuing theory approach, in *91st Transportation Research Board Annual Meeting*, 12-0798, Washington, D.C., 2012.

[115] A. Hofleitner, R. Herring, A. Bayen, Y. Han, F. Moutarde and A. de La Fortelle, Large scale estimation of arterial traffic and structural analysis of traffic patterns using probe vehicles, in *91st Transportation Research Board Annual Meeting*, 12-0598, Washington, D.C., 2012.

[116] R. Horst and H. Tuy, Global optimization: deterministic approaches, in *Journal of the Operational Research Society*, vol. 45 (5), pp. 595–596, 1994.

[117] E. Horvitz, J. Apacible, R. Sarin and L. Liao, Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service, in *21st Conference on Uncertainty in Artificial Intelligence*, 2005.

[118] P. L. Houtekamer and H. L. Mitchell, A sequential ensemble Kalman filter for atmospheric data assimilation, in *Monthly Weather Review*, vol. 129 (1), pp. 123–137, 2001.

[119] P. Hoyer, Non-negative Matrix Factorization with Sparseness Constraints, in *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[120] T. Hunter, P. Abbeel and A. M. Bayen, The Path Inference Filter: Model-Based Low-Latency Map Matching of Probe Vehicle Data, in *Algorithmic Foundations of Robotics X*, Springer Tracts in Advanced Robotics, pp. 591:1–607:17, Springer-Verlag, 2012.

[121] T. Hunter, A. Hofleitner, J. Reilly, W. Krichene, J. Thai, A. Kouvelas, P. Abbeel and A. Bayen, Arriving on time: estimating travel time distributions on large-scale road networks, in *arXiv preprint arXiv:1302.6617*, 2013.

[122] C. Imbert, R. Monneau and H. Zidani, A Hamilton-Jacobi approach to junction problems and application to traffic flows, in *arXiv preprint arXiv:1107.3250*, 2011.

[123] V. Isakov, *Inverse problems for partial differential equations*, vol. 127, Springer Verlag, 2006.

[124] Z. Jia, C. Chen, B. Coifman and P. Varaiya, PeMS algorithms for accurate, real-time estimates of g-factors and speeds from single-loop detectors, in *4th Intelligent Transportation Systems Conference*, pp. 536–541, IEEE, 2001.

[125] R. Jiang, Q. Wu and Z. Zhu, Full velocity difference model for a car-following theory, in *Physical Review E*, vol. 64 (1), p. 017101, 2001.

[126] I. Jolliffe, *Principal component analysis*, Wiley & Sons, 2005.

[127] M. I. Jordan, ed., *Learning in graphical models*, MIT Press, Cambridge, MA, USA, 1999.

[128] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola and L. K. Saul, An introduction to variational methods for graphical models, in *Machine learning*, vol. 37 (2), pp. 183–233, 1999.

[129] R. E. Kalman, A new approach to linear filtering and prediction problems, in *Journal of basic Engineering*, vol. 82 (1), pp. 35–45, 1960.

[130] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko and A. Wu, An efficient k-means clustering algorithm: Analysis and implementation, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 881–892, 2002.

[131] C. Y. Kao, S. Osher and J. Qian, Lax–Friedrichs sweeping scheme for static Hamilton–Jacobi equations, in *Journal of Computational Physics*, vol. 196 (1), pp. 367–391, 2004.

[132] E. L. Kaplan and P. Meier, Nonparametric estimation from incomplete observations, in *Journal of the American statistical association*, vol. 53 (282), pp. 457–481, 1958.

[133] D. G. Kendall, Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain, in *Annals of Mathematical Statistics*, pp. 338–354, 1953.

[134] A. Y. Khinchin, The Mathematical Theory of a Stationary Queue, in *Matematicheskii Sbornik*, vol. 39 (4), pp. 73–84, 1967.

[135] S. Kim, K. Koh, S. Boyd and D. Gorinevsky, l_1 trend filtering, in *SIAM Reviews*, vol. 51 (2), pp. 339–360, 2009.

[136] S. Kim, K. Koh, M. Lustig, S. Boyd and D. Gorinevsky, An Interior-Point Method for Large-Scale l_1-Regularized Least Squares, in *Journal of Selected Topics in Signal Processing*, vol. 1 (4), pp. 606–617, 2008.

[137] R. M. Kimber and E. M. Hollis, *Traffic queues and delays at road junctions*, Traffic Systems Division, Traffic Engineering Department, Transport and Road Research Laboratory, 1979.

