Real-Time Traffic Modeling and Estimation with Streaming Probe Data using Machine Learning

by

Ryan Jay Herring

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

in

Industrial Engineering and Operations Research

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Laurent El Ghaoui, Chair
Professor Zuo-Jun Shen
Research Advisor, Professor Alexandre Bayen

Fall 2010

The dissertation of Ryan Jay Herring, titled Real-Time Traffic Modeling and Estimation with Streaming Probe Data using Machine Learning, is approved:

Chair ______________________________ Date __________

______________________________ Date __________

Research Advisor ______________________________ Date __________

University of California, Berkeley

Real-Time Traffic Modeling and Estimation with Streaming Probe Data using Machine Learning

Abstract

Real-Time Traffic Modeling and Estimation with Streaming Probe Data using Machine Learning

by

Ryan Jay Herring

Doctor of Philosophy in Industrial Engineering and Operations Research

University of California, Berkeley

Professor Laurent El Ghaoui, Chair

Traffic information systems play an important role in the world as numerous people rely on the road transportation network for their most important daily functions. This dissertation proposes a general system architecture for processing traffic data and for disseminating accurate, timely traffic information via the internet. It also specifically addresses the challenges with estimating *arterial* traffic conditions using only data from GPS probe vehicles. GPS probe data promises to be the most ubiquitous source of traffic data for years to come as transit agencies decrease their investment in traditional fixed-location sensing infrastructure.

The dissertation introduces the architecture design and implementation of the *Mobile Millennium* system. A joint project between UC Berkeley, Nokia and Navteq, *Mobile Millennium* aggregates data from numerous sources, runs state of the art estimation and forecast algorithms, and provides timely traffic information to drivers and other targets. This system took over two years to build and the result is a robust framework for any traffic estimation researcher to access vast stores of data quickly and easily as well as test any number of estimation algorithms.

For estimating arterial traffic conditions, this dissertation proposes a hybrid approach leveraging advances in the fields of machine learning and traffic theory (based on hydrodynamic theory). This approach provides a foundation for any arterial traffic estimation model. A variety of model/algorithm approaches are presented, with one ultimately proving to be superior to the rest and the one that should be carried forward as research in this area continues.

This dissertation is dedicated to my amazing family. My mother, Linda, my father, Ric, and my sister, Megan. I love you all!

Contents

List of Figures

List of Tables

Acknowledgments

There are many people I would like to thank for all of their help in making this dissertation possible. First and foremost, I want to thank my parents, Ric and Linda Herring, for being supportive of absolutely everything I have done in life and always encouraging me to pursue any and every opportunity.

I would like to thank my advisor, Associate Professor Alexandre Bayen, for seeing my potential at critical crossroads for me in my pursuit of the Ph.D. His encouragement and guidance led to me becoming a key member of the *Mobile Millennium* team and ultimately to this dissertation.

I have worked with many amazing individuals in the course of my Ph.D. studies. At the top of that list is Aude Hofleitner, with whom I have worked extremely closely for more than two years now. Much of the work done in this dissertation was in collaboration with her and I owe her immense thanks for all of her encouragement as well as all of the times she pointed out that I was doing something wrong. Her insight and ability to think critically about the problems we were trying to solve made the difference between success and failure.

Earlier in my graduate career, I worked very closely with Assistant Professor Jeff Ban. I want to thank Jeff for teaching me a lot of what I know about traffic and for our early work together, which ended up being the inspiration for the work that would ultimately become my dissertation.

I would like to thank Timothy Hunter, who put up with numerous requests for data formatted in some specific way as well as for all of his work on data filtering that was instrumental to the success of the project. I'd also like to thank Assistant Professor Pieter Abbeel, who provided sound guidance as we found our way through the field of machine learning and for always seeing the right trick for how to solve difficult problems.

Finally, I would like to thank all of the *Mobile Millennium* team and everybody who works at the California Center for Innovative Transportation (CCIT). CCIT has been my home for more than three years and I have always found it a very supportive environment to work in. Special thanks to the *Mobile Millennium* project manager, Joe Butler, who has managed to find a way to keep us all on track. Also thanks to the systems team, including Saneesh Apte, Daniel Edwards, Jonathan Felder, Tom Kuhn, and former members Yanli Li, Bill Vogel, and Morgan Smith. They all made the implementation of the design that we constructed come to fruition. Additional thanks to the other students with whom I've worked closely, including Saurabh Amin, Dan Work, Samitha Samaranayake, Christian Claudel, Sebastien Blandin and former students and post-docs Olli-Pekka Tossavainen, Juan Carlos Herrera, Pierre-Emmanuel Mazare, Matthieu Nahoum, Marcella Gomez and Sarah Stern, not to mention numerous other students and engineers who helped our project at various points. Additional thanks to the administrative and management people of CCIT including Tom West, Coralie Claudel and Steve Andrews. Also, I want to specifically thank J.D. Margulici for bringing me into the CCIT family and encouraging me at an early stage in graduate school.

Chapter 1

Introduction and Motivation

Transportation is one of the most fundamental systems in the world today. Individuals increasingly rely on the ability to travel longer distances in shorter amounts of time to accomplish their personal and professional tasks. Among all forms of transport, vehicle transportation through the road network is perhaps the most important for people in many parts of the world, particularly the United States. The 2007 Urban Mobility Report [93] states that traffic congestion causes 4.2 billion hours of extra travel in the United States every year, which accounts for 2.9 billion extra gallons of fuel, costing taxpayers an additional $78 billion.

An essential step in mitigating the effects of traffic congestion is the creation of real-time traffic monitoring systems. This dissertation proposes a global approach to designing *traffic information systems* with a specific focus on estimating arterial traffic conditions from sparse GPS probe data. *Arterials* (also known as the secondary network) are major city streets (not highways) that provide for travel within and between cities. *Probe data* refers to location and speed measurements provided by a subset of vehicles traveling through the road network. This work leverages the increasing availability of mobile devices (i.e. cell phones) with sensors such as GPS that are capable of providing detailed information about the traffic conditions experienced by the driver carrying the device. The goal is to ultimately provide accurate, timely estimates and forecasts of traffic conditions to anyone.

The remainder of this chapter introduces the context in which this work has been performed by presenting the existing technology for traffic estimation (section 1.1) followed by a summary of the rise of GPS probe data (section 1.2). It is precisely this context which led to the creation of the *Mobile Millennium* project, described in section 1.3. The work of this dissertation was completed almost entirely within the context of the project and the specific contributions are presented in section 1.4.

1.1 Background

Historically, traffic monitoring systems have been mostly limited to highways and have relied on public or private data feeds from a dedicated sensing infrastructure, which often includes loop detectors, radars, video cameras. It is clear why fixed-location sensors were preferred when transportation engineers first sought to measure traffic conditions as there was no easy way of tracking any significant subset of vehicles. By placing fixed-location sensors, traffic engineers ensured that they could record data from nearly every vehicle passing by a given location. Of course, the obvious disadvantage to placing sensors in this manner is that data is only recorded within a short distance of the sensor. Thus, to cover a significant portion of the road network, a large number of sensors must be placed. Even the cheapest fixed-location sensors are too expensive to cover the entire road network.

Given the high cost of placing sensors on the road network, it was an obvious choice for most government transit agencies to place these sensors on the highway network first, which accounts for a small fraction of the total length of the road network, but which is arguably the most important for drivers (particularly in the United States). In most major metropolitan areas today, the road network is generally very well covered by fixed-location sensors that provide streaming data to various traffic information systems. However, in many parts of the world, this kind of infrastructure still does not exist at all and the state of the economies in these locations is often such that there will not be resources to build out such an infrastructure in the foreseeable future.

While the highway network is generally well-equipped with dedicated sensing infrastructure (in the United States), the arterial network is far behind. There are very few locations in the United States where fixed-location sensors are providing streaming data to central servers that can process the data in real-time. There is also a general lack of commitment to building more sensors due to the high cost to cover the whole road network. Without a means to acquire traffic data on roads lacking fixed-location sensors, the traffic engineering community naturally began to explore other possibilities for measuring traffic conditions. It is precisely this context where the work of this dissertation begins.

1.2 Introducing GPS Data

The GPS system has been in place for decades. However, only within the last decade have devices become available that are capable of providing high-accuracy tracking information at relatively low costs. This development has led to the use of cheap GPS tracking devices placed in vehicles for gather traffic information. Today, GPS data comes from various sources, each of them with specific issues:

- *Fleet data.* Numerous vendors offer feeds coming from fleets (FedEx, UPS, taxis, etc.) which provide very valuable data (GPS data sampled at a one minute rate is currently the standard), however, these vehicles have specific spatio-temporal travel patterns and sometimes try to avoid congestion (since frequent drivers often learn the typical traffic patterns through a city).

- *Smartphone or aftermarket device data.* Several companies collect cell phone data from various GPS-enabled smartphone applications or 2-way navigation devices, for example Garmin, INRIX, Google, Nokia or Waze. The challenge of this data is that it is unpredictable, sparse, and no single company has ubiquitous coverage.
- *RFID tag data.* In some cities, RFID tags carried by vehicles used for toll collection are also used for traffic monitoring, based on the deployment of readers along the transportation network. The spatial density of such readers comes with its own challenges for estimation, and its coverage is not ubiquitous.

The primary benefit of GPS probe data is the fact that traffic conditions can be measured wherever probe vehicles go, without needing to build any new sensors. However, there are some obvious concerns associated with this type of data as well. These include the fact that tracking is privacy invasive and that if few individuals are willing to provide such data, there will be insufficient information to estimate traffic conditions with high accuracy. This dissertation proposes multiple solutions for addressing these concerns.

1.3 *Mobile Millennium*

The *Mobile Millennium* project resulted from a partnership between Nokia [11], Navteq [9] and UC Berkeley, with support from Caltrans [3] and the United States Department of Transportation [18]. It also includes other sponsors: the National Science Foundation [12], the Volvo Foundation [16], UCTC [17], Tekes [7], VTT [19], CITRIS [6], and CalFrance [4]. The project was officially launched on November 10, 2008 when a client for GPS-enabled smartphones became publicly available for download. In this initial system, traffic conditions were broadcasted back to drivers' mobile phones, which enabled commuters to make more intelligent route and trip decisions. Additionally, one can view traffic data in the *Mobile Millennium* visualizer. The deployment area is focused on commuters in Northern California, including the San Francisco Bay Area and Sacramento, which are areas with heavy recurring congestion on numerous roadways. The project is a follow up to the *Mobile Century* experiment, in which 165 UC Berkeley graduate students were hired to drive a 10-mile loop of I880 in California for a day, demonstrating the feasibility of a real-time traffic estimation service using only GPS enabled devices for estimating highway traffic conditions [52]. The project, and the system that was built as part of it, are described in detail in section 2.3.

The motivation to start the *Mobile Millennium* project was driven by the aforementioned lack of sensing infrastructure on the majority of the road network combined with the increasing availability of GPS data from cell phones. The rise of the mobile internet was the underlying technology enabling the existence of the project. User-generated content (in the present case, data about how fast one is driving) is contributed to a central system, which provides information back to the user for personal use. This "web 2.0" application framework is commonly referred to as "participatory sensing", which means that anyone who contributes to the system also realizes the benefits from the system. In particular, *Mobile Millennium* is an example of a "cyber-physical" system, which couples physical and cyber components into a single application framework. The physical component of the system is the flow of vehicles along the roadway. Vehicles carrying GPS-enabled cell phones provide a link to the cyber components, which are sensing data and traffic estimation. In general, there are a number of challenges to overcome with any sensing technology including unknown location, sparsity of the data, and unpredictability of the frequency of data collection.

The *Mobile Century* experiment and the research that followed demonstrated the ability of the system to estimate traffic conditions using only GPS probe data. The second challenge for the project was to do the same for arterial traffic. Aside from less abundant sensing compared to existing highway traffic monitoring systems, the arterial network presents additional modeling and estimation challenges as the underlying flow physics which governs them is more complex because of traffic lights (often with unknown cycles), intersections, stop signs, parallel queues, and others. Collecting the detailed parameters of the arterial road network into an accessible electronic database would require the cooperation of numerous government agencies, making this information unreliable and difficult to obtain. Moreover, at the low penetration rate typical for arterial traffic, even small changes in the road network can greatly affect the estimation. This makes the detailed spatio-temporal modeling and estimation approaches developed for highway traffic impractical for arterials—at least until the data volume significantly increases [24, 78].

1.4 Contribution

To summarize the context of this work, the previously unsolved challenge was to estimate traffic conditions on arterial roads that have no dedicated sensing infrastructure and for which the only potentially available source of data is sparse GPS probe data. Due to all of the difficulties associated with arterial traffic (which will be explained in more detail later), this dissertation proposes a machine learning framework for leveraging large amounts of historic data for use in real-time estimation algorithms. The statistical approach proposed in this dissertation relies on advances in multiple fields, including traffic theory (based on hydrodynamic theory) and machine learning. The primary contribution is the fusion of these fields to solve the problem of arterial traffic estimation within the specified context. This work is presented in chapters 4 - 6.

The contribution of this dissertation is not only limited to the context of *Mobile Millennium*, however. While the context of the work was the primary motivation for taking this specific research approach, the result of the work is that it can be applied in a general context in which any type of traffic data is available for any type of roads (including highways).

Additionally, the core of the *Mobile Millennium* system was built as part of this dissertation, which is the second main contribution. Building a traffic information system that can handle a wide variety of data types and can handle multiple models all running on top of the same infrastructure was a big challenge that has been solved, enabling future research on traffic estimation with many fewer obstacles to success. These system development contributions are presented in chapter 2.

Chapter 2

Traffic Information Systems

Traffic information systems have numerous challenging requirements for them to be useful on a large scale. Among these requirements, traffic information must be accurate and available in real-time, while also easily interpreted by the user. This implies that the system needs to be built using redundant servers to handle the "always available" requirement and also requires substantial computing power to estimate traffic conditions across the entire road network (which can be hundreds of thousands of links within a geographic area). Furthermore, the system must be able to visualize traffic conditions in a way that users can understand the information they need quickly. This invariably includes color-coded maps indicating congestion, but also must include travel time information between arbitrary points on the network, vehicle density estimates for traffic management centers (and potentially air pollution estimation models), and in-vehicle navigation systems with real-time routing. All of these components need to interact through a common infrastructure and be able to provide information to any other part of the system quickly.

These requirements lead to a number of practical design issues that need to be examined in detail to understand how the entire traffic information system works as a whole. System design is a major part of the research presented in this dissertation and provides an essential basis for the more theoretical traffic estimation theory that follows. This chapter presents many of the core features of a prototypical traffic information system and details the design decisions that must be made to satisfy the global requirements as specified above. These features include privacy, data accuracy assessment, scalability, road network representation, map matching, visualization, and sensor deployment (all included in section 2.2).

The first part of this chapter (section 2.1) presents the numerous sources of traffic data that exist in the world today as well as how ubiquitous and reliable these sources of data are for performing real-time traffic estimation on the arterial network. All of these sources of data come with their own particularities in terms of coverage, accuracy, and timeliness. A good traffic information system is capable of using all available data sources, determining their relative merits and then estimating traffic conditions. The use of many (but not all) of these data sources will explicitly appear in later chapters. The overview presented in the present chapter provides a basis for interpreting the work presented in the literature review (chapter 3) as well as the models that appear later in this dissertation.

2.1 Taxonomy of Traffic Sensor Types

A number of sensors have been developed in the past 50 years designed to collect various types of traffic data. In general, traffic data includes flows (number of vehicles per time unit), density (number of vehicles per distance unit), occupancy (percentage of time a vehicle is over of specific location, which is directly related to density), velocity (distance per unit time), and travel time (time to travel between two locations). One additional data type possible are vehicle trajectories, which are always represented by a sequence of discrete time/location pairs for each vehicle. From vehicle trajectory data with a location-reporting frequency of several seconds or less, travel times and short distance velocities can be directly computed. When the location-reporting frequency is more than 10 seconds, directly measuring travel times and velocities becomes non-trivial. The mathematical details of these data types will be discussed in more detail at the beginning of the literature review in chapter 3.

The remainder of this section lists the most ubiquitous traffic sensors and describes the data types that each of them provides. This includes a discussion of the accuracy, timeliness, and spatial resolution of the data provided by each sensor type. Also presented are typical placement strategies and common road types that are covered by each sensor.

2.1.1 Loop Detectors

Inductive loop detectors are built into the roadway so that they can detect each vehicle that passes over them. They work by detecting the metal of a vehicle as it passes over the detector. Properly calibrated, a loop detector is capable of providing high-accuracy flow and occupancy data [33], the latter of which can be used to infer density [61]. When two loop detectors are placed close together, velocity can be measured by looking at consecutive crossing times. While the quality of the measurements from loop detectors is often good, filtering is still required from producing quality input data to highway estimation models [37]. Loop detectors are not capable of directly measuring travel times.

Loop detectors are commonly found on most major highways throughout the United States and Europe. Many of these locations have loop detectors connected to an internet connection that can be used to transmit the data to a central server in real-time (that can subsequently be used in traffic information systems). Many locations throughout the United States and Europe also have loop detectors placed on arterial roads. However, for arterial roads, it is very rare for the loop detector to be connected to the internet for easy transmission of the data to a central server. For this reason, arterial traffic information systems cannot rely on loop detector data as there is not enough of it to estimate conditions on the whole arterial network.

2.1.2 Radar

Radar detectors can be placed on poles along the side of the road enabling them to collect flow, occupancy and velocity data. In general, radar detectors provide lower accuracy data than loop detectors [72].

As of this writing, dedicated radar detectors that are connected to the internet and providing data in real-time are still relatively rare in the United States and Europe. Where these are available, they are placed almost exclusively on highways. Radars are generally not well suited to mass data collection on arterials due to the fact that accuracy decreases in arterial environments [72]. For this reason and the fact that almost no radar data exists on arterials, they are not considered viable inputs into an arterial traffic information system.

2.1.3 Video

Video recording can be used to collect traffic data in two ways. The first way is to use high resolution cameras placed high above the roadway to track all vehicles within the view of the camera. The second way is to use the video cameras to record license plate numbers at specified locations, which is equivalent to using video as a license plate reader (see section 2.1.4 for further description of this kind of data collection).

Using high-resolution cameras to track vehicle trajectories does 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 [72]. When properly processed, video can provided very high-resolution vehicle trajectories (vehicle positions every tenth of a second). However, this technology is expensive to deploy and can only cover a relatively small portion of the roadway (generally less than a mile). The NGSIM project [10] is an example of the use of this kind of technology, which to date has mostly been used to provide researchers with high-accuracy vehicle trajectories over a small spatio-temporal domain (less than a mile for less than an hour). This kind of data is valuable to arterial traffic estimation research, but given that the data does not come in real-time, it cannot be used in real-time traffic information systems.

2.1.4 License Plate Readers

When placed directly above a lane of traffic, license plate readers are capable of automatically extracting the numbers and letters from passing vehicles. When multiple readers are setup at two points along the road, it is possible to extract travel time information for vehicles passing both locations.

License plate readers suffer from the logistical problem of finding good locations to place them so that they can be used effectively. When properly positioned and calibrated, these devices are capable of providing high-accuracy travel times [5]. However, even when the devices correctly measure individual vehicle travel times, one still needs to filter the travel times to account for vehicles that stop in the middle of the route between the two sensors. In fact, this need to filter travel times arises whenever collecting travel times by placing two sensors capable of re-identifying vehicles (but that do not track the vehicle in between). One particular filtering strategy is the *Median Absolute Deviation* filter, described in [26].

Due to the difficulty in placing these devices, they are not common throughout the roadway. As of this writing, they remain a data collection tool for specific studies, but not for large-scale traffic data collection. This makes them unusable for arterial traffic information systems.

2.1.5 RFID Transponders

Radio-Frequency Identification (RFID) is a ubiquitous technology in many industries. Transit agencies make use of RFID in several ways. One of the original uses of this technology was for collecting tolls from drivers when crossing a bridge or entering/exiting a toll road. The vehicle has a RFID transponder which is detected by a reader placed at the entrance/exit of the toll road [26].

This same technology 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 in the same way that travel times from license plate readers can be processed (see section 2.1.4). The accuracy of RFID transponders varies depending on the strength of the signal. It is generally accurate enough to provide long distance travel time estimates, but may not provide high-accuracy travel times over short distances. 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. It is also not common to find this technology on arterial roads.

2.1.6 Bluetooth

Bluetooth readers have been developed in recent years [96], which are capable of scanning the surrounding airwaves for Bluetooth enabled devices. If readers are placed at points along the roadway, travel times can then be measured between consecutive readers for all vehicles carrying a Bluetooth device. In addition to the filtering challenges associated with these travel time measurements described in section 2.1.4, Bluetooth readers also suffer from the problem of having a relatively high detection range. This is a good thing in the sense that the readers rarely miss detecting a vehicle, but bad in the sense that it is difficult to determine the precise time that a vehicle passed the reader as it might be detected continuously for more than a minute.