[138] G. Kitagawa, Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, in *Journal of Computational and Graphical statistics*, vol. 5 (1), pp. 1–25, 1996.

[139] L. Kleinrock, *Queueing Systems. Volume 1: Theory*, Wiley-interscience, 1975.

[140] K. Knight and W. Fu, Asymptotics for lasso-type estimators, in *Annals of Statistics*, pp. 1356–1378, 2000.

[141] W. Knospe, L. Santen, A. Schadschneider and M. Schreckenberg, Human behavior as origin of traffic phases, in *Physical Review E*, vol. 65 (1), p. 015101, 2001.

[142] R. Kohavi et al., A study of cross-validation and bootstrap for accuracy estimation and model selection, in *International joint Conference on artificial intelligence*, vol. 14, pp. 1137–1145, Lawrence Erlbaum Associates Ltd, 1995.

[143] Y. A. Korilis, A. A. Lazar and A. Orda, Capacity allocation under noncooperative routing, in *IEEE Transactions on Automatic Control*, vol. 42 (3), pp. 309–325, 1997.

[144] A. Krause, E. Horvitz, A. Kansal and F. Zhao, Toward Community Sensing, in *7th Conference on Information processing in sensor networks*, pp. 481–492, IEEE Computer Society, 2008.

[145] W. Krichene, J. Reilly, S. Amin and A. Bayen, On Stackelberg routing on parallel networks with horizontal queues, in *51st IEEE Conference on Decision and Control*, pp. 7126–7132, IEEE, 2012.

[146] W. Krichene, J. Reilly, S. Amin and A. Bayen, On the characterization and computation of Nash equilibria on parallel networks with horizontal queues, in *51st IEEE Conference on Decision and Control*, pp. 7119–7125, IEEE, 2012.

[147] K. Kwong, R. Kavaler, R. Rajagopal and P. Varaiya, Arterial travel time estimation based on vehicle re-identification using wireless magnetic sensors, in *Transportation Research Part C*, vol. 17 (6), pp. 586–606, 2009.

[148] P. D. Lax, Hyperbolic systems of conservation laws II, in *Communications on Pure and Applied Mathematics*, vol. 10 (4), pp. 537–566, 2006.

[149] J.-P. Lebacque, S. Mammar and H. H. Salem, Generic second order traffic flow modelling, in *Transportation and Traffic Theory 2007. Papers Selected for Presentation at ISTTT17*, 2007.

[150] D. Lee and H. Seung, Algorithms for Non-negative Matrix Factorization, in *Advances in Neural Information Processing Systems*, vol. 13, pp. 556–562, 2000.

[151] H. Lee, A. Battle, R. Raina and A. Ng, Efficient sparse coding algorithms, in *Advances in Neural Information Processing Systems*, vol. 19, p. 801, 2007.

[152] J. S. V. Leeuwaarden, Delay analysis for the fixed-cycle traffic-light queue, in *Transportation Science*, vol. 40 (2), pp. 189–199, 2006.

[153] R. J. LeVeque, *Numerical methods for conservation laws*, Birkhäuser, 1992.

[154] J. Lewis, S. Lakshmivarahan and S. Dhall, *Dynamic data assimilation: a least squares approach*, vol. 104, Cambridge Univ Pr, 2006.

[155] M. Lighthill and G. Whitham, On Kinematic Waves. II. A theory of traffic flow on long crowded roads, in , vol. 229 (1178), pp. 317–345, 1955.

[156] C. Lin, Projected Gradient Methods for Non-negative Matrix Factorization, in *Neural Computation*, vol. 19 (10), pp. 2756–2779, 2007.

[157] H. Liu, H. van Zuylen, H. van Lint and M. Salomons, Predicting Urban Arterial Travel Time with state-space neural networks and Kalman filters, in *Transportation Research Record*, vol. 1968, pp. 99–108, 2006.

[158] I. Loris, On the performance of algorithms for the minimization of l_1-penalized functionals, in *Inverse Problems*, vol. 25 (3), pp. 35008–35023, 2009.

[159] J. MacQueen, Some methods for classification and analysis of multivariate observations, in *5th Berkeley symposium on mathematical statistics and probability*, pp. 281–297, 1967.

[160] H. Mahmassani and R. Herman, Dynamic user equilibrium departure time and route choice on idealized traffic arterials, in *Transportation Science*, vol. 18 (4), pp. 362–384, 1984.

[161] J. Mairal and B. Yu, Complexity Analysis of the Lasso Regularization Path, in *International Conference on Machine Learning*, 2012.