To assess the accuracy of travel times provided by Bluetooth readers on arterial roads, two sets of field tests were performed in the Bay Area as part of the present work. Each field test consisted of between 12 and 20 drivers, each carrying both a high-frequency GPS device (see section 2.1.10) and a Bluetooth device. Bluetooth readers were placed at various points along the roads covered in the experiments. The GPS data allowed for precise calculation of the time each vehicle passed each Bluetooth reader and these crossing times were compared to the crossing times reported by the Bluetooth readers. Complete results are presented in [41]. The conclusions from the tests were that the error in the Bluetooth readers were location dependent and could have errors in the detection time as high as one minute. The overall error in the travel time estimates was around 15%.

Bluetooth readers are an emerging technology and are therefore not available in most areas. If they were available in large quantities on arterials, they could potentially be used for traffic estimation, but due to the general lack of sensors of this type, they are not considered viable data providers for traffic information systems at the current time. Also, it is likely that enhancement algorithms will be developed that will correct for the inaccuracies in the data provided by these devices.

2.1.7 Wireless Sensors

Wireless sensors are small devices embedded into the roadway (similar to loop detectors), but the detection mechanism in these sensors allows for re-identifying vehicles at subsequent sensor locations with up to 80% accuracy for one particular system [66]. Thus, these sensors provide travel times for a large percentage of the flow of traffic (and the travel times must still be filtered as discussed in section 2.1.4). The primary advantage of these sensors over other travel time measurement sensors is that they are much cheaper to produce, potentially allowing for large-scale deployment on arterial roads. However, at the current time, they are only available in a small number of locations. Sensys Networks [15] is currently one of the leading providers of these sensors.

2.1.8 Virtual Trip Lines (VTL)

Virtual trip lines (VTL) comprise the basis of a “participatory sensing” system that allows individuals to download an application onto their GPS-enabled smartphone that both sends traffic data as well as receives traffic information and alerts. A VTL is a virtual line drawn on the road. The basic idea is that the phone monitors its own GPS position every few seconds and has downloaded a list of VTLs in the general region that the phone resides in. When the phone crosses one of the VTLs, it sends an update to the central VTL server indicating its velocity and time of crossing as well as the travel time from the previous VTL it crossed. The accuracy of the velocity data generated by frequent GPS sampling varies greatly with the type of GPS chip in the phone and can be very good in some cases and very bad in others. It is generally accurate enough for highway traffic estimation when properly filtered [98]. For arterials, the velocity measurements are not reliable, so the travel time measurements are the only data that is suitable for arterial traffic estimation. The travel time measurements also need to be filtered as described in section 2.1.4.

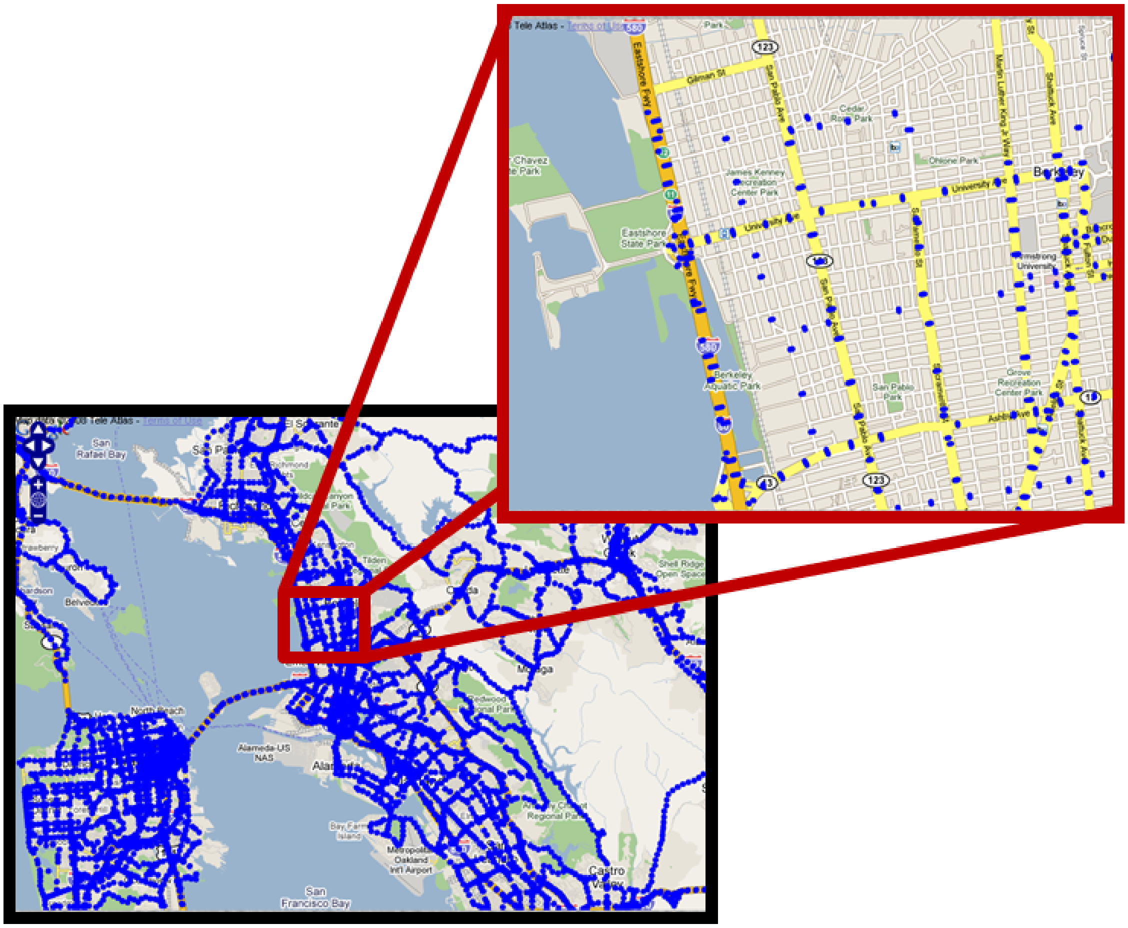


Figure 2.1: Example of a Bay Area VTL deployment as part of the *Mobile Millennium* system.

Nokia [11] originally developed the first VTL-based traffic data collection system in 2007. This system was first tested as part of the *Mobile Century* experiment in February, 2008. The results of this initial experiment can be found in [52].

VTLs are capable of providing high-quality traffic data while also helping to preserve the privacy of individuals by only disclosing data at pre-specified locations [56]. One challenge is determining where the VTLs should be placed throughout the network so as to collect relevant traffic data while not infringing individual privacy or placing VTLs so densely that an unnecessary amount of data is transmitted through the communication network. No studies have been conducted to date on proper VTL placement for addressing these issues. An experimental deployment of VTLs was tested as part of the *Mobile Millennium* project, covering all of California as seen in figure 2.1 for some parts of the Bay Area.

The VTL concept has been proven to work when sufficiently many individuals download the software on their phone [52]. However, there is not currently a VTL-based system with enough users to provide a reliable stream of data for either highway or arterial traffic estimation and it is not clear that this sampling paradigm will be used on a large-scale basis in the future. Also, given that companies in the business of traffic monitoring do not always disclose their proprietary implementations, it is not clear to this day how wide-spread the concept has become in practice.

2.1.9 Sparsely-sampled GPS

Sparsely-sampled probe GPS data refers to the case where probe vehicles send their current GPS location at a fixed frequency, which is not frequent enough to directly measure velocities or link travel times (i.e. sampling frequency is more than about 10 seconds). There are several challenges associated with this type of data. First, GPS measurements must be mapped to the road network representation used by the traffic information system, which means that the correct position on the road as well as the path in between successive measurements must be determined. This process is known as map matching and path inference, which is described in more detail in section 2.2.5. Second, probe vehicles can often travel multiple links between measurements when the sampling frequency is low, which means that one must infer what the likely travel times on each link of the path were. This is part of the traffic estimation algorithm, which will be described in detail in chapters 5 and 6.

Sparsely-sampled probe GPS data is currently the most ubiquitous data source on the arterial network. An example of this type of data comes from the Cabspotting project [1], which provides the positions of 500 taxis in the Bay Area approximately once per minute. Figure 2.2 shows one full day of raw data, which demonstrates that even just a single data source such as taxis can provide broad coverage of a city. This data clearly has some privacy issues as it is possible to track the general path of the vehicle. However, the majority of this data today comes from fleets of various sorts (such as UPS, FedEx, taxis, etc.). Most of this data is privately held among several companies, but between all sources there are millions of records per day in many major urban markets. One publicly available source of this kind of data is the Cabspotting project [1]. This project provides one-minute samples of the positions of over 500 taxis in San Francisco, CA. This results in upwards of 500,000 measurements per day. Due to the ubiquity of this data source, it is paramount that it be used in an arterial traffic information system. Indeed, it is the only source that is likely to be available across the arterial network in the next decade.

2.1.10 High-frequency GPS

High-frequency probe GPS data refers to the case where probe vehicles send their current GPS location every few seconds (no more than about every 10 seconds). This kind of data is generally the most accurate kind of vehicle trajectory data possible, especially when sampling every second with a high-quality GPS chip. From this data, one can directly infer velocities

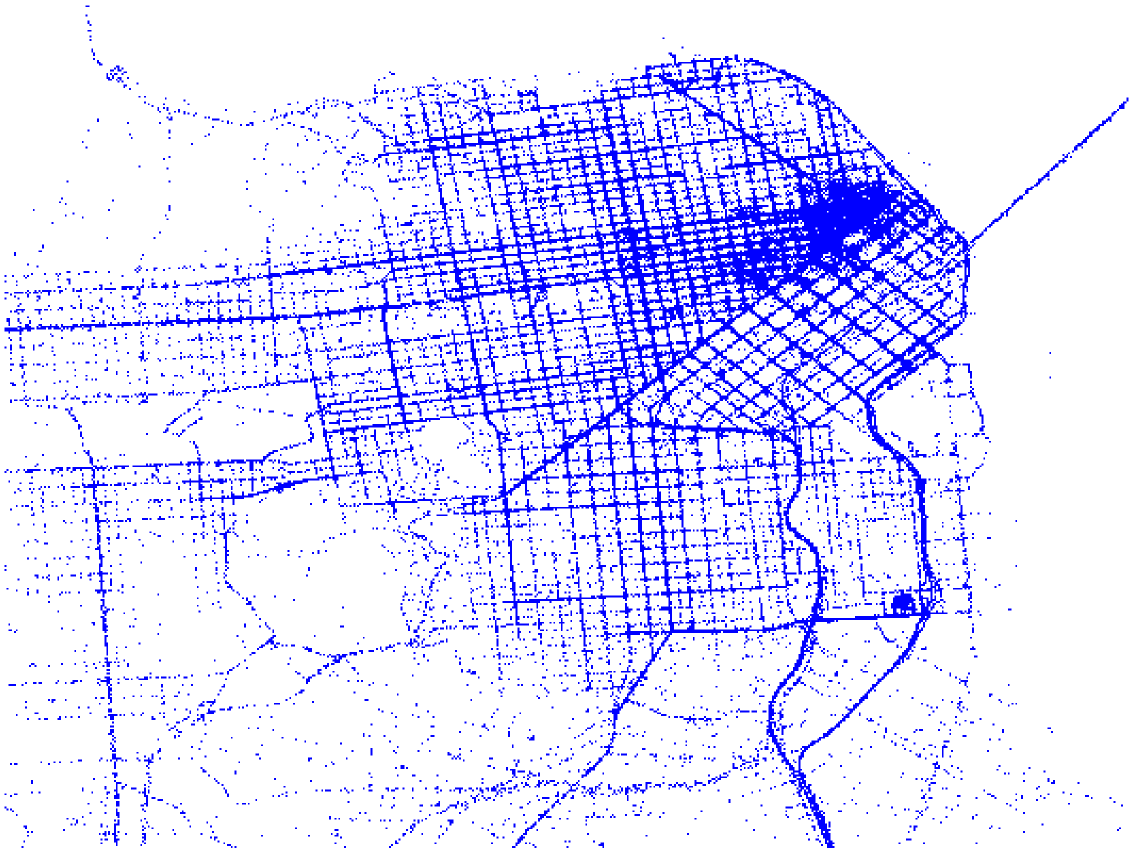


Figure 2.2: One day of sparsely-sampled GPS data from San Francisco taxi drivers as provided by the Cabspotting project.

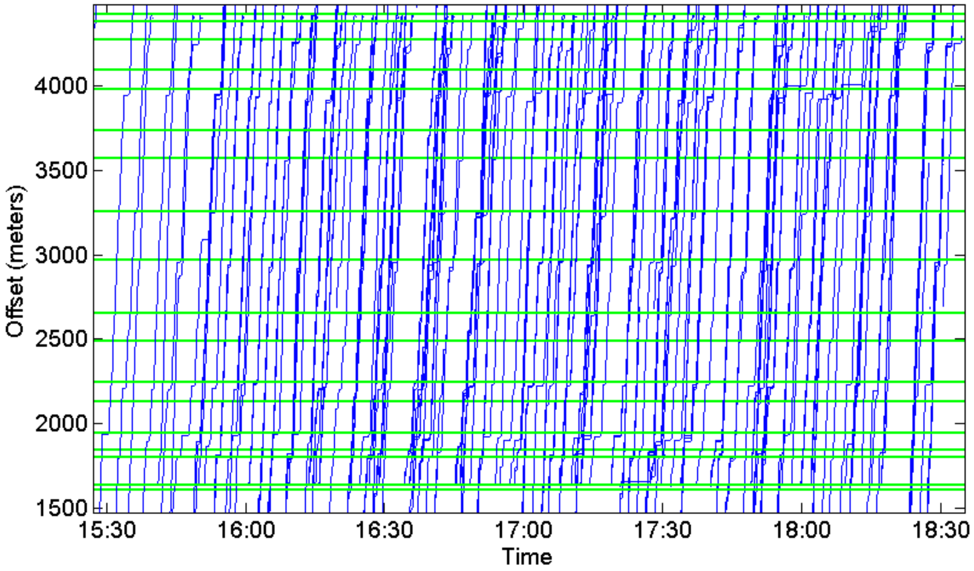


Figure 2.3: Vehicle trajectories from the *Mobile Millennium* evaluation experiment on San Pablo Avenue in Berkeley, Albany and El Cerrito, California. The high-frequency GPS data in this figure is represented as distance (meters) from an arbitrary start point upstream of the experiment location. The horizontal lines represent the locations of the traffic signals along the route.

and short distance travel times. The issue of map matching is still present as there can be ambiguity around intersections, but the path is usually easy to determine when examining the entire trace. Figure 2.3 depicts a sample of high-frequency data collected as part of the *Mobile Millennium* project. This figure illustrates the level of detail that can be extracted from high-frequency data, but also shows the relatively low percentage of vehicles that were being tracked as there are occasional gaps of five minutes or more between trajectories.

Sampling a vehicle's position every few seconds is clearly very privacy invasive and it also comes with large communication costs to send the high volume of data. For these reasons, it is not common to receive this data with any kind of regularity. This data is often collected for specific experimental studies, but is not generally available for real-time traffic information systems.

2.2 Practical Considerations for Designing a Traffic Estimation System

In this section, the core issues of practical importance to users of traffic information systems are discussed in detail. The work of this dissertation contributed to the *Mobile Millennium* system by designing and writing software that addressed all of these issues (often with other members of the team). In particular, the network abstraction and map matching functionalities of the system were primarily designed and implemented using this work, and they have been relied upon by more than 20 members of the *Mobile Millennium* team.

2.2.1 Driver Privacy

The expectations of drivers with respect to the privacy of their location measurements varies greatly and is also a generational problem. Some drivers will not be comfortable sharing any data at all, some will be willing to share some data in exchange for value of some kind (real-time routing around traffic, for example), and some would be willing to share any and all data as long as it does not interfere with their ability to use their phone or other GPS device. The *Mobile Millennium* system was designed to accommodate all of these privacy preferences. The system is "opt-in", meaning that drivers who do not wish to provide any data can simply choose to not install the application running on the phone that collects traffic data. For those who wish to participate and receive traffic information on their phone, a spatially-aware sampling system (based on VTLs) was designed to extract information without compromising user anonymity [56]. A technical description of how VTLs work can be found in the sensor taxonomy portion of this chapter (section 2.1.8).

There are two primary reasons why VTLs respect driver privacy more than fixed-interval location reporting. First, driver data is collected only at pre-defined locations which only include highways and major arterials, but not residential roads. Second, driver data collected at one location is not re-associated with that same driver's data at another location, except for short distance travel times. These two features combined mean that origin and destination information is unavailable for anyone who has access to the data collected inside the system.

GPS tracking data (both sparse, section 2.1.9, and high-frequency, section 2.1.10) is not intended to preserve the privacy of the driver. The *Mobile Millennium* system collects this data specifically from sources who have agreed to provide it in that form and are not concerned with privacy of the drivers (the primary function of the data is generally to track service vehicles). This is generally restricted to fleet delivery vehicles or taxis, but if an individual driver wanted to participate in this manner, the data can be collected in that form.

2.2.2 Raw Data Accuracy and Filtering

No data source is perfect and every piece of data received by the *Mobile Millennium* system goes through a specific filtering process. Data from fixed-location sensors generally requires a much different filtering process than GPS data. In the *Mobile Millennium* system today, fixed-location sensor data is only available on the highways and particular filtering algorithms have been developed specifically for highway traffic estimation algorithms. The basic idea behind fixed-location sensor filtering is to correct the values being reported by the sensors to account for the noise in the measurements. GPS data is the only available data on arterial roads in the *Mobile Millennium* system and this data requires both map matching and path inference (see section 2.2.5). These processes represent a different notion of filtering, where instead of correcting values, the data is actually being translated from one spatial reference system (GPS position on the Earth) to another (link identifiers and position along the link using a network representation of the road).

The purpose of mentioning issues of data quality and filtering here is to highlight the importance of the data cleaning process to the overall goal of building a high-quality traffic information system. Furthermore, it is important to have analytical measures for how accurate a particular data source is. While not the subject of this dissertation, it is worth mentioning that the *Mobile Millennium* system provides a framework for comparing data from multiple sources and validating the accuracy of those sources. For example, one way to validate the filtering of sparse GPS data is through the use of high-precision, high-frequency GPS devices, which have been used by drivers hired by the project and for which the correct path can be determined with certainty. By down-sampling this high accuracy data to the level typically received from sparse GPS sources, the reconstruction of the path from the sparse data can be compared with the true known path.

2.2.3 Scalability

Traffic systems are required to produce estimates across the entire network, continually updating themselves by processing thousands of new data records every few seconds. If scalability is not taken into account in the system design, it is quite likely that the system will not be able to keep up with all of the streaming data in real-time. The *Mobile Millennium* system was designed to be modular, so that each component of the system can run independently, potentially on its own server. This means that the various processes (from collecting raw data, filtering, running the estimation algorithms, disseminating estimations to third parties, visualization, validation, monitoring, etc.) can be divided up among all of the server resources available.

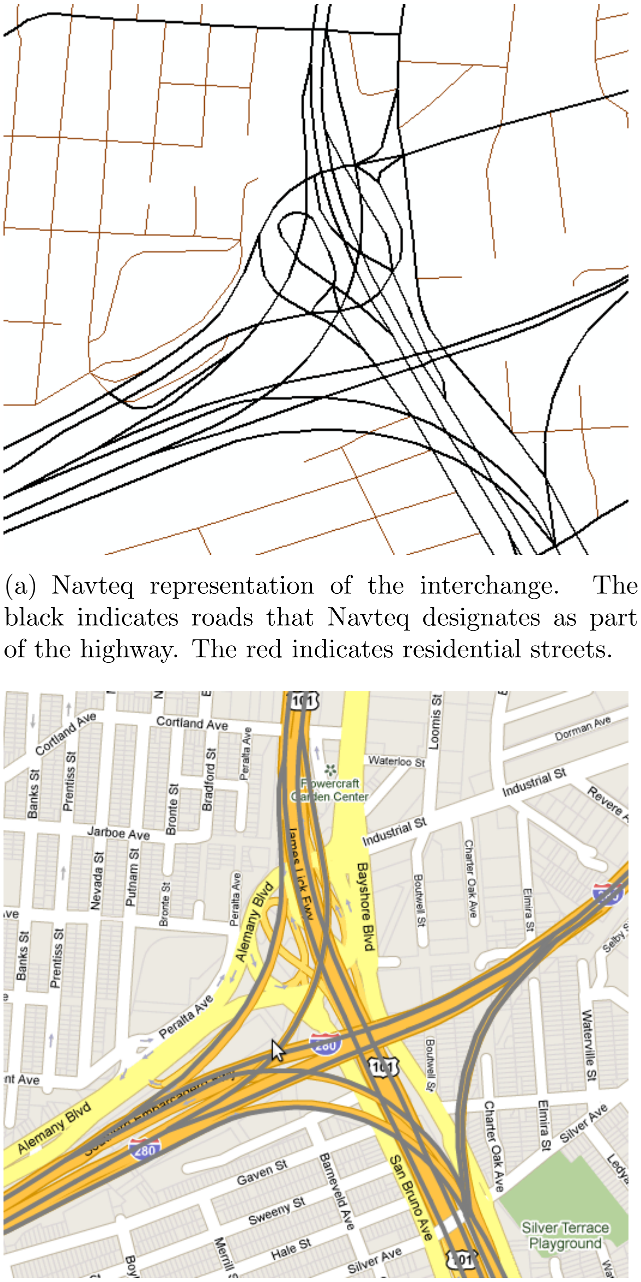
Beyond the modular design, it is also important to design estimation algorithms that scale in a reasonable way with the size of the raw data and the size of the network. In fact, a natural way of modeling the arterial network from processing sparse probe data requires solving a non-linear network optimization problem that grows quadratically in the number of links in the network. Given that a typical large city (such as San Francisco) may have on the order of 2,500 links, solving this optimization problem in real-time quickly becomes impractical. For this reason, the algorithms presented later in this dissertation (in chapters 5 and 6) are designed so that the optimization problem is linear in the number of links and therefore computationally tractable.