[162] F. J. Massey, The Kolmogorov-Smirnov test for goodness of fit, in *Journal of the American statistical association*, vol. 46 (253), pp. 68–78, 1951.

[163] P.-E. Mazaré, A. Dehwah, C. Claudel and A. Bayen, Analytical and grid-free solutions to the LighthillWhithamRichards traffic flow model, in *Transportation Research Part B*, vol. 45 (10), pp. 1727 – 1748, 2011.

[164] G. Meyers, H. Phillips, N. Smith and J. Sprintall, Space and time scales for optimal interpolation of temperature?Tropical Pacific Ocean, in *Progress in Oceanography*, vol. 28 (3), pp. 189–218, 1991.

[165] P. Michalopoulos and V. Pisharody, Derivation of delays based on improved macroscopic traffic models, in *Transportation Research Part B*, vol. 15 (5), pp. 299–317, 1981.

[166] X. Min, J. Hu, Q. Chen, T. Zhang and Y. Zhang, Short-term traffic flow forecasting of urban network based on dynamic STARIMA model, in *12th International Conference on Intelligent Transportation Systems*, pp. 1–6, IEEE, 2009.

[167] I. Mitchell, A. Bayen and C. Tomlin, A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games, in *IEEE Transactions on Automatic Control*, vol. 50 (7), pp. 947–957, 2005.

[168] I. Mitchell and C. Tomlin, Level set methods for computation in hybrid systems, in *Hybrid Systems: Computation and Control*, pp. 310–323, 2000.

[169] K. Moskowitz and L. Newan, Notes on freeway capacity, in *Highway Research Record*, 1963.

[170] K. Murphy, *Dynamic bayesian networks: representation, inference and learning*, Ph.D. thesis, University of California, Berkeley, 2002.

[171] R. M. Neal and al., Probabilistic inference using Markov chain Monte Carlo methods, Tech. Rep. CRG-TR-93-1, University of Toronto. Department of Computer Science, 1993.

[172] G. Newell, A simplified theory of kinematic waves in highway traffic, in *Transportation Research Part B*, vol. 27 (4), pp. 281–313, 1993.

[173] A. Ng, Feature selection, l_1 vs. l_2 regularization, and rotational invariance, in *21st International Conference on Machine learning*, p. 78, ACM, 2004.

[174] Next Generation Simulation, http://ngsim-community.org/.

[175] R. Noland and J. Polak, Travel time variability: a review of theoretical and empirical issues, in *Transport Reviews*, vol. 22 (1), pp. 39–54, 2002.

[176] P. Olszewski, Modeling probability distribution of delay at signalized intersections, in *Journal of advanced transportation*, vol. 28 (3), pp. 253–274, 1994.

[177] M. Osborne, B. Presnell and B. Turlach, A new approach to variable selection in least squares problems, in *Journal of Numerical Analysis*, vol. 20 (3), p. 389, 2000.

[178] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*, vol. 153, Springer Verlag, 2003.

[179] C. Osorio and M. Bierlaire, An analytic finite capacity queueing network model capturing the propagation of congestion and blocking, in *European Journal of Operational Research*, vol. 196 (3), pp. 996–1007, 2009.

[180] A. Ozdaglar and R. Srikant, Incentives and pricing in communication networks, in *Algorithmic Game Theory*, pp. 571–591, 2007.

[181] C. Papadimitriou, Algorithms, games, and the internet, in *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pp. 749–753, ACM, 2001.

[182] M. Papageorgiou, Some remarks on macroscopic traffic flow modelling, in *Transportation Research Part A*, vol. 32 (5), pp. 323–329, 1998.

[183] T. Park and S. Lee, A Bayesian Approach for Estimating Link Travel Time on Urban Arterial Road Network, in *Computational Science and Its Applications*, pp. 1017–1025, Elsevier, 2004.

[184] H. J. Payne, Models of freeway traffic and control, in *Mathematical models of public systems*, vol. 1, pp. 51–61, Simulation council, 1971.

[185] S. Peng, Stochastic Hamilton-Jacobi-Bellman equations, in *SIAM Journal on Control and Optimization*, vol. 30, pp. 284–304, 1992.

[186] F. Pollaczek, Über eine Aufgabe der Wahrscheinlichkeitstheorie. I, in *Mathematische Zeitschrift*, vol. 32 (1), pp. 64–100, 1930.

[187] A. Prékopa, On logarithmic concave measures and functions., in *Acta Mathematica Scientia*, vol. 34, pp. 335–343, 1972.

[188] M. Ramezani and N. Geroliminis, Estimation of Arterial Route Travel Time Distribution with Markov Chains, in *Transportation Research Board 91st Annual Meeting*, 12-0614, 2012.

[189] P. Richards, Shock Waves on the Highway, in *Operations Research*, vol. 4 (1), pp. 42–51, 1956.

[190] S. Rosset and J. Zhu, Piecewise linear regularized solution paths, in *The Annals of Statistics*, vol. 35 (3), pp. 1012–1030, 2007.

[191] T. Roughgarden, Stackelberg scheduling strategies, in *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pp. 104–113, ACM, 2001.

[192] S. Russell and P. Norvig, *Artificial Intelligence - A Modern Approach*, Prentice-Hall, Inc, Englewood Cliffs, NJ, 1995.

[193] M. Salman Asif and J. Romberg, Dynamic Updating for l_1 regularization, in *Journal of Selected Topics in Signal Processing*, vol. 4 (2), pp. 421–434, 2010.

[194] D. Schrank and T. Lomax, Urban mobility report, Tech. rep., Texas Transportation Institute, 2009.

[195] J. A. Sethian, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, vol. 3, Cambridge university press, 1999.

[196] D. Simon, *Optimal state estimation: Kalman, H_∞, and nonlinear approaches*, Wiley-Interscience, 2006.

[197] A. Skabardonis and N. Geroliminis, Real-time estimation of travel times on signalized arterials, in *16th International Symposium on Transportation and Traffic Theory*, U. of Maryland, College Park, MD, 2005.

[198] G. Stephanopoulos, P. Michalopoulos and G. Stephanopoulos, Modelling and analysis of traffic queue dynamics at signalized intersections, in *Transportation Research Part A*, vol. 13 (5), pp. 295–307, 1979.

[199] S. E. Street, Analysis and control of nonlinear infinite dimensional systems, in *IEEE Transactions on Automatic Control*, vol. 40 (4), p. 787, 1995.

[200] D. Su, A. Kurzhanskiy and R. Horowitz, Simulation of Arterial Traffic Using Cell Transmission Model, in *92nd Transportation Research Board Annual Meeting*, 13-2387, 2013.

[201] S. Sun, C. Zhang and G. Yu, A Bayesian network approach to traffic flow forecasting, in *IEEE Transactions on Intelligent Transportation Systems*, vol. 7 (1), pp. 124–132, 2006.

[202] X. Sun, L. Munoz and R. Horowitz, Mixture Kalman Filter Based Highway Congestion Mode and Vehicle Density Estimator and its Application, in *American Control Conference*, pp. 2098–2103, Boston, MA, 2004.

[203] G. Szekely and M. Rizzo, Hierarchical clustering via Joint Between-Within Distances:Extending Ward's Minimum Variance Method, in *Journal of Classification*, vol. 22, pp. 151–183, 2005.

[204] C. Tampere, S. Hoogendoorn and B. Van Arem, A behavioural approach to instability, stop and go waves, wide jams and capacity drop, in *16th International Symposium on Transportation and Traffic Theory*, pp. 205–228, 2005.

[205] A. Tarantola, *Inverse problem theory and methods for model parameter estimation*, Society for Industrial Mathematics, 2005.

[206] A. Thiagarajan, L. Sivalingam, K. LaCurts, S. Toledo, J. Eriksson, S. Madden and H. Balakrishnan, VTrack: Accurate, Energy-Aware Traffic Delay Estimation Using Mobile Phones, in *7th Conference on Embedded Networked Sensor Systems*, ACM, Berkeley, CA, 2009.

[207] R. Tibshirani, Regression shrinkage and selection via the Lasso, in *Journal of the Royal Statistical Society: Series B*, vol. 58 (1), pp. 267–288, 1996.

[208] O. Tossavainen, J. Percelay, A. Tinka, Q. Wu and A. Bayen, Ensemble Kalman filter based state estimation in 2d shallow water equations using Lagrangian sensing and state augmentation, in *47th Conference on Decision and Control*, pp. 1783–1790, IEEE, 2008.

[209] Transportation Research Board, *Highway Capacity Manual*, Transportation Research Board, National Research Council, Washington, D.C., 2000.

[210] M. Treiber, A. Hennecke and D. Helbing, Congested traffic states in empirical observations and microscopic simulations, in *Physical Review E*, vol. 62 (2), p. 1805, 2000.