2.2.4 Network Abstraction

The most fundamental, core piece of any traffic information system is the digital representation of the road network. All data is associated with some location and needs to be mapped precisely. Estimation models need precise information about the geometry and physical characteristics of the road network. Separate components of the system need to be able to communicate information about location information in a universally consistent manner. For these reasons, it is critical to build the traffic system starting with a good digital map of the roadway. A digital map is constructed using the common graph theory notions of *nodes* and *links*. A link is the stretch of road between two nodes and a node represents the intersection of multiple links. The digital map must include the geometry (i.e. the latitude/longitude coordinates) of each link and will generally contain a number of road attributes, such as the number of lanes, the speed limit, the road type (such as highway, arterial, or ramp), the name of the road, etc.

The *Mobile Millennium* system was built using Navteq maps [9] as the underlying representation of the road network. Navteq maps (see figures 2.4 and 2.5) were made available to the *Mobile Millennium* team as part of the Safe Trip 21 project, which was funded by the United States Department of Transportation [18] and the California Department of Transportation [3]. Navteq maps provide detailed geometry and numerous road attributes per link (over 100). Traffic models generally assume a directed graph representation of the network and the Navteq map goes beyond this in terms of the complexity of its digital representation. Figure 2.4(a) illustrates the level of complexity around an interchange in San Francisco, CA. For highway traffic estimation algorithms, the key pieces of information are simply the points where roads merge or diverge (the nodes of the directed graph) as well as where the on-ramps and off-ramps allow for entering and exiting the highway. The added detail in geometry for the ramps and approaches to the highway add unnecessary complexity to the model and it is therefore better to remove those. The result of the network abstraction algorithm for this part of the highway is shown in figure 2.4(b).

Arterial networks experience a different type of problem with the road network representation, which is illustrated in figure 2.5(a). The key issue here is that Navteq represents nodes of the graph by a single GPS point with no area and they also often represent each direction of traffic by a separate link. When two roads intersect as in the figure, four "short" links are created a result of the two links for each direction intersecting. These four links are not "real" links, but rather just artifacts of their choice of how to draw the road network. Traffic estimation models assume that this is just one intersection and should be represented



(a) Navteq representation of the interchange. The black indicates roads that Navteq designates as part of the highway. The red indicates residential streets.

(b) Simplified representation (in gray) of the interchange for traffic estimation algorithms overlayed on a Google map of the area. Although not pictured, the estimation algorithms are aware of the incoming and outgoing ramps, allowing them to account for incoming and exiting traffic while not worrying about modeling the ramp traffic conditions explicitly.

Figure 2.4: Navteq (a) and simplified (b) representations of the I-280, highway 101 interchange in San Francisco, CA.

as a single node in the graph with all links connecting to it. The result of the network abstraction algorithm is presented in figure 2.5(b). This simplified representation helps for both the traffic estimation algorithms as well as the map matching and path inference filters.

The final component of the network abstraction algorithm for both highways and arterials is that of link selection and merging. Link selection simply refers to the fact that not all roads should be considered for traffic estimation, particularly residential roads. For highway networks, all nodes that have exactly one incoming and one outgoing link are removed since these are unnecessary for traffic estimation (these nodes often exist in the original map to denote the presence of a physical sign along the side of the road). For arterial networks, traffic estimation algorithms operate on links defined by the stretch of road between signalized intersections (or intersections with stop signs). When a node has only one incoming and one outgoing link and there is no traffic signal or stop sign at this node, then the links are merged together.

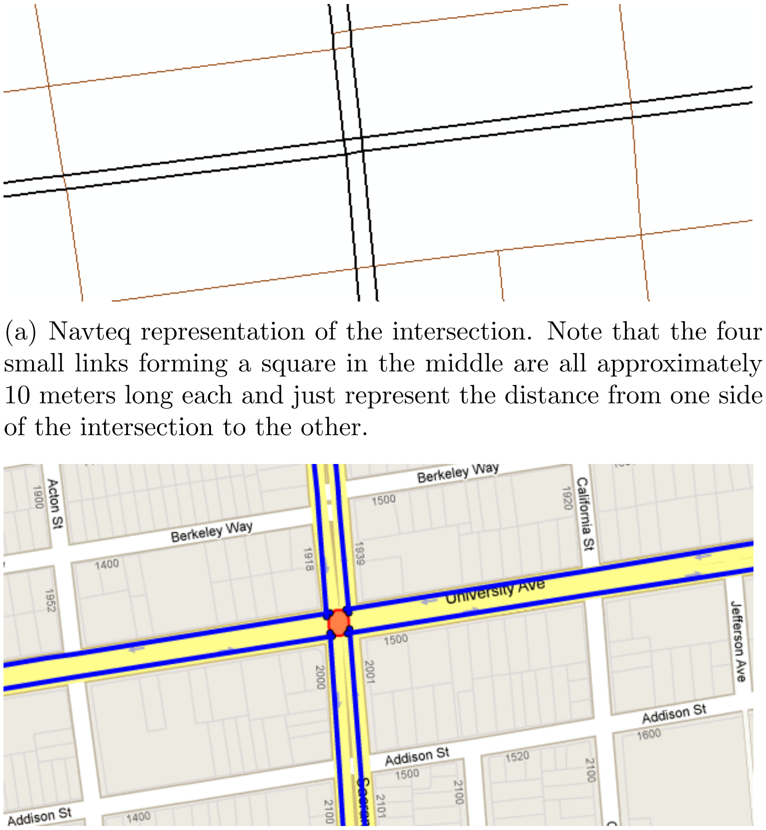
The result of this network abstraction procedure is a unified representation of the roadway that all components of the system can use to communicate location information. It also allows for intuitive visualization of the output of the traffic estimation models.

2.2.5 Map Matching and Path Inference

Probe data from GPS devices is often very accurate and easy to place on the digital map of the road network. However, there is enough noise in the data that there are several situations that frequently arise that make directly inferring the correct mapping difficult. One situation that is difficult to deal with on highways is when a frontage road is very close to the highway. In this situation, it can be difficult to distinguish which of the two roads the driver was on. Another difficult situation to deal with on arterials is when an observation occurs directly in the middle of an intersection (like that in figure 2.5). Furthermore, when a vehicle is transmitting its GPS position infrequently, the number of turns made between measurement locations could make it difficult to determine the correct path taken between successive measurements.

To address these difficulties, the *Mobile Millennium* system developed an algorithm that simultaneously performs map matching and path inference for both sparsely-sampled and high-frequency GPS probe data. The map matching component is performed using a spatial database capable of performing *spatially-indexed* queries (which are performed by a PostgreSQL database inside the *Mobile Millennium* system [14]). This speeds up the map matching process by several orders of magnitude by localizing the search for possible mappings to the set of nearby links. Several possible mappings are returned by the first stage of the map matching procedure. The second stage of the algorithm looks at all of the realistic paths between pairs of GPS measurements and determines the most probable path (which includes the most probable mappings on each end) [59]. Figure 2.6 shows a small sample of GPS points (hollow circles) along with the inferred mapping and traveled path.

Fixed-location sensors also require map matching, although the task is generally much



(a) Navteq representation of the intersection. Note that the four small links forming a square in the middle are all approximately 10 meters long each and just represent the distance from one side of the intersection to the other.

(b) Simplified representation of the intersection. The small links have been replaced by an intersection object that has a positive area. All of the connecting links connect only to this one intersection object instead of to each of the "short" links.

Figure 2.5: Navteq (a) and simplified (b) representations of an arterial intersection in Berkeley, CA. The intersection is represented by 4 "short" links in the Navteq database, but for traffic estimation it is more appropriate to have a single intersection object.

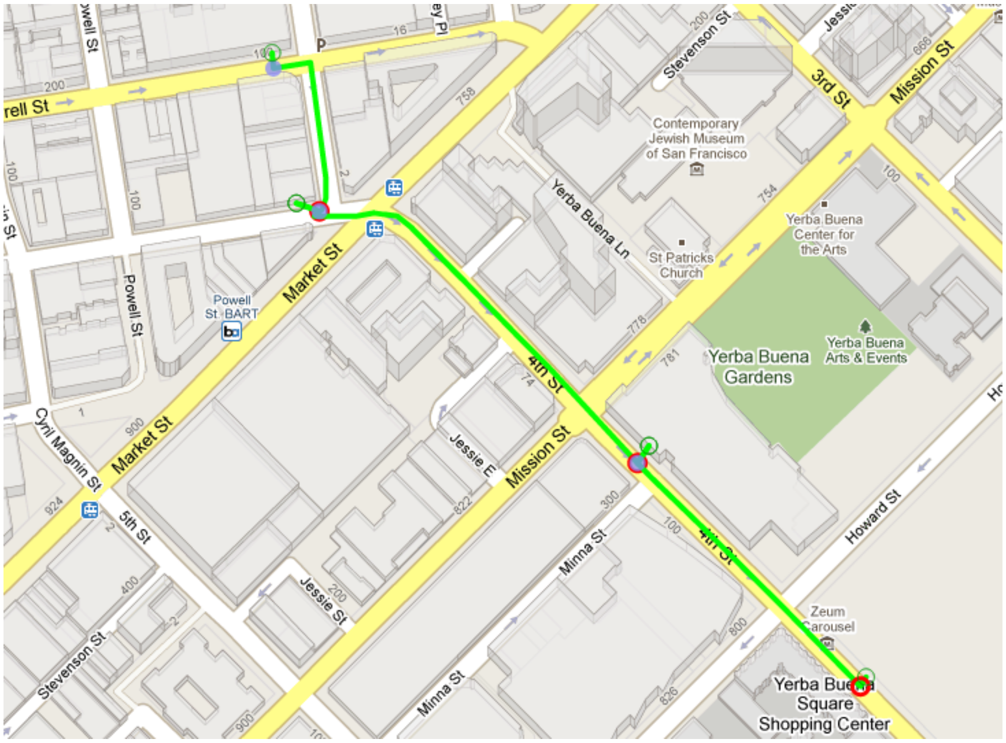


Figure 2.6: An illustration of the *Mobile Millennium* map matching and path inference algorithm. The hollow circles represent the GPS measurement locations. The blue/red circles represent the start/end of a pair of GPS points as mapped to the road. The green lines indicate the inferred path traveled by the probe vehicle.

easier than for GPS data. For these types of sensors, a spatial database is again required to identify the closest links to the GPS location of the sensor (which is generally how fixed-location sensors are identified). The GPS location often comes with a description of the location as text and this text is used in the case where the GPS location is close to several possible links. In that situation, the text acts as a discriminator for choosing the correct mapping.

2.2.6 Visualization

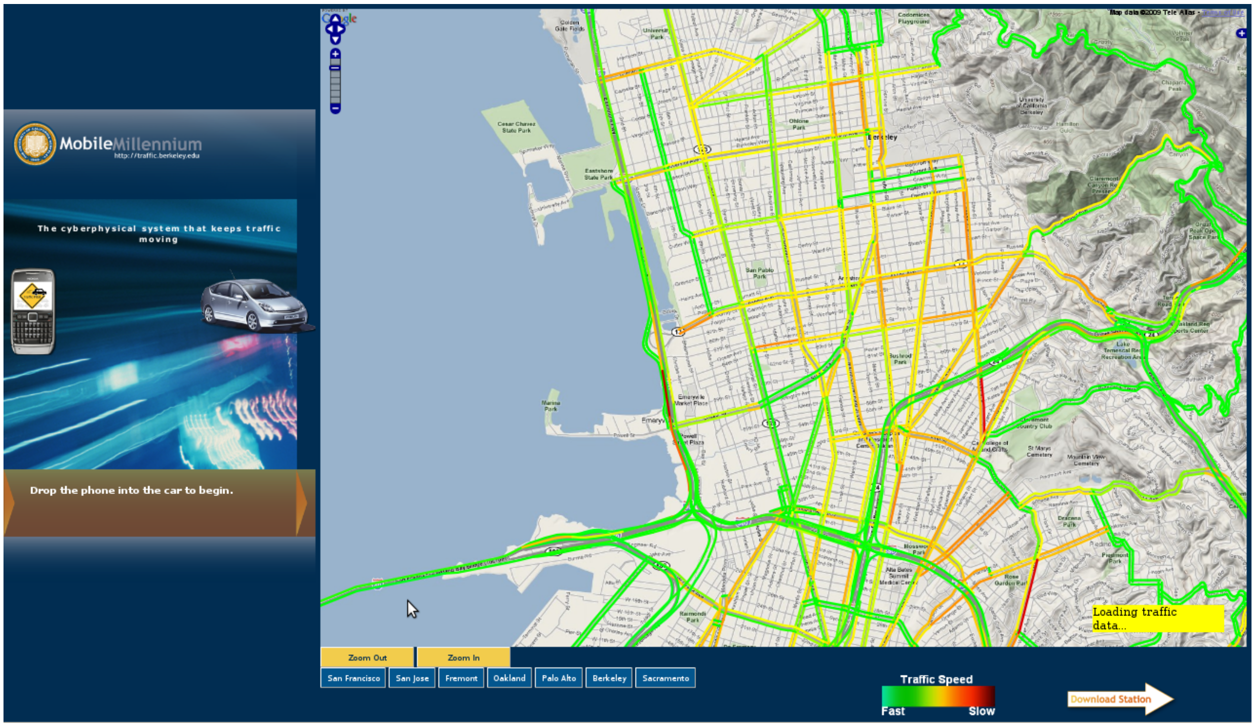
At the tail end of any traffic information system, there must be some way to visualize and interpret the results of the traffic estimation algorithms and routing services. The "color-coded" map has become the standard way to quickly disseminate real-time traffic estimates to a wide audience. A color map allows anyone to quickly spot the high congestion areas on the route of interest. In addition to the color map, it is also important to visualize other key pieces of information. Displaying travel times along different route choices between the same origin and destination pair allows drivers to quickly choose the right one. An example of the *Mobile Millennium* system visualizer, which is the public output of the system, is presented in figure 2.7.

In addition to the final output, it is also critical to visualize intermediate components of the traffic estimation process. The *Mobile Millennium* system developed both an internal and external visualizer to allow researchers and the public to view traffic information on a map easily. The internal version of the visualizer allows for detailed insight into how models are producing the estimates. It also allows the researcher to overlay multiple sets of information at once such as fixed sensor locations, portions of GPS traces, or accident information. These visualization tools have become a staple of the *Mobile Millennium* research team and have vastly improved the rate of progress of algorithm development. Figure 2.8 shows an example of some of the layers that are available inside the visualizer (current highway traffic estimates and PeMS loop detector locations).

2.2.7 Mobile Client

An increasingly important part of traffic information dissemination is through cell phones. The use of cell phones as part of traffic information systems was the primary inspiration for the *Mobile Millennium* system, which is described in more detail in section 2.3. Given that the amount of computing power, communication and sensing capabilities in phones is constantly growing, smartphones will continue to be of great value for both providing raw data and for drivers to see real-time conditions while driving.

As part of the *Mobile Millennium* project, Nokia built the first traffic monitoring mobile client that ran on their N95 and E71 series phones (shown in figure 2.9). The requirements of the client were to use the VTL system infrastructure that they had built and also to be able to display live traffic conditions via a color map directly on the phone. This allowed

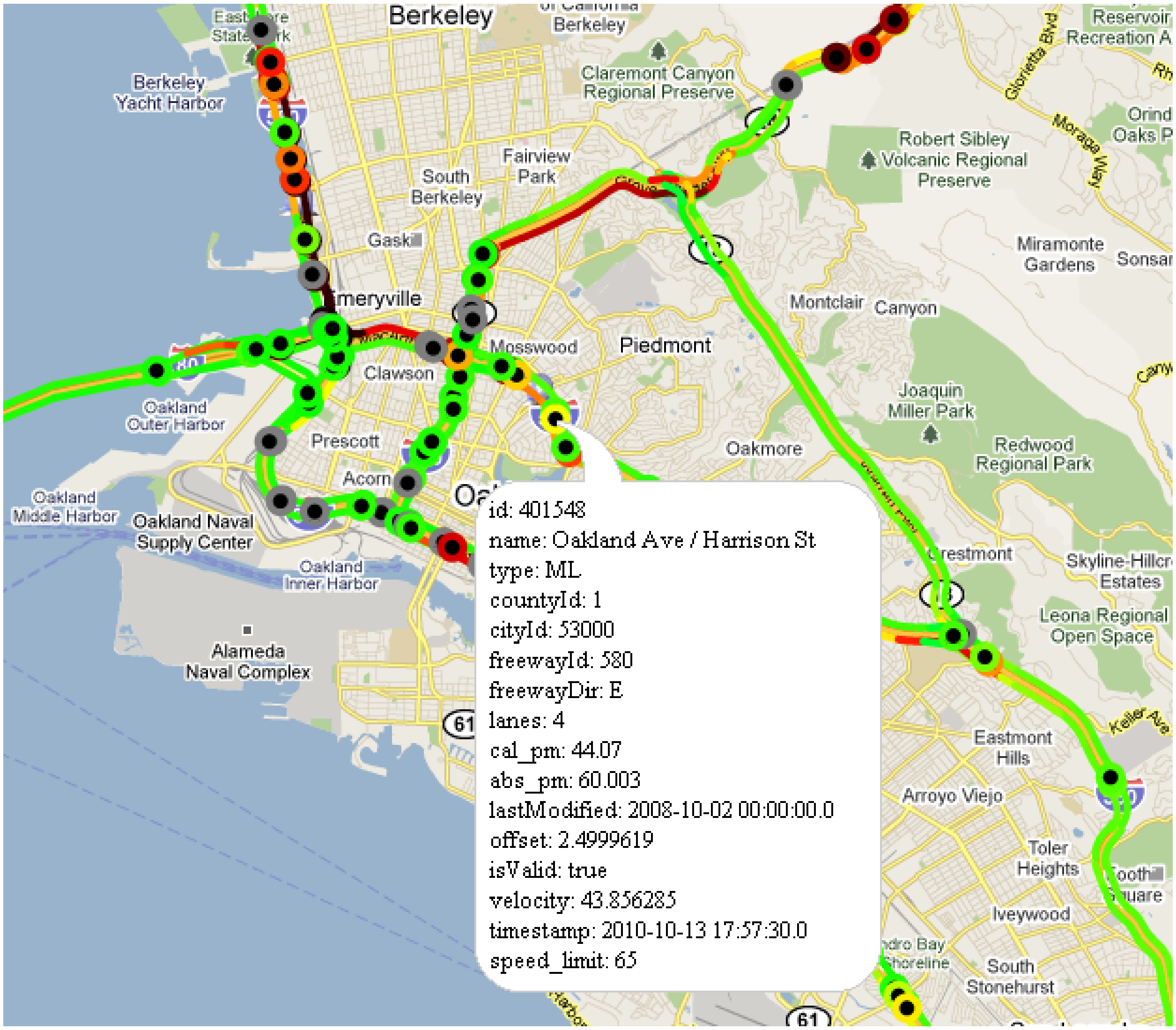


The cyberphysical system that keeps traffic moving

Drop the phone into the car to begin.

Loading traffic data...

Figure 2.7: *Mobile Millennium* public visualizer showing real-time traffic conditions in Berkeley and Oakland, CA.



id: 401548
name: Oakland Ave / Harrison St
type: ML
countyId: 1
cityId: 53000
freewayId: 580
freewayDir: E
lanes: 4
cal_pm: 44.07
abs_pm: 60.003
lastModified: 2008-10-02 00:00:00.0
offset: 2.4999619
isValid: true
velocity: 43.856285
timestamp: 2010-10-13 17:57:30.0
speed_limit: 65

Figure 2.8: *Mobile Millennium* internal visualizer showing model outputs and locations of PeMS loop detectors (hollow circles).

drivers to see current traffic information while providing data to the system, through the use of VTLs.

2.2.8 Sensor Deployment

Traffic information systems that use fixed-location sensors as the primary data source inevitably have the problem of *where* and *how many* sensors to deploy. Developing optimal deployment strategies is crucial for public transit agencies building and operating a traffic information system at minimal cost. Historically, sensors have been placed using "rules of thumb" such as every half mile or every third of a mile as is done in different parts of the California highway network [25]. Indirectly related to this dissertation, an optimization algorithm was developed for optimally placing sensors on highways. This work is presented in appendix A.