[211] T. Tsekeris and A. Skabardonis, On-line performance measurement models for urban arterial networks, in *83rd Transportation Research Board Annual Meeting*, Washington, D.C., 2004.

[212] M. Van Den Broek, J. Van Leeuwaarden, I. Adan and O. J. Boxma, Bounds and approximations for the fixed-cycle traffic-light queue, in *Transportation Science*, vol. 40 (4), p. 484496, 2006.

[213] N. M. Van Dijk, On the arrival theorem for communication networks, in *Computer networks and ISDN systems*, vol. 25 (10), pp. 1135–1142, 1993.

[214] J. Van Lint, S. P. Hoogendoorn and H. Van Zuylen, Accurate freeway travel time prediction with state-space neural networks under missing data, in *Transportation Research Part C*, vol. 13 (5-6), pp. 347–369, 2005.

[215] P. Varaiya, *Complex networks and dynamic systems*, Springer Science and Business Media New York, 2013.

[216] F. Viti and H. J. Van Zuylen, The Dynamics and the Uncertainty of Queues at Fixed and Actuated Controls: A Probabilistic Approach, in *Journal of Intelligent Transportation Systems*, vol. 13 (1), 2009.

[217] Y. Wang and M. Papageorgiou, Real-time freeway traffic state estimation based on extended Kalman filter: a general approach, in *Transportation Research Part B*, vol. 39 (2), pp. 141–167, 2005.

[218] F. V. Webster, Traffic signal settings, Tech. Rep. 39, Department of Scientific and Industrial Research, Road Research Technical Paper, 1958.

[219] R. W. Wolff, *Stochastic modeling and the theory of queues*, vol. 14, Prentice hall Englewood Cliffs, NJ, 1989.

[220] D. B. Work, S. Blandin, O.-P. Tossavainen, B. Piccoli and A. M. Bayen, A traffic model for velocity data assimilation, in *Applied Mathematics Research eXpress*, vol. 2010 (1), pp. 1–35, 2010.

[221] Q. Wu, X. Litrico and A. M. Bayen, Data reconciliation of an open channel flow network using modal decomposition, in *Advances in Water Resources*, vol. 32 (2), pp. 193–204, 2009.

[222] H. M. Zhang, A non-equilibrium traffic model devoid of gas-like behavior, in *Transportation Research Part B*, vol. 36 (3), pp. 275–290, 2002.

[223] F. Zheng and H. van Zuylen, Reconstruction of delay distribution at signalized intersections based on traffic measurements, in *13th Intelligent Transportation Systems Conference*, pp. 1819 –1824, IEEE, 2010.

[224] F. Zheng and H. Van Zuylen, Uncertainty and Predictability of Urban Link Travel Time, in *Transportation Research Record*, vol. 2192, pp. 136–146, 2010.

[225] A. Zhigljavsky and A. Zilinskas, *Stochastic global optimization*, vol. 504, Springer New York, 2007.

[226] H. Zou and T. Hastie, Regularization and Variable Selection via the Elastic Net, in *Journal of the Royal Statistical Society: Series B*, vol. 67 (2), pp. 301–320, 2003.

Appendix A

Supplement: Probability distribution of delay in the congested regime

TODO: Add table summary of the results

This appendix derives the probability distribution of travel times for vehicles traveling from a location x_1 to a location x_2 on the link. As defined in the article, x is the distance to the intersection and n_s is the maximum number of stops in the remaining queue, between x_1 and x_2 (the indices x_1 and x_2 are omitted for notational simplicity). In the duration of a light cycle, the distance traveled by vehicles stopped in the queue is l_s. Thus, the maximum number of stops in the remaining queue, between x_1 and x_2,

$$n_s = \left\lceil \frac{\min(x_1, l_r) - \min(x_2, l_r)}{l_s} \right\rceil .$$

The delay experienced when reaching the triangular queue is readily derived from the expression of the delay in the undersaturated regime. The delay experienced when reaching the remaining queue is the duration of the red time R. The expression of the delay at location x is then

$$\delta^c(x) = \begin{cases} R & \text{if } x \leq l_r \\ R\,\frac{l_r + l_s - x}{l_s} & \text{if } x \in [l_r, l_r + l_s] \\ 0 & \text{if } x \geq l_r + l_s \end{cases} .$$

Case 1: x_1 is upstream of the total queue and x_2 is in the remaining queue (Figure A.1)

Since x_1 is upstream of the total queue and x_2 is in the remaining queue, all the vehicles stop once in the triangular queue between x_1 and x_2. The critical location x_c is defined as the location in the triangular queue such that

- Vehicles reaching the triangular queue upstream of x_c stop n_s times in the remaining queue. They represent a fraction $\frac{l_r+l_s-x_c}{l_s}$ of the vehicles entering the link in a cycle.
- Vehicles reaching the triangular queue downstream of x_c stop $n_s - 1$ times in the remaining queue. They represent a fraction $\frac{x_c-l_r}{l_s} = 1-\frac{l_r+l_s-x_c}{l_s}$ of the vehicles entering the link in a cycle.

The location x_c is given by $x_c = x_2 + n_s l_s$. The values of the minimum and maximum delays are given by $\delta_{\min} = (n_s - 1)R + \delta^c(x_c)$ and $\delta_{\max} = n_s R + \delta^c(x_c)$. The delay experienced by the vehicles is uniformly distributed on $[\delta_{\min}, \delta_{\max}]$.
Note that $n_s \geq 1$ since $x_2 \leq l_r$.

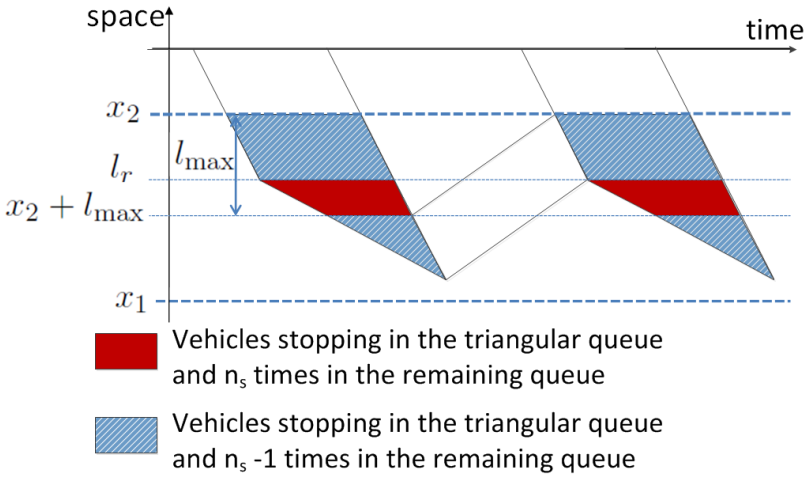


Figure A.1: Case 1: All the vehicles stop in the triangular queue. A fraction stops n_s times in the remaining queue, the other ones stop $n_s - 1$ times.

Case 2: x_1 and x_2 are upstream of the remaining queue (Figure A.2)

Given that x_2 is upstream of the remaining queue, this case is similar to the undersaturated regime. A fraction of the vehicles is not delayed between x_1 and x_2. The vehicles reaching the queue between x_1 and x_2 experience a delay in the triangular queue. This delay is a random variable, uniformly distributed on $[\delta^c(x_1), \delta^c(x_2)]$. The fraction of vehicles experiencing delay is $\eta^c_{x_1,x_2} = \frac{\min(l_s+l_r,x_1)-\min(l_s+l_r,x_2)}{l_s}$.

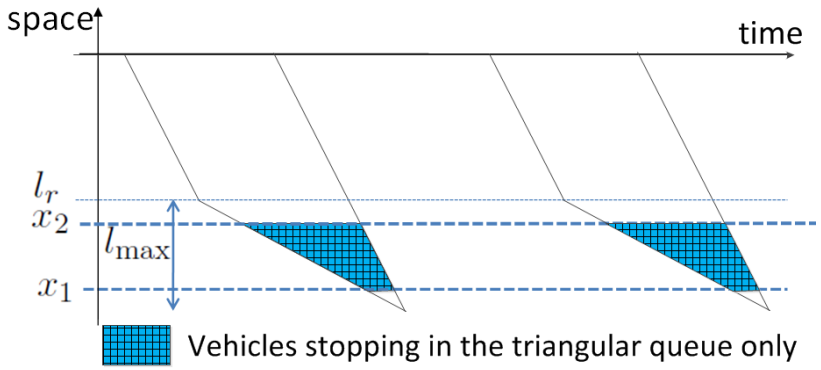


Figure A.2: Case 2: Some vehicles stop in the triangular queue. The others do not experience delay.