2.3 A specific Berkeley Prototype: the *Mobile Millennium* System

One of the primary contributions of this dissertation has been the design, implementation and testing of the *Mobile Millennium* system. This section presents an overview of the project and highlights the important design decisions that led to the successful execution of the project. These design decisions include using a database-centric approach as opposed to a data queue approach as well as using flexible modules rather than dependently linked processes. The design of the system has allowed for easy scalability as well as the introduction of entirely new research areas to be built into the system seamlessly.

2.3.1 History of the Project

Mobile Millennium began immediately following the successful *Mobile Century* experiment on February 8, 2008 [52]. The initial stated goals of the project were to build a fully operational traffic information system using VTL-based sensing from individual cell phones. The initial partnership was primarily between the Nokia Research Center in Palo Alto, CA and UC Berkeley. Nokia was responsible for developing the software application to go on individual cell phones as well as providing the VTL infrastructure for processing the raw data coming from the phones. UC Berkeley was responsible for building a system capable of using the raw VTL data as input into traffic models that would then output estimates and forecasts of traffic conditions along the major roads in northern California (including both highways and arterials). The traffic estimates were sent back to Nokia so they could be displayed on the cell phones that were running the same software application that was also providing VTL data. This was the first "participatory sensing" project ever for traffic estimation.

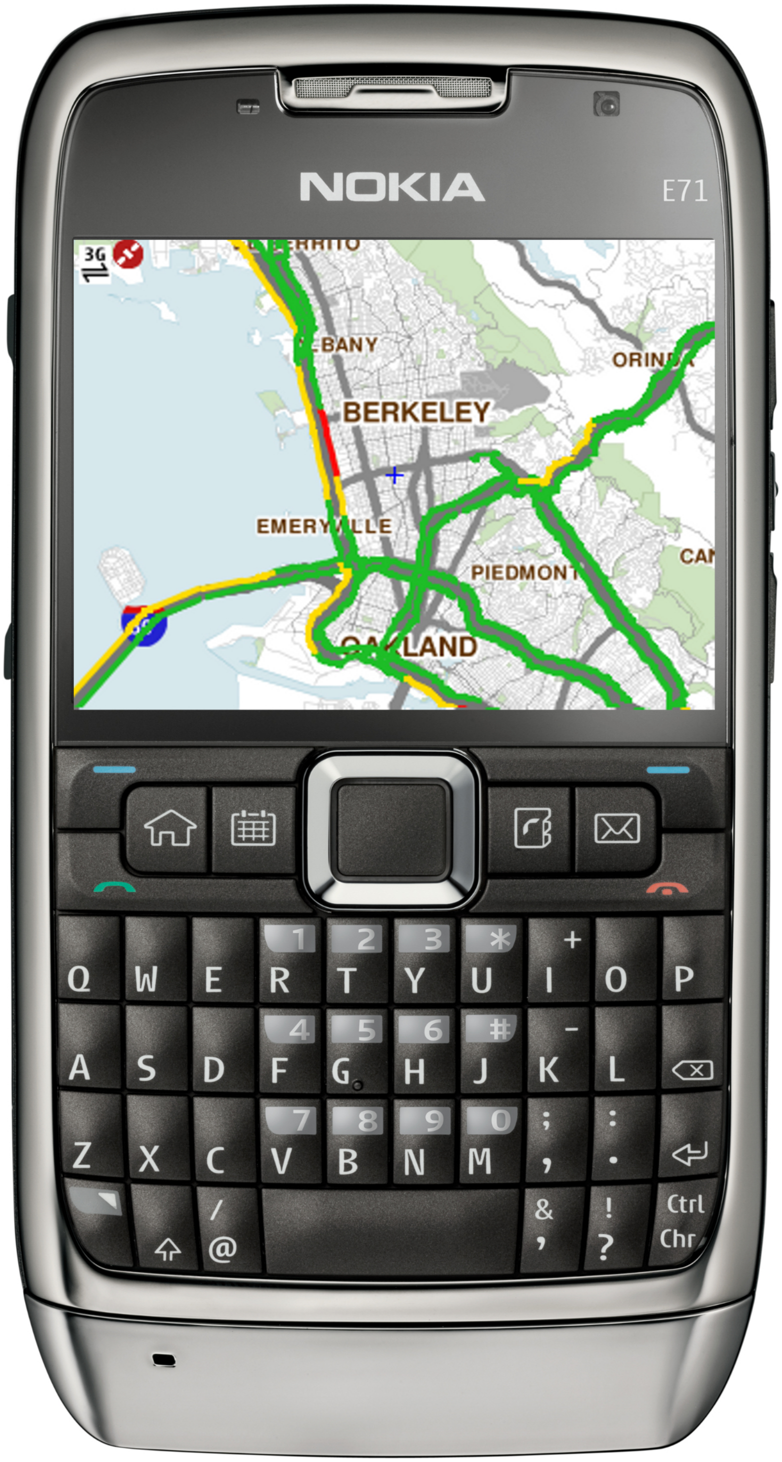


Figure 2.9: The *Mobile Millennium* phone client running on a Nokia E71. The phone client displays a color map of current traffic conditions around the driver's location while simultaneously providing VTL data to a central server.

On November 10, 2008, the official phone application was released to the public (shown in figure 2.9). There were more than $5,000$ downloads in the first few months, which provided a small amount of data on a daily basis in parts of the Bay Area and Sacramento. VTL data from the phones was supplemented with data from PeMS as well as Navteq radar data to feed the live *Mobile Millennium* system. The first demonstration of the systems capabilities occurred on November 18, 2008 when 20 drivers equipped with cell phones running the official phone application drove for 3 hours in Manhattan, New York. The VTL data was the only source of data available for that experiment and was relied upon entirely for estimating traffic conditions in real-time. The model estimates were displayed live for the attendees of the ITS World Congress that was taking place next to the experiment site. The location of the experiment is shown in figure 2.10.

Following the launch of the official *Mobile Millennium* phone application, the project continued to expand the volume of data sources and the sophistication of the traffic estimation models. The goals of the project expanded to include real-time routing algorithms and also for the system to become a central data collection point for many different traffic related data sources. Today, the *Mobile Millennium* system continues to expand its reach to new applications centered around data from mobile devices, including air quality estimation, river flow estimation, and earthquake detection.

2.3.2 System Architecture

There are multiple ways to look at the architecture of the *Mobile Millennium* system. Figure 2.11 illustrates the flow of information through the system from raw data to useful information. In the *Mobile Millennium* system, raw data always goes through at least one filter before being delivered to the models and estimation algorithms. The output of these models is used in a number of applications before being sent to third parties for consumption or analysis. Underneath the flow of data through the system are several components needed for quality analysis and visualization of each step of the process. With this in mind, the *Mobile Millennium* team built an evaluation framework and internal visualizer for comparing and analyzing data from multiple sources using several quality metrics. These allow for quick checking of the data through all steps of the process, from raw data to filtered data to model outputs.

Another way of looking at the *Mobile Millennium* system is depicted in figure 2.12. This figure illustrates the way the components of the system interact. In general, the database is the central point for communication between processes. This allows for a modular system where components can function independently without worrying about if another component fails. The system software was designed so that one core module directly interacts with the database and requires that all requests to receive or send data are passed through that module. There were two members of the *Mobile Millennium* team responsible[1] for maintain-

[1] The two team members responsible for the core module were Ryan Herring and Saneesh Apte.

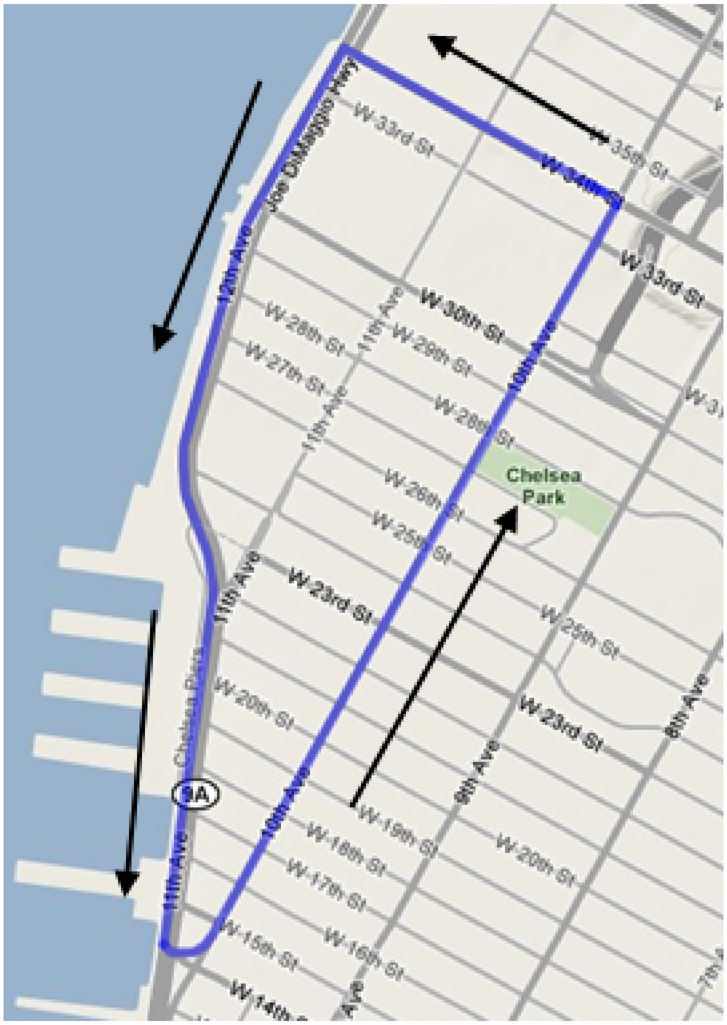


Figure 2.10: Site of the first *Mobile Millennium* system demonstration in Manhattan, New York on November 18, 2008.

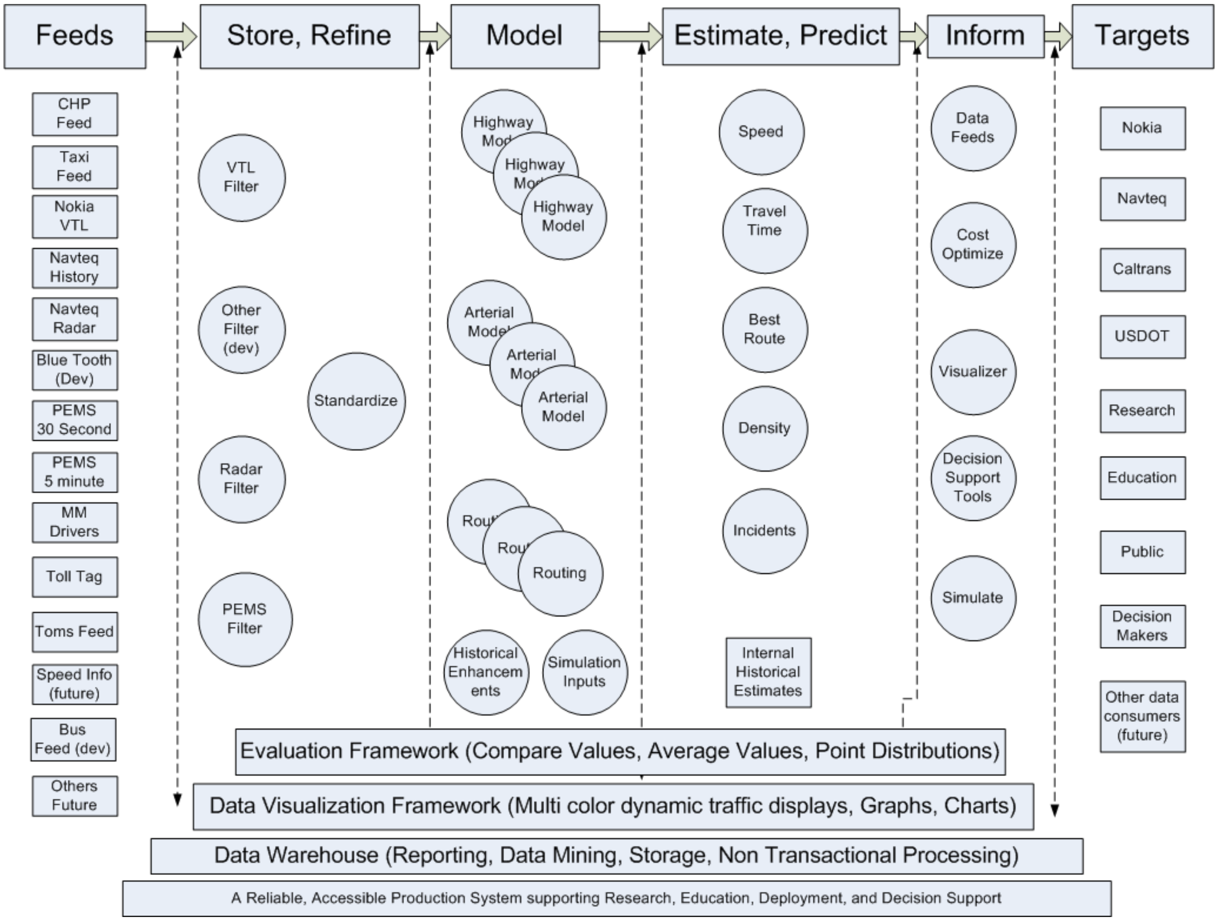


Figure 2.11: An overview of the *Mobile Millennium* system.

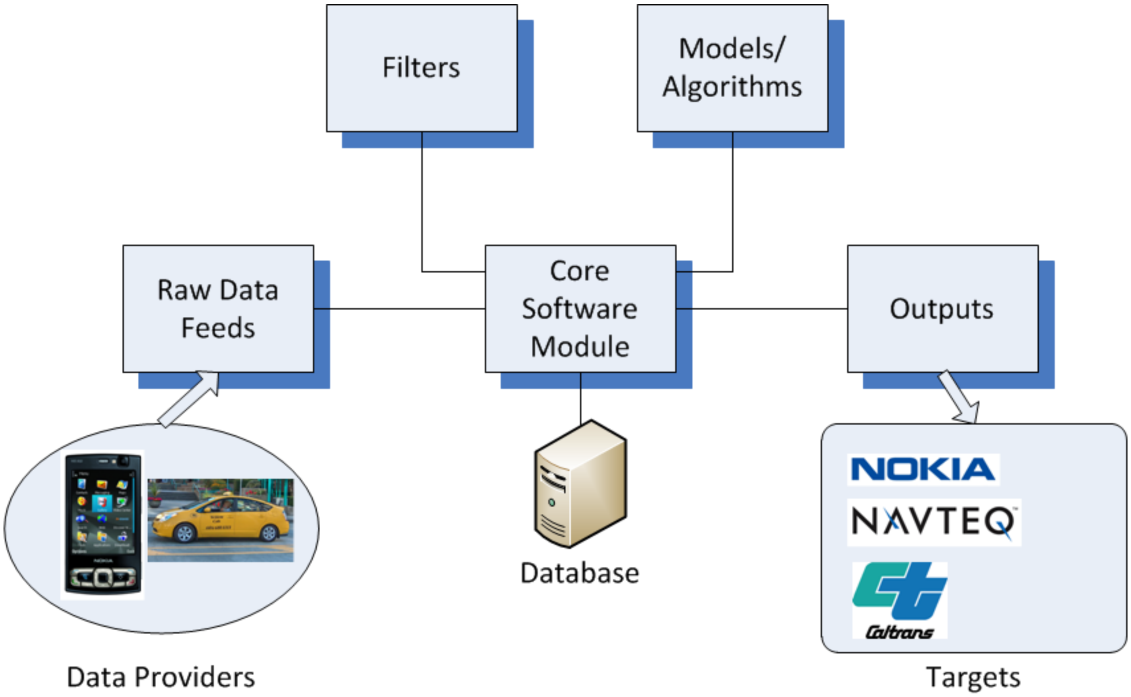


Figure 2.12: *Mobile Millennium* system database-centric architecture.

ing that core module and adding new functionality with regards to reading or writing data. While this places a burden on the people responsible for the core module, it ensures that access to the database is done in a consistent way, which prevents instabilities from occurring. This decision also enabled the project to bring inexperienced student programmers in and have them contribute quickly without having to learn the details of the database design. The core module includes the following basic functions:

- Accessing a simplified representation of the road network for any geographic area of interest. (See section 2.2.4 for more details on how this network representation is created.)
- Accessing all raw and filtered data.
- Writing all data or model estimates.

The disadvantages of this database-centric design are that the input/output bandwidth needs to be high enough to handle all of these requests with minimal delay and if there is any problem with the database, it affects all other processes. The input/output bandwidth constraint is handled by having high-powered machines hooked up together using direct gigabit ethernet links. In general, the importance of the database has led to the team focusing on ways of optimizing database performance and keeping a close eye on database maintenance.

2.3.3 Database Design

The requirements for the database software for the *Mobile Millennium* system were that it be open-source (due to budget constraints) and that it have spatial features (because of the need to do spatial queries on GPS data). These two requirements led to choosing PostgreSQL [14] as the database software with the PostGIS extensions [13] for spatial queries.

The *Mobile Millennium* database's design is centered around the map data provided to the project by Navteq. This data provides the underlying structure of the road network along with a number of key attributes (e.g. lanes, speed limit, etc.). The simplified network construction described in section 2.2.4 (called the "Model Graph" in the *Mobile Millennium* system) sits on top of the Navteq map data, meaning that there is a direct relationship from the Model Graph links back to the original Navteq links (known as a foreign key relationship in database terminology). This structure allows any data that is mapped to a Navteq link to be easily converted into a mapping on the associated Model Graph link or vice versa. This is important as the system follows the general rule that raw data sources are mapped to the Navteq map, whereas the traffic estimation models use the Model Graph for inputs and outputs. Translating back and forth is a process that needs to be able to occur very quickly with no delay and the design of the database tables relating the Navteq map and the Model Graph make this possible.

Figure 2.13 displays the important tables of the database for the core infrastructure as well as a few example modules that use it (the full database structure is far too big to put in a single graphic). This figure emphasizes the design decision to have raw data feeds dependent upon the Navteq map and the models dependent upon the Model Graph. As the system evolves, new data feeds and traffic estimation models will follow that same pattern.

2.3.4 System Modules

The *Mobile Millennium* system modules can be divided into 4 broad categories: raw data feeds, raw data filters, estimation models/algorithms, and output handlers. Each type of module interacts with the core software module responsible for interacting with the database. In this way, each of these modules in similar, but each comes with a few specific considerations, which are detailed here. All of these modules are monitored in real-time with alerts sent to key members of the team if any problem occurs.

Raw Data Feeds

All raw data feeds in the *Mobile Millennium* system are constructed using the same principles. First, care is taken to ensure that no raw data is missed in transmission from

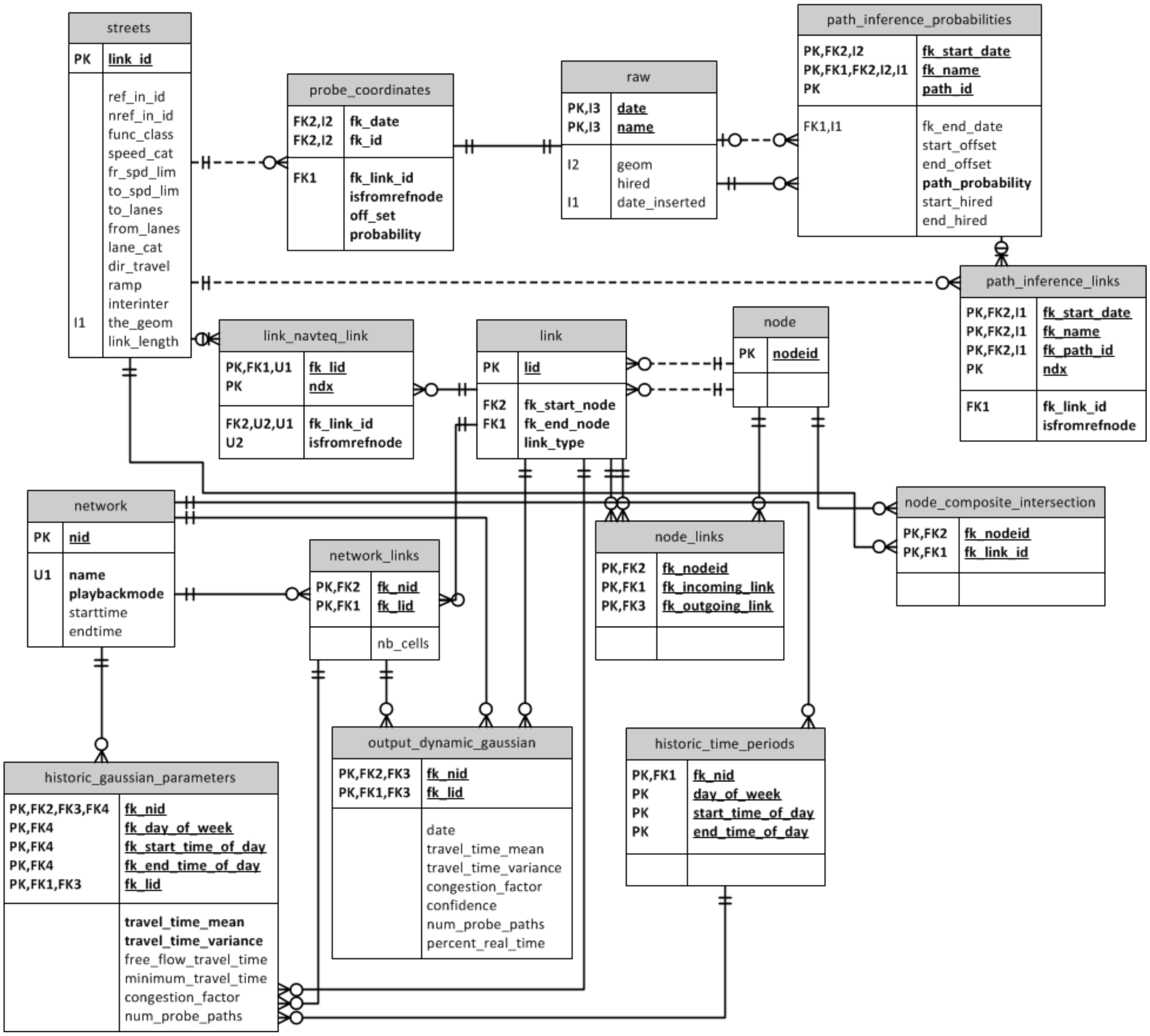


Figure 2.13: An illustration of the relationship between the core tables and a few auxiliary modules of the *Mobile Millennium* database. The symbols between tables represent the relationship between data structures (such as one-to-one, many-to-one, etc.).

the source. Most of the feeds into the system are of the *pull* variety, meaning that a *Mobile Millennium* server queries the data provider every fixed number of seconds for the latest data. This requires checking for duplicates as well as using a rolling window to make sure any late data is still picked up. Some data feeds are *pushed* to the system, meaning that the system must have a process in place to handle any data that is sent by the data provider. This requires enough bandwidth to process any data that is sent without adding delay to the system.

Filters

There are two general classes of filters in the *Mobile Millennium* system and both must satisfy the requirements of locating the data on the map and running with minimal delay so as to provide the models with the data as soon as possible. One filter class encompasses data coming from fixed-location sensors such as loop detectors or radar, the other class is for GPS probe data. The first type of filter requires a static mapping component which places each fixed sensor on the Navteq map and a dynamic component that must process real-time data as quickly after it arrives as possible. These class of filters rely on a fixed map matching procedure developed for fixed sensor data and then a custom filter is built to handle the specifics of the raw data arriving. In terms of the database design, there is a cost savings by having the map matching done once and then only needing to reference a sensor identification number when processing real-time data.

The second class of filters for GPS data requires both filtering in the more traditional sense of the word (removing outliers, smoothing, etc.) as well as map matching for every data point that arrives. This type of filter is much more computationally intensive than the first class as performing map matching is a time consuming path because it relies on a spatial query to the database for each GPS point. This type of filter has been implemented in such a way as to leverage parallel computing technology when it becomes available to the team in the coming months. Choosing this design strategy means that this computationally intensive filter can scale well when the volume of data substantially increases as is expected in the coming years.

Models/Algorithms

The models and algorithms modules in the *Mobile Millennium* system are the most important components of the system from a scientific point of view. They represent the implementation of research ideas that often take years to put into production. Since they are generally significantly bigger modules than the other types, much of the focus of the team is on how to make them run as smoothly as possible. There are two requirements that models must adhere to in the *Mobile Millennium* system. First, it is necessary that any estimation algorithm run fast enough to be used in real-time. This means that the algorithm itself needs to be computationally efficient, but also means that the systems team needs to ensure that the algorithm has the server resources necessary to run at fast speeds. The second requirement is that all inputs and outputs of the algorithm go through the core software module, which ensures a smooth interaction with the database.

Outputs

The outputs of the system are occasionally directed outward, as is the case when data is sent to Nokia for real-time visualization on cell phones, but these outputs can also be directed toward further research and analysis as well. The team designed a evaluation framework and visualizer to take advantage of the outputs of the models and both of these again rely on the database as the central point of communication. This places a few requirements on the output processes, whose job it is to take the raw model outputs and convert them into whatever form is expected for analysis, visualization or end-user information. Some transformation is always necessary for presenting the model outputs in the correct format, whether it be encoding velocity values in a color scale on a map or computing estimated travel times from a dynamic velocity field. It is precisely the job of the output modules to adapt to whatever the natural outputs of the models are and transform them into the format expected by the final targets.

2.3.5 Field Experiments

Numerous field experiments have been conducted to test the *Mobile Millennium* system. Two that were already mentioned were the original *Mobile Century* experiment in the East Bay, CA followed by the first true *Mobile Millennium* experiment in Manhattan. The following additional experiments were also run:

Three experiments in June/July 2008 in Berkeley, CA. These experiments were run on a small set of arterial roads in Berkeley as preparation for the Manhattan experiment that was run later in 2008. The primary question of interest to be answered by this set of experiments was if the GPS in the phones could provide good enough data on arterials. Additionally, the experiment was designed to specifically study the travel patterns through one of Berkeley's busiest intersections at San Pablo Avenue and University Avenue. Each experiment involved 20 vehicles driving several loops through Berkeley, with all 20 vehicles traversing the segment of University Avenue going west through the San Pablo Avenue intersection. Each test lasted approximately two hours.

Three experiments April 27-29, 2010 in the East Bay, CA. Along with the following set of experiments, this set was designed to be the official test of the production version of the *Mobile Millennium* arterial model. These experiments each had 20 drivers (although data from some of the vehicles was never captured). The goal of this set of experiments was to estimate travel times along a 2.3 mile stretch of San Pablo Avenue that went through Berkeley, Albany and El Cerrito, CA during a three-hour time period. Bluetooth readers collected data intended to be used as ground truth, although the data was not accurate enough to be considered ground truth.

Three experiments June 29-July 1, 2010 in San Francisco, CA. The other half of the official *Mobile Millennium* evaluation. The goal of this set of experiments was to estimate travel times on a 1.1 mile stretch of 3 parallel roads (Van Ness Avenue, Franklin Street, and Gough Street) over a three-hour time period. 20 vehicles were used in each experiment, split into two loops. Bluetooth readers were used again to collect ground truth, although the same issues prevent it from being considered true ground truth data.

The set of data collected as a result of all of these experiments is large and it has proven to be valuable for research conducted by the *Mobile Millennium* team. It can be argued that the most valuable assets of the entire *Mobile Millennium* project have been the construction of the system described in section 2.3.2 and the set of data collected described above. The combination of data and robust system tools make it easy for new researchers that join the project to get started quickly analyzing data and trying out new algorithms, which was one of the primary initial goals of the project.

Chapter 3

Literature Review and Background Material

The major contribution of this dissertation is the creation of algorithms which fuse together previously disparate disciplines into a single unified framework. Specifically, the goal of the work is to leverage results in traffic flow theory, machine learning theory, sensor networks, and estimation techniques to provide traffic estimation algorithms capable of processing heterogeneous sources of data.

Given the hybrid nature of this work, it is necessary to give some background on several subjects before diving into the contributions of this work. Sections 3.1 and 3.2 introduce the fundamental concepts from traffic flow theory and machine learning, respectively, that will be used throughout the rest of this dissertation. While this dissertation focuses primarily on arterial traffic estimation, it is relevant to provide some background on highway traffic estimation as this subject was studied first and is related in terms of the final goal (i.e. estimating traffic conditions). Section 3.3 provides an overview of highway traffic estimation techniques using data from both fixed sensors and GPS probe vehicles. This then leads to a review of the other techniques that have previously been proposed for estimating arterial traffic conditions, presented in section 3.4. Previous work on arterial traffic estimation has varied greatly in the methodology and the data used. Elements from these previous efforts have provided inspiration for some of the ideas of this dissertation.

3.1 Traffic Flow Theory

In traffic flow theory, it is common to model vehicular flow as a continuum and represent it with macroscopic variables of *flow* $q(x,t)$ (veh/s), *density* $\rho(x,t)$ (veh/m) and *velocity* $v(x,t)$ (m/s). The definition of flow gives the following relation between these three variables [68, 82]:

$$q(x,t) = \rho(x,t)\, v(x,t). \tag{3.1}$$

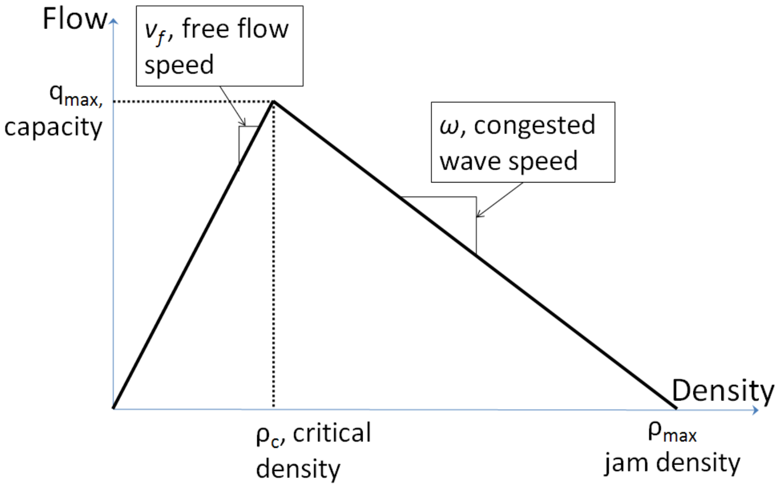


Figure 3.1: The fundamental diagram: empirically constructed relation between flow and density of vehicles.

This property will be used throughout the derivations of traffic probability distributions (chapter 4).

For low values of density, experimental data show that the velocity of traffic is relatively insensitive to the density; and all vehicles travel close to the so called free flow velocity, v_f, of the corresponding road segment. As density increases, there is a critical density ρ_c at which the flow of vehicles reaches the capacity $q_{\max}$ of the road. As the density of vehicles increases beyond ρ_c, the velocity decreases monotonically until reaching zero at the maximal density $\rho_{\max}$. The maximal density can be thought of as the maximum number of vehicles that can physically fit per unit length, and at this density, vehicles are unable to move without additional space between vehicles. Experimental data indicates a decreasing linear relationship between flow and density, as the density increases beyond ρ_c. The slope of this line is referred to as the congested wave speed, noted w. This leads to the common assumption of a triangular *fundamental diagram* (FD) to model traffic flow dynamics [39].

The triangular FD (illustrated in figure 3.1) is thus 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).

Note that ρ_c represents the boundary density value between (i) free flowing conditions for which cars have the same velocity and do not interact and (ii) saturated conditions for which the density of vehicles forces them to slow down and the flow to decrease. When a queue dissipates, vehicles are released from the queue with the maximum flow—capacity $q_{\max}$—which corresponds to the critical density $\rho_c = q_{\max}/v_f$.

For a given road segment of interest, the arrival rate at time t, i.e. the flow of vehicles entering the link at t, is denoted by $q_a(t)$. Conservation of flow relates it to the arrival density $\rho_a(t) = q_a(t)/v_f$.

In arterial networks, traffic is driven by the formation and the dissipation of queues at intersections. The dynamics of queues are characterized by shocks, which are formed at the interface of traffic flows with two different densities.

Two discrete traffic regimes are considered: *undersaturated* and *congested*, which 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. Figure 3.2 illustrates these two regimes under the assumption of the triangular fundamental diagram. The speed of formation and dissolution of the queue are respectively noted v_a and w. Their expression is derived from the Rankine-Hugoniot [42] jump conditions and given by

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

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 called the maximum queue length, denoted $l_{\max}$, which can also be computed from traffic theory:

$$l_{\max} \;=\; R\frac{w v_a}{w - v_a} \;=\; R\frac{v_f}{\rho_{\max}}\frac{\rho_c \rho_a}{\rho_c - \rho_a}. \tag{3.2}$$

The duration between the time when the light turns green and the time when the queue fully dissipates is the *clearing time*, denoted τ, which is calculated as

$$\tau \;=\; l_{\max}\left(\frac{1}{w} + \frac{1}{v_f}\right). \tag{3.3}$$

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.

All notations introduced up to here are illustrated for both regimes in figure 3.2.

Stationarity of the two regimes. In studying statistical properties of the traffic regimes, it is often convenient to analyze their stationary behavior. A regime is defined to be *stationary* when the cycle time (C), red time (R), fundamental diagram parameters ($\rho_{\max}, \rho_c, v_f$), and arrival density (ρ_a) remain constant for a period of time. Stationarity implies that the queue evolutions are periodic (see figure 3.2). As indicated by the slopes of the trajectories in the figure, when vehicles enter the link, they travel at the free flow speed v_f. The distance between two vehicles is the inverse of the arrival density $1/\rho_a$. The time during which vehicles are stopped in the queue is represented by the horizontal line in the queue. The length of this line represents the delay experienced in the corresponding queue. The distance between vehicles stopped in the queue is the inverse of the maximum density $1/\rho_{\max}$. When the queue dissipates, vehicles are released with a speed v_f and a density ρ_c. The trajectory is represented by a line with slope v_f, the distance between two vehicles is $1/\rho_c$.

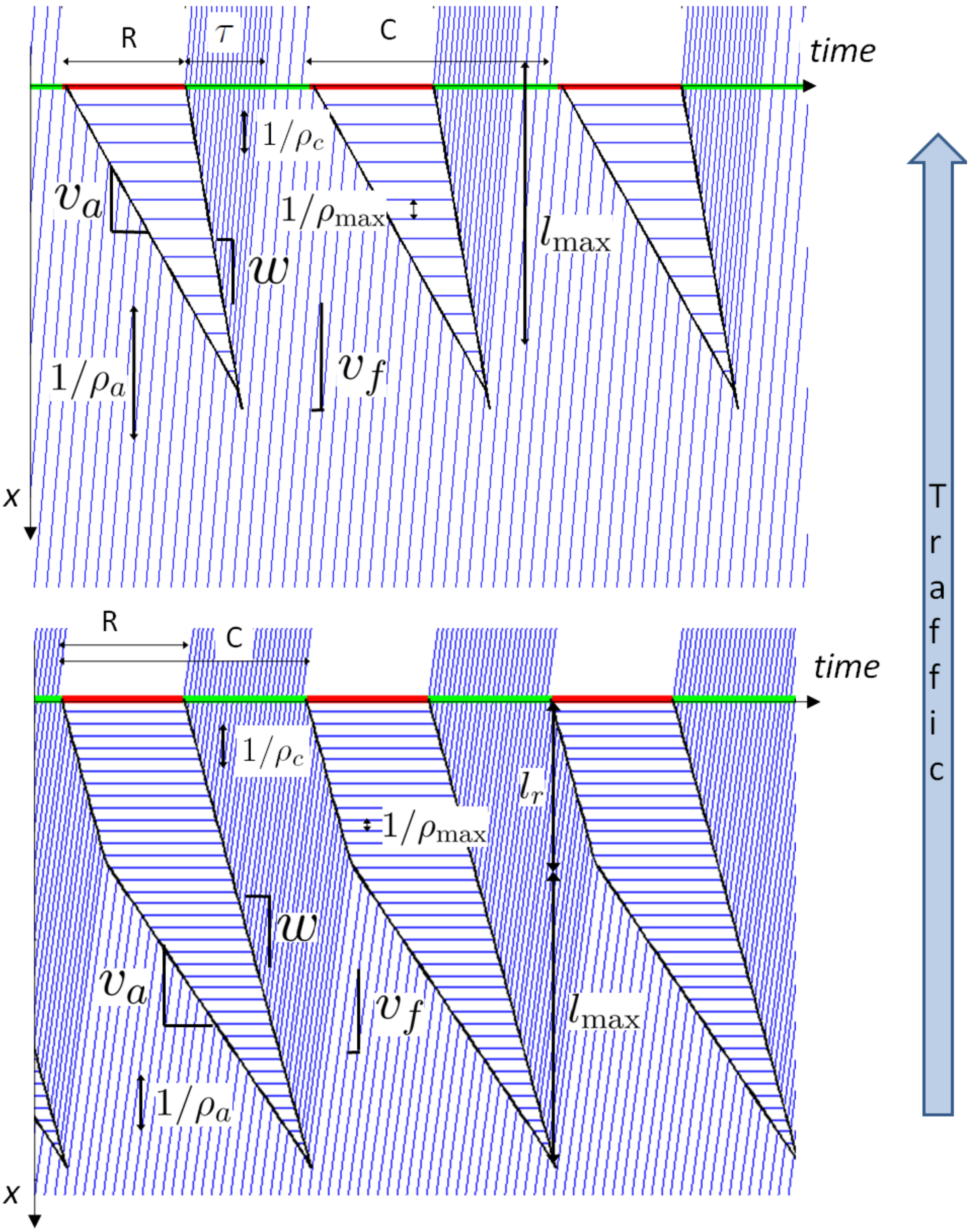


Figure 3.2: Space time diagram of vehicle trajectories with uniform arrivals under an undersaturated traffic regime (top) and a congested traffic regime (bottom).

3.2 Machine Learning/Probabilistic Graphical Models

This dissertation builds upon algorithms and techniques from machine learning and artificial intelligence. Two introductory books on these subjects are Russell/Norvig [85] and Hastie et al. [49]. The components of machine learning relevant to the problem of estimating arterial traffic are generally categorized as *Probabilistic Graphical Models* [94]. Graphical models provide a logical framework for analyzing complex dependencies between random variables in a system. They can be used in a variety of settings and one can find many examples in [94].

Starting with an overview of graphical models in section 3.2.1, the remainder of this chapter then focuses on the key ideas of statistical filtering (section 3.2.2) and the expectation maximization algorithm (section 3.2.3) used in the estimation algorithms presented in chapters 5 and 6.

3.2.1 Probabilistic Graphical Models

Graphical models can be defined in a very general setting. Consider a graph $G = (V, E)$, where V is a set of nodes and $E \subseteq V \times V$, which can be either directed or undirected. For each vertex $v \in V$ has an associated random variable X_v which can take values in some space $\mathcal{X}_v$ (continuous or discrete). In [94], the authors give an exhaustive introduction to the fundamental properties of a graphical model defined in this general setting. The main concept to note for our purposes is the notion of conditional independence. In an undirected graph setting, variables $X_v, X_u,\ v, u \in V$ are independent given variables $X_A,\ A \subset V$ if there are no paths between v and u that do not intersect some variable $\bar{v} \in A$. A similar definition exists for directed graphs replacing path with directed path. Conditional independence enables the joint distribution of all variables $X_v,\ v \in V$ to be written more compactly. For undirected graphs, it is written

$$p(x_1, x_2, \ldots, x_m) = \frac{1}{Z}\prod_{C \in \mathcal{C}} \psi_C(x_C), \tag{3.4}$$

where $\mathcal{C}$ is the set of all (maximal) cliques of the graph, ψ_C is a *compatibility function* defined on each (maximal) clique, x_u is a specific assignment of the random variable X_u for $u \in V$, and Z is a normalization constant. A clique is a complete subgraph, meaning that for the nodes in the clique, every possible edge between them exists. For directed graphs, it is written

$$p(x_1, x_2, \ldots, x_m) = \prod_{v \in V} p_v(x_v | x_{\pi(v)}), \tag{3.5}$$

where $\pi(v) \subseteq V$ represents the parents of $v \in V$. In both settings, a sparse graph means that the number and complexity of the probability distributions that need to be defined is small. The only requirement is to specify a function for each clique in the undirected setting. In the directed setting, each vertex needs a probability distribution, but this distribution depends only on the vertex's parents. Computing the full probability distribution $p(x_1, x_2, \ldots, x_m)$ for any $(x_1, x_2, \ldots, x_m) \in (\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_m)$ can be done efficiently using equations (3.4) and (3.5) for the undirected and directed cases, respectively. These concepts are described in full detail in [94].

Given the temporal nature of estimating traffic, a directed graph is the most appropriate for our purposes. This type of graphical model is also known as a *Dynamic Bayesian Network* (DBN). The review of filtering techniques and the expectation maximization algorithm will assume the use of a directed graph in sections 3.2.2 and 3.2.3.

3.2.2 Statistical Filtering

In the Bayesian approach to dynamic state estimation, one attempts to construct the probability density function of the state at time interval t based on all available measurements up to and including time interval t. This probability density function is known as *posterior* probability distribution function. The process of estimating the posterior probability distribution function of the state of the network at time interval t is called *filtering*. This filtering process can be used to compute the E step of the EM algorithm presented in section 3.2.3, where E stands for expectation and M stands for maximization. Such a filter consists of essentially two stages: *prediction* and *update*. The prediction uses the transition probabilities to predict the state probability distribution from one measurement to the next. The update operation uses the latest available measurements to modify the state probability distribution using Bayes theorem.