Case 3: x_1 is in the remaining queue, and thus so is x_2 (Figure A.3), *i.e.*

The path starts downstream of the triangular queue. Some vehicles stop n_s times and experience a delay $n_s R$ and the other vehicles stop $n_s - 1$ times and experience a delay $(n_s - 1)R$. The critical location x_c is defined as the location in the remaining queue such that

- Vehicles joining the queue between x_1 and x_c stop n_s times between x_1 and x_2. Their stopping time is $n_s R$ and they represent a fraction $(x_1 - x_c)/l_s$ of the vehicles entering the link in a cycle.
- Vehicles joining the queue between x_c and $x_c - l_s$ stop $n_s - 1$ times between x_1 and x_2. Their stopping time is $(n_s - 1)R$ and they represent a fraction $1 - (x_1 - x_c)/l_s$ of the vehicles entering the link in a cycle.

The critical location x_c is given by $x_c = x_2 + (n_s - 1)l_s$.

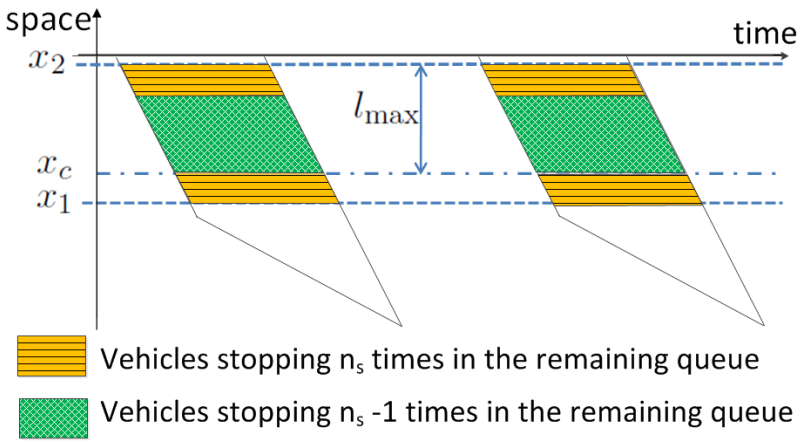


Figure A.3: Case 3: A fraction of the vehicles stop n_s times in the remaining queue. The rest stop $n_s - 1$ times in the remaining queue.

Case 4: x_1 is in the triangular queue, x_2 is in the remaining queue

The critical location x_c is defined as $x_c = x_2 + n_s l_s$ and derive probability distributions of travel times for two subcases 4a ($x_c \leq x_1$, Figure A.4 (top)) and 4b ($x_c \geq x_1$, Figure A.4 (bottom)).

Case 4a. $x_c \leq x_1$. The delay patterns are the following:

- One stop in the triangular queue and n_s stops in the remaining queue. The queue is first reached between x_1 and x_c. The delay is a random variable with uniform distribution with support $[\delta^c(x_1)+n_sR,\ \delta^c(x_c)+n_sR]$. The vehicles following this pattern represent a fraction $\frac{x_1-x_c}{l_s}$ of the vehicles entering the link in a cycle.
- One stop in the triangular queue and $n_s - 1$ stops in the remaining queue. The queue is first reached between x_c and l_r. The delay is a random variable with uniform distribution with support $[\delta^c(x_c)+(n_s-1)R,\ \delta^c(l_r)+(n_s-1)R]$. Noticing that $\delta^c(l_r) = R$, it follows that the support of the delay distribution is $[\delta^c(x_c) + (n_s - 1)R,\ n_sR]$. The vehicles following this pattern represent a fraction $\frac{x_c-l_r}{l_s}$ of the vehicles entering the link in a cycle.
- No stop in the triangular queue and n_s stops in the remaining queue. The queue is first reached between l_r and $x_1 - l_s$. The delay is n_sR. The vehicles following this pattern represent a fraction $\frac{l_r-(x_1-l_s)}{l_s}$ of the vehicles entering the link in a cycle.

A sanity check validates that the weights of the different components sum to 1:

$$\frac{x_1 - x_c}{l_s} + \frac{x_c - l_r}{l_s} + \frac{l_r - (x_1 - l_s)}{l_s} = 1.$$

Remark that $x_2 \leq l_r$ implies that $n_s \geq 1$. Then using the definition of x_c, $x_c = x_2 + n_s l_s$ and the fact that $x_1 \geq x_c$, it follows that $x_1 - l_s \geq x_2$ and all vehicles reach the queue between x_1 and $x_1 - l_s$.