On small graphs, it is possible to do exact inference (compute the optimal Bayesian estimate) by using the *Junction Tree* algorithm [85]. For graphs that have the structure of a *Hidden Markov Model* (HMM), the probability of a state at time t given observations up to and including time t is computed using alpha recursion (also know as forward algorithm). For more details on inference in HMMs, refer to [80, 31]. This algorithm is often presented in the case of unique observation of the hidden state at each time interval. It can be generalized to this model where the number of observations is variable and unknown a priori. HMMs are an important tool built upon in this dissertation for both modeling and parameter estimation (chapters 5 and 6).

When one cannot use the inference techniques for HMMs (because the structure is more complicated) and when the junction tree algorithm is not applicable, an approximation method is needed to perform filtering. One commonly used method is particle filtering (also known as bootstrap filtering or condensation algorithm). For more information on particle filtering, see, for example [85, 22]. It is a technique for implementing a recursive Baysesian filter algorithm by Monte Carlo Simulations. The idea is to represent the required posterior density by a set of random samples with associated weights (importance weights) and to compute estimates based on these samples and weights. As the number of samples becomes very large, this Monte Carlo approximation approaches the exact optimal Bayesian estimate.

In chapter 6, a specific particle filter for traffic estimation is introduced. There are many issues and potential problems with using particle filters and those will be addressed in that chapter.

Mathematically, the evolution of the state of the system (e.g. the current congestion levels on the road network) is denoted $\{x_0, x_1, \ldots, x_n\}$, where x_t is the state of the system at time interval t. It is assumed that x_t is discrete and never observed directly, but rather a series of noisy observations $\{y_0, y_1, \ldots, y_n\}$ are observed where $y_t|x_t$ is distributed according to a known function $p_{y|x}(y|x_t)$. A particle filter can be used in the situation where the process to be estimated is modeled as a Markov process with a known transition probability function $p_{x_t,x_{t-1}}(x|x_{t-1})$ (i.e. conditioned on the current state, the next state of the process is independent of the history of the process). Section 3.2.3 describes how to use particle filtering when the function is not known with certainty (in the context of the EM algorithm).

The goal of the particle filtering algorithm is to estimate the probability distribution of x_T given the set of observations $\{y_0, y_1, \ldots, y_T\}$. The algorithm approximates this distribution using a set of weighted samples of the state (the samples are also called particles). The weights are computed using the observation distribution function, which determines how likely a sample state is. Determining the optimal number of samples to use as representative of the state is dependent upon the size of the state space and the dynamics of the process. Experimentation is generally needed to determine a good value for the number of samples to produce.

The particle filtering algorithm is described in algorithm 1, which assumes that the transition and observation probability functions and a total number of samples (denoted N) are given. The end of the algorithm includes a *resampling* stage to avoid the problem of degeneracy (when one sample has a weight of 1 and all others have weight 0). This algorithm is also known as *Sequential Importance Resampling* (SIR) [48]. The algorithm is presented here for reference and will be referred to later in chapter 6.

Algorithm 1 One iteration of a basic particle filtering algorithm [48].

```
1: The algorithm for time period $t$. The samples are kept from the previous time period $x_{t-1}^{(i)}, i = \{1, \ldots, N\}$.
2: Generate $N$ random samples of the state $x_t^{(i)}, i = \{1, \ldots, N\}$.
3: Generate intermediate weights using $\hat{w}_t^{(i)} = p(y_t|x_t^{(i)})$.
4: Renormalize the weights: $w_t^{(i)} = \frac{\hat{w}_t^{(i)}}{\sum_{j=1}^{N} \hat{w}_t^{(j)}}$.
5: Resample: choose $N$ random samples $\{\hat{x}_t^{(i)}\}_{i=1}^N$ from $\{x_t^{(i)}\}_{i=1}^N$ with replacement in proportion to the weights $\{w_t^{(i)}\}_{i=1}^N$.
6: Replace the sample set with these new samples, i.e., $\{x_t^{(i)}\}_{i=1}^N \leftarrow \{\hat{x}_t^{(i)}\}_{i=1}^N$, and set the weights to be equal: $w_t^{(i)} = \frac{1}{N}, i = 1, \ldots, N$.
```

3.2.3 Expectation Maximization

In order to use any statistical filtering technique described in section 3.2.2, it is necessary to know the value of the parameters of the probability distribution functions used in these techniques. In practical applications, these parameters are frequently unknown and need to be learned from the data. The goal of the *Expectation Maximization* (EM) algorithm is to learn the parameters of a graphical model using incomplete data. An example of incomplete data in the context of traffic is for the traffic *state* to be unobserved (i.e. how congested the road is) while some indirect, noisy measurements are observed (i.e. the travel time of a few vehicles). A graphical model is frequently used in this situation to capture the dynamics of the hidden state as it evolves over time with observation nodes used to represent the noisy measurements that will be collected. The goal of the EM algorithm is to use a set of noisy measurements observed over time along with the model of the process dynamics to estimate the parameters of the transition and observation distribution functions.

The EM algorithm addresses the following paradox: given the (unknown) parameters of the distribution functions, a statistical filtering technique can be used to estimate the probability distribution of the current state of the process; and given the (unknown) the current state of the process, the maximum likelihood estimator of the distribution parameters can be computed. At the outset, neither the true state nor the distribution parameters are known. The EM algorithm deals with this by iterating between these two steps, first fixing the values for distribution parameters and determining the probability distribution for the state (the E-step) and then fixing the state probabilities and determining the distribution parameters (M-step). The EM algorithm is described in detail in [85].

The most important part of the EM algorithm as it pertains to this dissertation is that the E-step can be computed via a particle filter (or any other statistical filter). This fact will be used when applying the EM algorithm to the traffic models presented in chapter 6.

3.3 Highway Traffic Estimation Techniques

Highways are crucial to the transportation network as they generally provide the fastest means of travel between two locations not located in the same city. The importance of the highway network has led public transit agencies to place their primary focus on estimating highway traffic. In terms of miles of roadway, highways are only a small percentage of the entire road network, and so it is also easier to cover the highway network with sensors. For these reasons, the highway network has been well-instrumented with fixed-location sensors throughout many major urban environments in the United States and Europe.

For highway networks covered by such an infrastructure, 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 very fine spatio-temporal scale [98, 29]. These highway traffic monitoring approaches heavily rely upon both the ubiquity of data and highway traffic flow models developed over the last half century [68, 39, 91].

Highway networks were also the first to be studied when GPS probe data first started becoming available. The *Mobile Century* experiment (a predecessor to *Mobile Millennium*) demonstrated the capability of estimating highway traffic conditions using only GPS probe data [52, 51, 98]. Several extensions to that original work have been developed in the past two years, leading to a thorough understanding of how to perform traffic estimation using a combination of fixed-location sensor data and GPS probe data [98, 35, 36]. The techniques used in those papers combine an underlying flow model of highway traffic with an *Ensemble Kalman Filtering* (EnFK) algorithm ([43]) for estimating real-time traffic conditions. These techniques rely on the assumption that highway traffic acts roughly as a continuous fluid. The fluid approximation does not work as well on arterials due to the number of impediments to the flow (such as traffic lights, stop signs, pedestrians, etc.). The important thing to note here is that the problem of using GPS probe data on highways (along with fixed-location sensor data) has been solved with a high degree of accuracy.

3.4 Previous Arterial Traffic Estimation Techniques

Existing research on arterial traffic estimation varies greatly in the models presented and the data sources assumed available for those models to work. This section categorizes all of these models as either *flow models* (built on traffic flow theory) or *statistical models* (using data-driven knowledge of traffic). To the best of our knowledge, there exists no research that uses a hybrid approach of flow models and statistical models, which is the basis of this dissertation.

Every approach to estimating arterial traffic conditions requires answering two fundamental questions:

1. What data is available or assumed available to the model?

2. What structure should the model have for processing the data?

Each of these questions is actually very broad and requires answering a series of subquestions in order to fully characterize the estimation technique. With respect to data, the following points must be addressed:

Sensing Infrastructure Does the model assume that fixed-infrastructure sensors have been placed at specific locations on the roadway? Does the model only use data from probe vehicles (i.e. GPS)? Can it handle both types of data?

Frequency Does the model expect new data at a regular repeating interval (i.e. every 30 seconds)? Can the model handle data streams (i.e. process each new observation when it is received)? For probe data, how often does the probe need to report its position? How much data is transmitted by the probe vehicle?

Coverage In the case of fixed sensors, where do they need to placed to satisfy the demands of the model? For GPS probe data, how much of the road network do the probes need to cover (and how often do they need to cover it)?

Latency How timely must the data be for the model to be accurate? How is the model affected when data arrives 1 minute late, 5 minutes late, half an hour late, etc.?

Coverage and frequency are directly related. One way of characterizing this relationship is to consider the *spatio-temporal* coverage of the data, which means how often and with what spatial resolution is data received. Chapter 2 provides specific details about each of the currently available sensing paradigms and how each type of sensing performs with respect to the list above.

With respect to model structure, the following points must be addressed:

Data preprocessing Does the data need to be changed from its raw form? (i.e. aggregating data within a time window, taking averages, medians, maximums, minimums, etc.)

Estimation frequency How frequently will the estimate be updated? (i.e. every minute, every 5 minutes, every hour, etc.)

Estimation quantities Which state types will the model estimate? Velocity, density, flow, travel time?

Spatial resolution of estimates Will the model produce an estimate for every link of the network? Will links be aggregated and estimates be produced at the aggregate level? Will the model produce sub-link level estimates?

Estimation types Will the model estimate historic traffic conditions (i.e. a typical Monday at 9am), estimate real-time traffic conditions (i.e. what is happening *right now*), or forecast future traffic conditions (i.e. what will traffic be like in half an hour)?

Previous research efforts have studied how to directly measure delays and travel times through the arterial network through vehicle re-identification (for example [76, 83, 77, 66]). With the direct measurement approach, it is necessary to place sensors everywhere in the network where traffic information is needed. See chapter 2 for an overview of the different sensing technologies available on arterial roads as well as their strengths, weaknesses, and costs. Given that no direct measurement sensors are available for the entire arterial road network (or even close to that coverage), the focus here is on reviewing previous research that infers conditions from incomplete measurements.

In the remainder of this section, previous arterial traffic estimation techniques will be reviewed in the context of how they address each of the points above (for both data and modeling).

3.4.1 Flow Models

The basic idea for all estimation techniques based on flow models is to use the traffic theory principles introduced in section 3.1 as the basis for estimating traffic quantities of interest (flow, density, velocity, travel time). The goal of such techniques is to determine the parameters of the model (fundamental diagram parameters, signal settings, etc.) as well as how to incorporate real-time data into the model. These types of estimation models attempt to relate various traffic variables and assume that measurements of at least one variable are available to estimate the others. For example, one could use flow data to estimate travel times, or one could use travel time measurements to estimate density.

Skabardonis and Geroliminis have written a series of papers on estimating travel times on arterial roads [46, 88, 90]. Their model shares many similarities with several other previous researchers, such as [45, 86, 102, 100, 87, 70, 101], but the focus here remains on the model of Skabardonis and Geroliminis, which is representative of the issues of these types of modeling approaches. Their model is built upon the traffic flow fundamentals described in section 3.1. The goal of the model is to estimate travel times along a single arterial link, as opposed to estimating travel times through any route in an arterial network. They assume that every traffic signal along the street of interest has the same total cycle length and that the signals are coordinated by a fixed offset from each other. They make the following assumptions about data availability:

Sensing Type Loop detectors are installed along the route and the signal system is capable of providing data about the green and red light times back to the traffic estimation system.

Frequency The loop detectors provide data once per cycle.

Coverage A loop detector is placed upstream of each signalized intersection.

Latency The model needs all current data to estimate the travel time, so the latency of the model is equal to the longest communication delay of all the loop detectors on the route (plus some small model processing time).

In terms of the model, the authors make the following structural decisions:

Data preprocessing No preprocessing is needed. The raw flow and occupancy measurements are used directly and are assumed correct (i.e. no inaccurate measurements enter the system).

Estimation frequency The model estimates the travel time of each arriving vehicle, so the estimation frequency is very high.

Estimation quantities The model estimates travel times on the entire stretch of road.

Spatial resolution of estimates The model only estimates a single travel time value for the road of interest.

Estimation types The model only provides estimates of the travel times that have just occurred. The model does not provide historical traffic conditions and does not perform prediction.

The Skabardonis and Geroliminis model provides a very sound basis for estimating travel times on arterial roads. They build upon the fundamentals of traffic flow theory to be able to incorporate data from one of the most common sensor types available (loop detectors). The primary drawback of the work is the fact that loop detectors and traffic signals are very rarely connected to a communication network that would allow for processing of the data in real-time as well as the fact that this dedicated sensing infrastructure does not have global coverage nor does it have the prospect of good coverage in the near future. The other drawback is that the model focuses on estimating one specific travel time along a road and does not provide a means for calculating arbitrary travel times through the arterial network unless every signal in the network has the same signal cycle length, which is not true in general.

3.4.2 Statistical Models

Many different statistical approaches to arterial traffic estimation have been proposed. There is much variety in terms of the goals of each techniques and the assumptions made to achieve those goals. In general, the techniques reviewed here can be categorized as attempting to apply a standard statistical technique to a particular traffic data type in order to estimate or predict other traffic variables.

Regression is a common statistical tool used frequently in many applications and can be considered one of the simplest forms of machine learning. Given the spatio-temporal nature of traffic conditions, a STARMA model [79] is a logical choice to use for arterial traffic estimation. STARMA is a specific type of regression model that can be used to predict the space-time evolution of a variable (such as link travel time). This type of model requires both space and time to be discrete quantities, so when applying this technique to arterial traffic, one must determine (theoretically or experimentally) the appropriate level of discretization. A common spatial discretization for the arterial network is to look at aggregate link quantities such as travel time (directly related to average link velocity), average link flow, or average link density. Time discretization is a trickier subject as there is no logical discrete value for how often conditions can change. Researchers often end up choosing values that seem logical, but without a lot of evidence for why the value was chosen. Several papers have presented various STARMA-based approaches to arterial traffic estimation [73, 63]. These papers have the following data and modeling characteristics:

Data

Sensing Type Fixed-infrastructure loop detectors.

Frequency The loop detectors provide data at a fixed interval.

Coverage Estimates will be made only at locations where sensors exist, so the coverage is directly equivalent to where the sensors are placed.

Latency The models need a complete set of data for a given time interval in order to compute the next set of predictions, so the latency of the system is equal to the last reporting sensor.

Modeling

Data preprocessing A single aggregate quantity is needed for each time interval. That means using either average flow or average density in the time interval.

Estimation frequency The model produces estimates at the same time interval that data is collected at.

Estimation quantities The model predicts the same quantities that are being collected as input into the model.

Spatial resolution of estimates One estimate is given per sensor location, so estimates are only available where the sensors are located.

Estimation types The model can do short-term prediction and potentially long-term prediction, although accuracy often decreases as a function of the prediction horizon. It is assumed that the data coming in real-time is a perfect description of the current state of the road, so no real-time estimation is needed. These models do not provide an estimate of historical traffic conditions.

Previous work has also studied the use of *neural network* [95] and *pattern-matching* [47] algorithms with GPS probe data as input [40]. In this work ([40]), the traffic estimation system makes the critical assumption that the average velocity driven by a few probe vehicles over a link is equal (or nearly equal) to the actual average link velocity for all vehicles. A primary argument of this dissertation is that taking an average of a small number of probe vehicles is insufficient for estimating arterial traffic conditions. After making this critical assumption, the authors then proceed to show how one can use both neural network and pattern-matching algorithms to predict future link velocities. The neural network approach uses two feedforward neural network models per link with one hidden layer. The pattern-matching approach categorizes link velocities into a small set of discrete categories and then performs prediction by examining the currently available data and seeing which historical pattern the current data is most similar to. The summary of the data and model characteristics is as follows:

Data

Sensing Type GPS probe data from private vehicles.

Frequency The GPS devices send data approximately every 12 minutes when driving on specified roadways.

Coverage A set of roadways is specified and data is collected only on this set. The article assumes that a very large percent of drivers are using a GPS device that will send data. The study was conducted in Italy, where approximately 2% of drivers are using this system. In the United States, no single device (and associated system) has a penetration rate even remotely close to this value.

Latency Given that vehicles report data only every 12 minutes, the travel time values can be 10-15 minutes behind. The model accounts for this by doing prediction, but with a decrease in accuracy due to lack of real-time information.

Modeling

Data preprocessing GPS data is processed into average link velocities per vehicle and then a single average link velocity is computed using an exponential weighting scheme for the individual vehicle values. If data is missing because no vehicles have driven there recently, past values are used. In the case of pattern matching, the velocities were put into discrete categories.

Estimation frequency Estimates are produced every 3 minutes.

Estimation quantities Average link velocities for every link in the studied network.

Spatial resolution of estimates Limited to the pre-specified network where data is collected.

Estimation types The model is capable of both prediction and of providing a historical estimate of traffic conditions. Real-time estimation is assumed perfect by the way they process the GPS probe data.

Belief propagation is one of the standard machine learning tools from the world of probabilistic graphical models (see section 3.2.1). The basic idea of the technique is to take observations on a subset of the nodes of a graph and propagate the information contained in those observations to infer the state of all nodes of the graph. In [44], the authors base their arterial traffic model on the Ising model from statistical physics. While the ideas in this type of model originated in the field of statistical physics, the concepts are part of the generalized theory of probabilistic graphical models. In the Ising model, the set of states for each node of the graph are binary. In [44], the authors assume the fundamental states of arterial traffic are undersaturated and congested. Their model then uses a standard belief propagation algorithm (based on the Ising model) to take "observations" (defined as a measurement between 0 and 1 indicating the level of congestion of a link) on a subset of the links of the road network to then infer the probability of every link of the network being congested.

The key limitation of this model is that it requires pre-processing all of the observations into a single value between 0 and 1 for each link where measurements were recorded. It is not clear that the pre-processing technique described in this article can account for the variability in the measurement coming from a single vehicle. Another limitation is that the algorithm does not always converge and it is unclear under what conditions this situation can arise. Overall, this model shows great potential for being able to estimate traffic conditions on many parts of the network when data is only ever available over a small subset of the network in real-time. The idea of using belief propagation is similar to some of the algorithms described in this dissertation (chapter 6). In summary, the data and model characteristics of this model are as follows:

<u>Data</u>

Sensing Type GPS probe data from private vehicles.

Frequency The GPS devices determine the travel time for the probe vehicle on each link of the network and transmit the link travel time upon completing the traversal of the link. Thus, the frequency of the observations depends on the number of probe vehicles and how often they traverse a link of the network.

Coverage The algorithm is independent of the exact coverage, but the authors state that having data on 10% of the road network is roughly the level that the algorithm needs to perform well. This is the key major benefit of the algorithm, as it does not require data on all links of the network at all times.

Latency The link travel time information sent by the vehicles is only sent after the traversal of the link, so it is necessarily latent by the amount of the travel time. In general, this effect is small and it is reasonable to neglect it, particularly for arterial links that tend to be relatively short (generally no more than several hundred meters).

Modeling

Data preprocessing The link travel times are converted to a single probability on each link where measurements were received. The authors do this by looking at the cumulative distribution function of all travel times ever received on a link and seeing where the received travel time falls. This is a key limitation of the model as it is well-documented that travel times in the undersaturated regime can appear long just because of the presence of a traffic signal. This issue will be examined in more detail in chapter 4.

Estimation frequency The model can be run as often as desired. The authors do not specifically state how often the traffic estimates are updated.

Estimation quantities Probability of each link of the network being congested. The authors also have a method for determining average travel times from this value (through the historic cumulative distribution function).

Spatial resolution of estimates One estimate per link of the network.

Estimation types The method of collecting data allows the authors to give a historic probability distribution of travel times for each link of the network (without the need of a specific historic model). The model is designed for real-time estimation and it is not clear from the article how one could do prediction with this model.

Similar to the belief propagation algorithm just presented, *Bayesian Networks* can be used to model arterial traffic, as in [78]. In this article, the authors assume that traffic data is discretized into a few categories and that these categories fully represent the traffic conditions. Given the discretization of the data, the message passing (similar to belief propagation) algorithm used is considered standard in the machine learning community. The algorithm works by maximizing the posterior probability of the current traffic state given the data, and the model is able to work with observations on only a subset of the links of the network by propagating the information in the observations to the other links of the network. The main limitation of this work is that the choices for discretization of traffic data are unrealistic and lead to the model just performing classification without interpretation in the context of arterial traffic. The data and model characteristics are as follows:

Data

Sensing Type Dual-loop detector data.

Frequency Data is received every 5 minutes.

Coverage In the study described in the article, there were 5 dual-loop detectors each placed on a single link and there were 6 links total along the route studies.

Latency The model needs the data from each of the loop detectors before computing the next estimate, so the total latency is equal to the last reporting sensor.

Modeling

Data preprocessing The loop detector readings are put into discrete categories by simple interval thresholds.

Estimation frequency One estimate per 5 minutes.

Estimation quantities Average link velocities.

Spatial resolution of estimates One estimate per link (for the 6 links in the study).

Estimation types No historical or prediction model is given. The focus is on a real-time model capable of assessing the current state of traffic.

Another article that assumes high-frequency data is [92]. This article does not present a traffic estimation model, but focuses on simply extracting link travel times from probe vehicles. The primary contribution of the article is that the probe vehicles may have location information based on WiFi (in addition to other vehicles using GPS). The actual estimation of travel times are then just computed by extracting the travel times deduced from the tracking of vehicles through the network. No attempt is made to account for the variability of link travel times, nor to estimate travel times on links where no data has been received. However, the contribution of the article to map-matching and travel time extraction from noisy position measurements is an important practical issue that the authors were able to overcome.

To our knowledge, only one article (outside of the research done by the *Mobile Millennium* team) proposes a model using sparse probe data that may cover many links in between measurements. In [50], the authors propose a *travel time decomposition* approach for determining link travel times when the observations cover several links. The authors do not address the issue of map-matching or path inference, which are two critical pre-processing steps needed for the algorithm to perform well. However, the method proposed in the article for decomposing travel times is a novel one worth mentioning. The authors define likelihood functions for determining how likely it is for a vehicle to get stopped at traffic signal along the path and therefore are able to associate the delay along the path to traffic signals in addition to attempting to identify delay due to congestion. In searching through the literature, this is the first known attempt to use machine learning techniques to determine most likely travel times through the network from sparse probe data. The conclusions of the article ultimately indicate that the likelihood functions proposed are insufficient when the sampling frequency less than once per 90 seconds. For sampling frequencies above ones per 30 seconds, no benefit is gained from the approach as travel times can almost be directly inferred. The maximum benefit seems to occur around the sampling frequency of once every 60 seconds, where travel times cannot be directly inferred. This article served as inspiration for looking at the issue of travel time decomposition, although the approach presented in this dissertation is soundly based on traffic theory principles instead of intuition as in [50].

Chapter 4

Probabilistic Formulation of Deterministic Queueing Models

Travel times through arterial networks are dependent upon a large number of factors. The primary sources of delay for drivers are the presence of traffic signals and stop signs, as well as the queues that form as a result of intersecting traffic flows. After a brief introduction to the notation and assumptions used (section 4.1), this chapter presents an idealized model of travel times through a signalized intersection and derives the *delay pattern* that characterizes this theoretical model (section 4.2). Building upon this idealized model of travel times, section 4.3 brings this model into a probabilistic context. The goal is to derive parametric travel time probability distributions for all of the possible states of a link, where the link parameters are the same as those presented in the overview of arterial traffic theory in section 3.1.

Developing these travel time probability distributions is critical to the development of the estimation algorithms to follow in chapter 5. The basic idea is to consider any individual probe measurement as a random sample from the travel time distribution through the links that the vehicle traveled. By understanding the shape of the distribution from which the travel time was generated, it is possible to infer the most likely traffic conditions through which the vehicle traveled. This can then be used to propagate information about the state of one link to its neighboring links, all of which is the subject of chapters 5 and 6.

4.1 Notation and Assumptions

Traffic engineers employ a variety of notations that are considered common in the transportation community. Due to the use of both typical traffic notation and the introduction of new notation for the probabilistic approach, a summary of all notation used is presented in section 4.1.1. Also relevant to present in this section are all of the assumptions that are made throughout the development of the models in this dissertation (section 4.1.2).

4.1.1 Notation

The list below summarizes the notation introduced earlier (in section 3.1) and to be used in the rest of this chapter. The parameters are specific for each network link i. The variable t is always used to denote time. Sometimes it is used to refer to a time period (in a discrete time domain) or a time instant (in a continuous time domain). The context will make it clear which use is being employed at any given point.

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	$\rho^i_c v^i_f/(\rho^i_{\max} - \rho^i_c)$, Backward shockwave speed of link i.
v^i_f	Free flow speed of link i.
p^i_f	Free flow pace (inverse of free flow speed) of link i. Note that $p^i_f = 1/v^i_f$.
L^i	Length of link i (not a model parameter, but an attribute of the road that is used frequently).

2. **Traffic signal parameters**
   The traffic signal parameters characterize the properties of the traffic signal at the end of a link i.

C^i	Duration of a light cycle on link i.
R^i	Duration of the red time on link i.

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 $t \in \{0 \ldots T\}$. The reference to the link or to the time interval may be omitted when the derivations are not link or time specific.

$\rho_a^{i,t}$	Arrival density on link i during time interval t.
$v_a^{i,t}$	$\rho_a^{i,t}v_f^i/(\rho_{\max}^i-\rho_a^{i,t})$, arrival shockwave speed on link i during time interval t (speed of growth of the queue due to additional vehicles arrival).
$l_{\max}^{i,t}$	$R^i w^i v_a^{i,t}/(w^i - v_a^{i,t})$, length of the triangular queue on link i during time interval t.
$\tau^{i,t}$	$l_{\max}^{i,t}(1/w^i + 1/v_f^i)$, duration of the clearing time on link i during time interval t, which is the amount of time for the queue to clear in the undersaturated regime (defined for the undersaturated regime only).
$l_r^{i,t}$	Length of the remaining queue when the light turns red (defined for the congested regime only).

This set of variables is sufficient to characterize the model and the time evolution of the state of traffic. In particular, through all of the relational formulas, note that only $l_r^{i,t}$ and one of $\rho_a^{i,t}$, $v_a^{i,t}$, $l_{\max}^{i,t}$, or $\tau^{i,t}$ are needed to specify the traffic state given that the traffic parameters are fixed. The location x on a link corresponds to the distance from the location to the downstream intersection. From these variables, the other traffic variables can be computed, including velocity v, flow q, and density ρ of vehicles at any x and time t.

The following functions are used for compactness. In general, each function often has a different form depending on whether the undersaturated or the congested regime is being considered. The subscript s is used to denote the regime-specific form of the functions, which can be undersaturated, u, or congested, c.

$d^{s,i}(t)$	The travel time through link i for a vehicle entering the link at time instant t (in the continuous time domain) for regime s.
$\delta^{s,i}(x)$	The delay function for a given location x along link i in regime s.
$g^{s,i}(y_{x_1,x_2})$	The density (in the probability sense) of the travel time y_{x_1,x_2} between locations x_1 and x_2 along link i for regime s.

4.1.2 Traffic Flow Modeling Assumptions

The following assumptions are made on the dynamics of traffic flow:

1. *Triangular fundamental diagram*: this is a standard assumption in transportation engineering.

2. *Stationarity of traffic*: during each estimation interval, the parameters of the light cycles (red and cycle time) and the arrival density ρ_a are constant. Moreover, it is assumed that there is not a consistent increase or decrease in the length of the queue, nor instability. With these assumptions, the traffic dynamics are periodic with period C (length of the light cycle). The work is mainly focused on deriving travel time distributions for cases in which measurements are sparse. Constant quantities (for a limited period of time) do not limit the derivations of the model since the interest here is in trends rather than fluctuations.

3. *First In First Out* (FIFO) model: overtaking on the road network is neglected. When traffic is congested, it is generally difficult or impossible to pass other vehicles. In undersaturated conditions, vehicles can choose their own free flow speed, but it is assumed that the free flow speeds are similar enough that the "no overtaking" assumption is a good approximation.

4. *Model for differences in driving behavior*: the free flow speed is not the same for all vehicles: it is modeled as a random variable with a mean $\bar{v}_f$ and variance σ^2—*e.g.*, a Gaussian or Gamma distribution.

4.2 Travel Time Patterns Through Signalized Intersections

In this section, analytical patterns of intersection delays are developed for the undersaturated and congested regimes. The first regime occurs when queues can be cleared completely during the green phase of a cycle, while the second regime occurs when queues cannot be cleared within one cycle and the remaining queues will have to wait for extra cycles (i.e. more delay) to be cleared. In specific situations (e.g. heavy congestion), queues may spillover to upstream intersections and cause further delays. This third regime is omitted in this analysis and is subject to ongoing research as it requires a network model to determine the effect of neighboring links on the distribution of the link of interest.

Using the assumptions from section 4.1.2 and following the standard traffic theory model of vehicles through signalized intersections (presented in section 3.1), the travel time function $d(t)$ is derived. This function represents the amount of time needed for a vehicle entering the link at time t to pass through the intersection. For the remainder of this section, a single link is considered (so the index i is dropped from the notation) and it is assumed that the link parameters are fixed during the time period of the analysis (so the time interval index t as presented in the notation section 4.1.2 is dropped). In this section, t refers to a specific time instant in the continuous time domain.

The minimum travel time on the link is equal to the time it takes to traverse the link at the free flow speed, v_f, given the assumption that all vehicles travel at the free flow speed when not stopped. The minimum travel time is called the *free flow travel time* and is equal to $\frac{L}{v_f}$. If the light is red or there is an existing queue when the vehicle approaches the downstream intersection, the vehicle will join the end of the queue and thus be delayed. Otherwise, if a vehicle reaches the intersection when the light is green and there is no queue, the vehicle will pass through the intersection with no delay (thus experiencing the free flow travel time). More importantly, by analyzing the geometry of the triangles in the space-time diagram (figure 3.2), one can easily observe that if a vehicle enters the link at a time that would make it get to the intersection just after the start of the red time (assume no interruption had existed), delay for this vehicle will be the maximum for the specific cycle.

After that, delays will be reduced linearly until no delay is reached. The slope, b, of the travel time function can be calculated analytically as (see [24] for the derivation, which can also be seen graphically in figure 3.2):

$$b = -\frac{v_f(w - v_a)}{w(v_f + v_a)} = -1 + \frac{\rho_a}{\rho_c}. \tag{4.1}$$

Here w is the wave speed, v_f is the free flow speed, v_a is the wave speed when a vehicle joins the queue, $\rho_{\max}$ is the jam density, and ρ_a is the arrival density, which is assumed to be constant within a cycle. The three parameters $v_f, w, \rho_{\max}$ are specific to actual arterial locations, which also determine the fundamental diagram of the link. Using equation (4.1), the travel time function for the undersaturated regime is computed as follows:

$$d^u(t) = \frac{L}{v_f} + \max\left\{0, R + b(t + \frac{L}{v_f})\right\}, \text{ for } -\frac{L}{v_f} \leq t < C - \frac{L}{v_f}, \tag{4.2}$$

where $t = 0$ is defined to be the time at which the light turns red. The function is periodic with a period equal to the signal cycle time, so $d^u(t) = d^u(t + kC), k \in \mathbb{Z}$.

Note that the above analysis and equation (4.2) only works for the undersaturated regime when the minimum delay reaches 0 in each cycle. In the congested regime, the remaining queue, l_r, must wait for additional cycles to be cleared. In this situation, delay will still be reduced linearly from the maximum value after the start of the red time (when the vehicle arrives at the intersection). However, the delay will never reach zero. Instead, it will have a sudden increase from a nonzero delay to another maximum delay, indicating the vehicle will have to wait for an extra cycle to be cleared. The slope of the curves can all be computed analytically by looking at the geometry of the queue forming and discharging triangles (see figure 3.2). Using the assumption of stationarity from section 4.1.2, the slope of the travel time function is still the same as the undersaturated regime (computed in equation (4.1)). The difference in the congested regime is simply to calculate the delay due to the presence of the remaining queue. First, the maximum number of stops a vehicle will make before exiting the link is calculated as

$$n = \left\lceil \frac{l_r}{l_{\max}} \right\rceil. \tag{4.3}$$

The minimum travel time is then

$$tt_{\min} = \frac{L}{v_f} + \left((n-1) + \frac{l_r - (n-1)l_{\max}}{l_{\max}}\right) R. \tag{4.4}$$

This leads to the expression for the travel time function for the congested regime:

$$d^c(t) = tt_{\min} + \max\left\{0, R + b(t + tt_{\min})\right\}, \text{ for } -tt_{\min} \leq t < C - tt_{\min}, \tag{4.5}$$

where $t = 0$ is again the time when the light turns red (for some cycle). Just as in the undersaturated case, this function is periodic with a period equal to the cycle time.

Figure 4.1 depicts the travel time function in the undersaturated and congested regimes for a given set of link parameters. Note that both the undersaturated and congested functions are piecewise linear. The two functions are so similar that it is straightforward to calculate a single travel time function by setting $tt_{\min} = \frac{L}{v_f}$ when $l_r = 0$ (the undersaturated regime) and then equation (4.5) represents both the undersaturated and congested regimes. A complete description of this model with experimental results (using microsimulation data) can be found in [24].

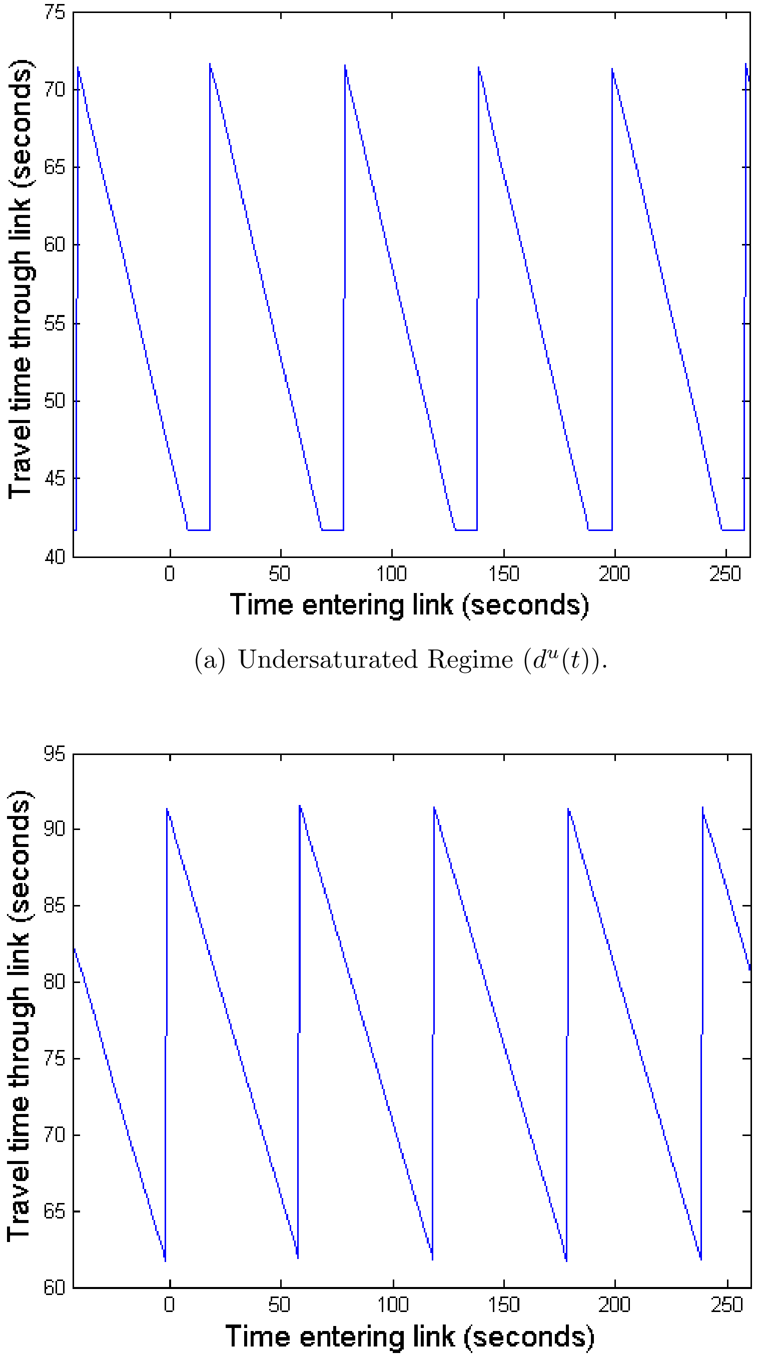
4.3 Travel Time Probability Distribution

The previous section presented the theoretical travel time function for vehicles traveling through a signalized link of the network. However, without measuring travel times from a high proportion of the vehicles traveling through the link, it is impossible to reconstruct that function directly. The penetration rate of probe vehicles will not be high enough to do this reconstruction until nearly all vehicles are being tracked, so another method needs to be developed to infer traffic conditions from more sparsely-sampled data. This section uses the concepts of the previous section, but from the perspective of sampling vehicles at random throughout the network. The goal is to develop probability distributions for the travel time through a link, which are functions of the link parameters.

The travel time experienced by vehicles traveling on arterial networks is conditioned on two factors. First, the traffic conditions, given by the parameters of the network, dictate the state of traffic experienced by the vehicle. Second, the time (relative to the beginning of a light cycle) at which the vehicle arrives at the link. The arrival time determines how much delay will be experienced due to the presence of a traffic signal and the presence of other vehicles (traffic conditions). Under similar traffic conditions, drivers experience different travel times depending on their arrival time. Using the assumption that the arrival density (and thus the arrival rate) is constant (see section 4.1.2), arrival times are uniformly distributed on the duration of the light cycle. This allows for the derivation of travel time probability distributions which depend on the characteristics of the traffic light and the traffic conditions as defined in section 4.1.

Using the modeling assumptions defined in section 4.1.2, one can derive travel time distributions on arterial links. An arterial link is defined as the stretch of road between two signalized intersections. Probe vehicles may send travel time information between arbitrary start and end locations on a link. The travel time from a location x_1 to a location x_2 on a link (with x_1 upstream of x_2) is called a partial link travel time from x_1 to x_2 and denoted y_{x_1,x_2}. The free flow travel time between x_1 and x_2 is defined as the travel time experienced when traveling at the free flow pace p_f. The difference between the travel time and the free flow travel time is referred to as *delay.*

The derivation of the link travel times in the undersaturated and congested regimes are



(a) Undersaturated Regime ($d^u(t)$).

(b) Congested Regime ($d^c(t)$).

Figure 4.1: Theoretical link travel time function for the undersaturated (a) and congested regimes (b). The link length was 500 meters and the signal parameters were a cycle time of 60 seconds and a red time of 30 seconds. The free flow travel time was 42 seconds and the length of the remaining queue for the congested case was 100 meters.

derived in [55]. The results of those derivations are presented here since they are used later in the dissertation. The probability distribution of travel times, $g_u(y)$, on an arterial link in the undersaturated regime is

$$g_u(y) = \begin{cases} 0 & \text{if} \quad y \leq 0 \\ \frac{1-\eta}{L}\varphi(\frac{y}{L}) + \frac{\eta}{R}(\Phi(\frac{y}{L})) & \text{if} \quad y \in [0, R] \\ \frac{1-\eta}{L}\varphi(\frac{y}{L}) + \frac{\eta}{R}(\Phi(\frac{y}{L}) - \Phi(\frac{y-R}{L})) & \text{if} \quad y \geq R \end{cases},$$

where $\Phi(x) = \int_{-\infty}^{x} \varphi(x)\, dx$ is the cumulative distribution function of φ, which is the distribution function for the free flow pace. The intuition behind this result is that for vehicles that do stop in the queue, the delay is uniform from 0 to the duration of the red cycle (depending on what time within the cycle the vehicle arrived). Some percentage of vehicles will not have to stop and these vehicles experience the free flow travel time. The assumption that drivers have some distribution of free flow pace results is used in conjunction with the law of total probability to derive the complete continuous distribution of travel times.