Case 4b. $x_c \geq x_1$. The delay patterns are the following:

- One stop in the triangular queue and $n_s - 1$ stops in the remaining queue. The queue is first reached between x_1 and l_r. The delay is a random variable with uniform distribution on $[\delta^c(x_1) + (n_s - 1)R,\ \delta^c(l_r) + (n_s - 1)R]$, *i.e.* uniform distribution on $[\delta^c(x_1) + (n_s - 1)R,\ n_s R]$. The vehicles following this pattern represent a fraction $\frac{x_1 - l_r}{l_s}$ of the vehicles entering the link in a cycle.
- No stop in the triangular queue and n_s stops in the remaining queue. The queue is first joined between l_r and $x_c - l_s$. The delay is $n_s R$. The vehicles following this pattern represent a fraction $\frac{l_r - (x_c - l_s)}{l_s}$ of the vehicles entering the link in a cycle.
- No stop in the triangular queue and $n_s - 1$ stops in the remaining queue. The queue is first joined between $x_c - l_r$ and $x_1 - l_s$. The delay is $(n_s - 1)R$. The vehicles following this pattern represent a fraction $\frac{x_c - x_1}{l_s}$ of the vehicles entering the link in a cycle.

A sanity check validates that the weights of the different components sum to 1:

$$\frac{l_r - (x_c - l_s)}{l_s} + \frac{x_1 - l_r}{l_s} + \frac{x_c - x_1}{l_s} = 1.$$

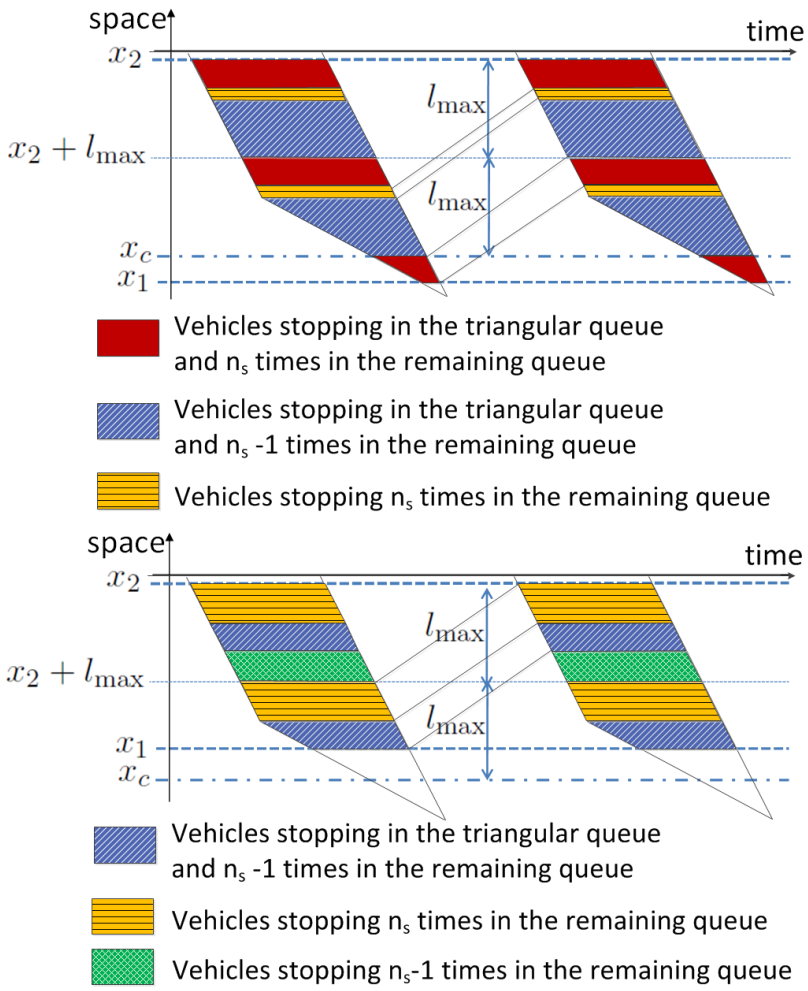


Figure A.4: Case 4: (**Top**) Case 4a: a fraction of the vehicles stop in the triangular queue and n_s times in the remaining queue, a fraction of the vehicles stop in the triangular queue and n_s times in the remaining queue, the rest stop n_s times in the remaining queue. (**Bottom**) Case 4b: a fraction of the vehicles stop in the triangular queue and $n_s - 1$ times in the remaining queue, a fraction of the vehicles stop n_s times in the remaining queue, the rest stop $n_s - 1$ times in the remaining queue.