The probability distribution of travel times, $g_c(y)$, on an arterial link in the congested regime is

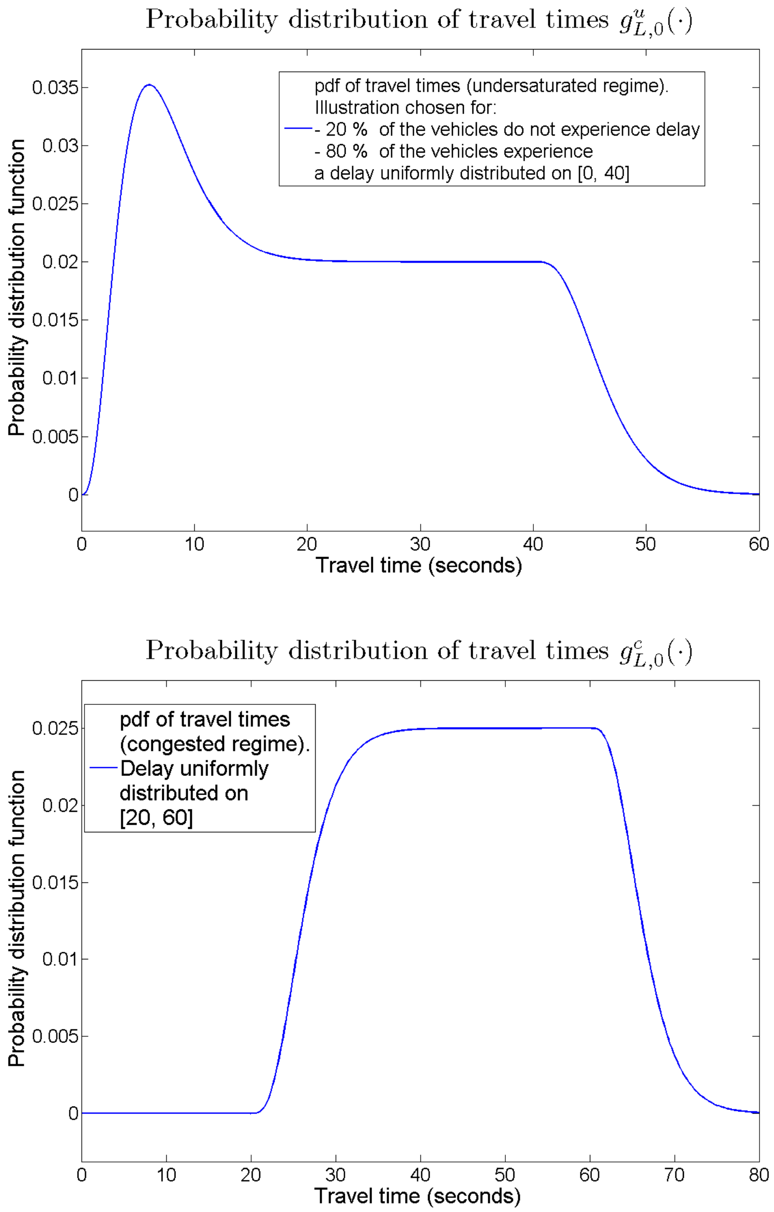
$$g_c(y) = \begin{cases} 0 & \text{if} \quad y \leq \delta_{\min} \\ \frac{1}{\delta_{\max}-\delta_{\min}}(\Phi(\frac{y-\delta_{\min}}{L})) & \text{if} \quad y \in [\delta_{\min}, \delta_{\max}] \\ \frac{1}{\delta_{\max}-\delta_{\min}}(\Phi(\frac{y-\delta_{\min}}{L}) - \Phi(\frac{y-\delta_{\max}}{L})) & \text{if} \quad y \geq \delta_{\max} \end{cases},$$

where $\delta_{\min}$ and $\delta_{\max}$ are the minimum and maximum delay given the queue length for the link. Note that $\delta_{\max} - \delta_{\min} = R$, by definition of the congested regime. The intuition behind this result is that there is some minimum delay that vehicles will experience since the definition of the congested regime is that every vehicle stops at least once in the queue. The delay is distributed uniformly on a given range and then the complete travel time distribution is again computed using the law of total probability. See [55] for complete details on how to derive these probability distributions and all related expressions. Figure 4.2 illustrates the general shape of these distributions for both the undersaturated and congested regimes.

Travel time distribution properties

The functions g_u and g_c are *quasi-convex* as shown in [55]. This is a useful property that will be exploited for one component of an estimation algorithm presented in chapter 6. However, a more important observation is that the log-likelihood function defined for g_u and g_c are in general *non-convex*. The log-likelihood function is defined as

$$ll(\mathcal{P}) = \sum_{j=1}^{n} \ln(g_s(y_{x_{j,1},x_{j,2}})), \tag{4.6}$$



Probability distribution of travel times $g^u_{L,0}(\cdot)$

Probability distribution of travel times $g^c_{L,0}(\cdot)$

Figure 4.2: Probability distributions of link travel times. **Top:** Undersaturated regime, $g_u(y)$. 20% of the vehicles do not experience delay, the other vehicles experience a delay uniformly distributed on [0,40], simulating a traffic light of red time 40 seconds. **Bottom:** Congested regime, $g_c(y)$. The vehicles experience a delay uniformly distributed on [20,60], simulating a traffic light of red time 40 seconds, where the vehicles where the vehicles stop once in the triangular queue and half of them stop a second time in the remaining queue. Both travel time distributions represent a link of length 100 meters. The mean free flow pace is 1/15 m/s and the standard deviation is 0.025 m/s. The red curves represent the conditional travel time distribution. The blue curves represent the travel time distribution when the differences in driving behavior are taken into account.

where $s \in \{u, c\}$, $\mathcal{P}$ is the set of parameters that define probability distribution function g_s, n is the number of observations, and $y_{x_{j,1},x_{j,2}}$ represents partial travel time observation j. It is easy to construct examples to show that $ll(\mathcal{P})$ is non-convex, which means that when trying to maximize this function, general non-linear, global optimization techniques must be used.

Chapter 5

Traffic Models for Estimation and Prediction

This chapter presents models for estimating arterial traffic conditions using GPS probe data as the only source of data. Two different data collection methodologies are considered as input into three different estimation models. The model presented in section 5.1 uses a VTL-based system as the only source of input data (see section 2.1.8 for a description of the VTL). This model is based on regression techniques considered standard in the statistics community and presents two variants, one with a discrete representation of the state of traffic and one with a continuous representation. The models presented in sections 5.2 and 5.3 use sparsely-sampled GPS probe data from fleets (see section 2.1.9). The sparsely-sampled GPS data format is more general than travel times from VTLs (since the two VTLs can just be considered to be the start and end points of each of the sparsely sampled vehicle trajectories), so data from VTLs can also be easily incorporated into the models presented in sections 5.2 and 5.3. The first of these two models focuses on how to allocate travel time data to the different links traveled between GPS observations. The last model is a graphical model (common in the machine learning community, see section 3.2), which is a much more generic and flexible framework than the other models.

5.1 Regression Models

This section introduces the use of regression models for estimating *Level of Service* (LoS) indicators which are the aggregate travel times and congestion states for an arterial road network. After stating the general assumptions of the regression models (section 5.1.1), the general problem of sensing on a graph (section 5.1.2) is presented followed by the formal definitions of LoS indicators (section 5.1.3). The problem description of estimating the LoS indicators based on STARMA and logistic regression is presented in section 5.1.4. The algorithms to solve these models are presented later in chapter 6.

5.1.1 Assumptions

There are a few key general assumptions that enable the regression models to be computationally tractable. First, it is assumed that there is a VTL-based infrastructure collecting travel time data between all adjacent pairs of VTLs on the road network. The VTL pairs are the fundamental building block for these models. All data and model estimates are assumed to be in the form of travel times for each VTL pair. Additionally, the models estimate the average travel time for a discrete time interval and do not estimate the probability distribution of travel times. The second general assumption is that there exist a discrete number of congestion states for each VTL pair and that this number of states is the same for all VTL pairs. Other technical assumptions are made in the context of deriving the models and will be presented when needed.

5.1.2 Graph Model of the Road Network

Consider an arterial network with a total of N pairs of VTLs deployed. Each pair has a unique identification number $i \in \{1, \ldots, N\}$. The set of all VTL pairs is denoted by $\mathcal{V} = \{1, \ldots, N\}$. Each VTL pair has a segment of road in between with a possibility of one or more road features such as an intersection (with or without traffic lights), pedestrian walkways, stop/slow signs etc. The characteristics of these road features can be static (such as presence of a stop sign) or dynamic (such as phase of a signalized intersection) with respect to time. The travel time experienced by a vehicle traveling through a VTL pair depends on the characteristics of the road features as well as the demand-capacity restrictions imposed by the dynamics of traffic flow.

The upstream (resp. downstream) VTL for the pair i is the VTL at which the traffic enters (resp. leaves) the corresponding stretch of road. For pair i, let the upstream and downstream VTLs be denoted by i_u and i_d, respectively. The VTL sensor network can be represented as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of all VTL pair as defined earlier and $\mathcal{E}$ is the set of all edges. Two VTL pairs i and j form an edge directed from pair i to pair j, denoted e_{ij}, if i_d and j_u correspond to same VTL. Then i (resp. j) is called the upstream (resp. downstream) node of edge e_{ij}.

Define the set of first order neighbors for VTL pair j as

$$\mathcal{N}^1(j) = \{j\} \cup \{i \in \mathcal{V} : e_{ij} \in \mathcal{E}\} \cup \{\kappa \in \mathcal{V} : e_{j\kappa} \in \mathcal{E}\}$$

which is simply the set of all the upstream and downstream VTL pairs for the pair j (in which pair j itself is included).

The above definition can be extended to define n^{th} $(n \geq 1)$ order neighbors as:

$$\begin{cases} \mathcal{N}^0(j) & = & \{j\} \\ \mathcal{N}^n(j) & = & \mathcal{N}^{n-1}(j) \cup \left(\bigcup_{l \in \mathcal{N}^{n-1}(j)} \{i \in \mathcal{V} : e_{il} \in \mathcal{E}\} \cup \{\kappa \in \mathcal{V} : e_{l\kappa} \in \mathcal{E}\} \right) \end{cases} \tag{5.1}$$

5.1.3 Traffic Level of Service Indicators

It is assumed that for any VTL pair $i \in \mathcal{V}$, the travel time data is available at times $0 \leq t_1 \leq t_2 \leq \ldots$. As an alternative representation to travel time data, the pace, can also be used (travel time divided by the length of road for the VTL pair). The data obtained at time t_1 for VTL pair i is denoted $X_{t_1,i}$ (i.e. the travel time or pace of a vehicle traversing VTL pair i starting at time t_1).

Since the data obtained is event-based, it cannot be directly used for training statistical models that needs regular sampling rates (i.e. one quantity per discrete time step). To address this, the travel time data is aggregated in t second windows to obtain a time series of observations at times $k = 0,\, t,\, 2t, \ldots$. Here t is the aggregation interval. Henceforth, k is used to denote the time interval $[(k-1)t, kt)$. The set of available observations during the time period k for any VTL pair i is denoted as $A_{k,i}$, that is,

$$A_{k,i} = \{X_{t_m,i} \,|\, (k-1)t \leq t_m < kt\}.$$

Define the spatial aggregation function for VTL pair i, $h_i(\cdot) : A_{k,i} \mapsto [0, \infty)$, as the function that aggregates the set of observations $A_{k,i}$ in to an *aggregate representative quantity*, denoted $Z_{k,i}$. In the remainder of this section, $Z_{k,i}$ is an aggregated travel time (seconds). Thus, the aggregate travel time for VTL i during interval k is

$$Z_{k,i} = h_i(\{X_{t_m,i} \,|\, (k-1)t \leq t_m < kt\}).$$

The *mode* of a VTL pair is defined as the categorical variable indicative of the extent of delay experienced in navigating through the VTL pair. For example, a binary mode classification can be *uncongested or congested.* Thus, the mode of a VTL pair can also interpreted as a *congestion state.* Let the mode of VTL pair i during time interval k be denoted as $Q_{k,i}$. In order to convert the total number of observations available for VTL pair i during time interval k into a congestion state, a *congestion indicator function* is defined as $g_i(\cdot) : A_{k,i} \mapsto \{1, \ldots, M\}$ where M is the number of congestion states consider. M is a meta-parameter of the model that is chosen based on a preliminary analysis of the data for the site under consideration. Several values of M can be chosen to see which fits best. With these definitions, $Q_{k,i}$ is determined as

$$Q_{k,i} = g_i(\{X_{t_m,i}|(k-1)t \leq t_m < kt\}).$$

From a statistical modeling perspective, both the aggregate speed or travel time, $Z_{k,i}$, and the congestion state, $Q_{k,i}$, for $i \in \mathcal{V}$ and $k \in \{0, 1, \ldots\}$ can be considered as random processes generated by space-time varying traffic flow phenomena on the arterial network. Both $Q_{k,i}$ and $Z_{k,i}$ can be regarded as LoS indicators.

5.1.4 Estimating Level of Service Indicators

If data was available from all the vehicles for all the VTL pairs over the entire time horizon of interest, the entire probability distribution of $Z_{k,i}$ and $Q_{k,i}$ could be computed directly. However, the challenge of arterial traffic state estimation and forecast is that the typical penetration rates are very low. The focus here is to develop reliable estimation and forecasting methods for such situations.

Typically only a small percentage of the total number of vehicles provide data to the system. Thus, the choice of the aggregation function $h_i(\cdot)$ (resp. the congestion indicator function $g_i(\cdot)$) becomes critical to obtain reliable estimates of $Z_{k,i}$ (resp. $Q_{k,i}$). For a given choice of $h_i(\cdot)$, the best estimate of the aggregate travel time or speed for VTL pair i during interval k is given by the conditional expectation of $Z_{k,i}$ given the aggregate travel times up-to (and excluding) the current time interval:

$$\hat{Z}_{k,i} = \mathbb{E}[h_i(A_{k,i})|h_j(A_{j,v}), j < k, v \in \mathcal{V}] = \mathbb{E}_{h_i}[Z_{k,i}|Z_{j,v}, j < k, v \in \mathcal{V}],$$

where $\mathbb{E}_{h_i}[\cdot]$ is used to indicate the dependence of the expectation on the aggregation function h_i. It is assumed that $Z_{k,i}$ is conditionally independent of all other data conditioned on the data from the past r time intervals for VTL pairs in the set $\mathcal{N}^s(i)$. Under this assumption, $\hat{Z}_{k,i}$ can be rewritten as

$$\hat{Z}_{k,i} \approx \mathbb{E}_{h_i}[Z_{k,i}|Z_{j,v}, k-r \leq j < k, v \in \mathcal{N}^s(i)] \tag{5.2}$$

Thus, $\hat{Z}_{k,i}$ only depends on data with r *temporal dependencies* in the past and s *spatial dependencies* from the neighbors. Similarly, for given choices of the aggregation function $h_i(\cdot)$ and the congestion indicator function $g_i(\cdot)$, the conditional expectation of $Q_{k,i}$ given all the aggregate travel times up-to (and excluding) the current time interval is

$$\begin{aligned} \hat{Q}_{k,i} &= \mathbb{E}[g_i(A_{k,i})|h_j(A_{j,v}), j < k, v \in \mathcal{V}] \\ &\approx \mathbb{E}_{h_i,g_i}[Q_{k,i}|Z_{j,v}, k-r \leq j < k, v \in \mathcal{N}^s(i)], \end{aligned} \tag{5.3}$$

In the statistics terminology, the quantities $Z_{k,i}$ and $Q_{k,i}$ in (5.2) and (5.3) are known as the *response variables*; the conditioned variables $Z_{j,v}$ and $Q_{j,v}$ are called the *dependent variables* or *covariates*.

This section compares two estimators. The first estimator is based on expressing (5.2) as a *linear regression problem.* For a temporal and spatial dependence of orders r and s respectively, a linear dependence of response $Z_{k,i}$ on the covariates $Z_{j,v}$ is assumed:

$$\hat{Z}_{k,i} = \beta_i^0 + \sum_{v \in \mathcal{N}^s(i)} \left(\sum_{j=k-r}^{k-1} \beta_i^{j,v} Z_{j,v} \right). \tag{5.4}$$

In order to make the notation concise, let $\mathbf{Z}_{k,i}^{r,s}$ be the $r \times \mathcal{N}^s(i)$ vector of covariates or dependent variables obtained by stacking the aggregate travel times $Z_{j,v}$ for $k - r \leq j < k$ and $v \in \mathcal{N}^s(i)$, β_i be the corresponding $r \times \mathcal{N}^s(i) + 1$ vector of parameters to be estimated. Then the equation (5.4) can be rewritten as

$$\hat{Z}_{k,i} = \beta_i^\top \mathbf{Z}_{k,i}^{r,s}.$$

As described later in section 6.1.2, instead of a simple regression model (5.4), a STARMA model is used, which is an extended version of the simple regression model.

The second estimator is based on expressing (5.3) as a logistic regression problem which assumes a linear dependence of the *logit* or the log-odds ratio of conditional expectation $\hat{Q}_{k,i}$ on the response variables. That is, for a temporal and spatial dependence of orders r and s respectively,

$$\log\left(\frac{\hat{Q}_{k,i}}{1 - \hat{Q}_{k,i}}\right) = \beta_i^\top \mathbf{Z}_{k,i}^{r,s}.$$

This equation can be expressed as

$$\hat{Q}_{k,i} = f_{\beta_i}(\mathbf{Z}_{k,i}^{r,s}) := \frac{1}{1 + \exp\left(-\beta_i^\top \mathbf{Z}_{k,i}^{r,s}\right)}, \tag{5.5}$$

where the subscript β_i in $f_{\beta_i}(\cdot)$ encodes the dependence on the β_i.

The implementation of a logistic regression estimator is detailed in section 6.1.1 and the STARMA-based estimator in section 6.1.2. However, two important points need to be mentioned here. First, the above formulation can be modified to include the case of multiple steps forecast. For example, an $m-$step forecast at time k for VTL pair i can be written as

$$\hat{Z}_{k+m,i} = \mathbb{E}_{h_i}[Z_{k+m,i} | Z_{j,v}, j \leq k, v \in \mathcal{V}], \tag{5.6}$$

where data up to time k is used to predict traffic at time $k + m$.

Second, note that for some VTL pairs and time intervals, there may not be any available data, that is, $A_{j,v} = \emptyset$ for some $j \in \{k - r, \ldots, k\}$ and $v \in \mathcal{N}^s(i)$. In this case, one has to employ a technique of *estimation with missing data*. The forecast problem is addressed for the STARMA model but the issue of missing data is something that is handled by the more robust graphical model approach presented in section 5.3.

5.2 Historic modeling using Bayesian inference for real-time estimation

In this section, an *independent link travel time* model of arterial traffic is proposed. This model learns the historic traffic patterns of the network and uses these historic patterns (section 5.2.1) as prior information to be used in a Bayesian real-time estimation model (section 5.2.2). The assumptions of the model are as follows:

1. The travel time distribution for each link of the network is independent from all other links of the network. The set of links of the network is denoted $\mathcal{L}$.

2. Any given moment in time belongs to exactly one *historic time period*, during which traffic conditions are assumed constant. The set of historic time periods are denoted $\mathcal{T}$.

3. All of the travel time observations from a specific link l are independent and identically distributed with a given time period, $t \in \mathcal{T}$.

4. Sparse probe measurements are the only data available to the model (see section 2.1.9).

5.2.1 Historic Model of Traffic

The historic model of arterial traffic estimates the average travel time as well as the standard deviation of travel time within a given historic time period (i.e. Mondays 4pm-5pm, 5pm-6pm, etc.) for each link of the network. The model parameters for link l and time period t are denoted $Q_{l,t}$. The set of all link parameters for a given time period is denoted $\mathbf{Q}_t$. The link travel time probability density function for link l during time period t is denoted $g_{Q_{l,t}}(y)$ for a given travel time y.

The observations available to the estimation model are assumed to be in the form of *path observations*. The pth path observation, y_p, is defined as a set of consecutive links traveled, L_p, through the network along with the fraction of the first and last link traversed as well as the total travel time associated with the entire path. The fraction of link l traversed for the pth observation on link l is denoted $w_{p,l}$. The set of path observations for time period t are denoted $\mathbf{P}_t$. For the pth path observation, the path travel time distribution is denoted $G_{P_{L_p,t}}(y_p)$, which is the convolution of the link travel time distributions that make up the path, where $P_{L_p,t}$ denotes the parameters of the L_p links along the pth path observation.

The goal is to determine the values of $Q_{l,t}$ for each link and time period that are most consistent with the probe data received. This is achieved by maximizing the likelihood of the data given the parameters, which is written

$$\arg\max_{\mathbf{Q}_t} \sum_{p \in \mathbf{P}_t} \ln(G_{Q_{L_p,t}}(y_p)), \tag{5.7}$$

where N_t is the number of observations available for time period t. This optimization problem is challenging due to the high number of variables (number of links times number of parameters per link travel time distribution) and it is not directly decomposable. In section 6.2.1, an intuitive decomposition scheme is presented for finding near-optimal solutions to this optimization problem. Given the assumption that each time period is independent, equation 5.7 can be solved for each value of $t \in \mathcal{T}$ separately.

5.2.2 Real-Time Estimation

The parameters learned by the historic model (section 5.2.1) are used as priors to estimate current traffic conditions via a *Bayesian update* (see [84] for more on Bayesian statistics). The process of doing a Bayesian update maximizes the likelihood of the current data given the prior distribution, with the assumption that the travel times are normally distributed.

In the general form, given a prior $p(Q_{l,t})$ for some set of link parameters (from the historic model) and a set of measured travel times $y_{l,t}$ for that same link, the posterior function is written

$$p(Q_{l,t}|y_{l,t}) = \frac{p(y_{l,t}|Q_{l,t})p(Q_{l,t})}{\int p(y_{l,t}|Q_{l,t})p(Q_{l,t})dQ_{l,t}}. \tag{5.8}$$

The goal is then to choose $Q_{l,t}$ that maximizes $p(Q_{l,t}|y_{l,t})$, which is dependent upon the specific distributions chosen for the model. The details of how to compute the Bayesian update are left for chapter 6. The specifics regarding the real-time estimation step are found in section 6.2.2.

5.3 Probabilistic Graphical Model

The modeling approaches presented in the first two sections of this chapter made a number of strong assumptions. The goal of the model presented in this section removes some of these assumptions and presents a modeling framework that is flexible and extensible. The key features that this model possesses are:

- Each link has a discrete traffic state that cannot be directly observed.
- Traffic states of nearby links are correlated and evolve over time in a Markov manner (i.e. the future is independent of the past given the present).
- Expectation maximization is an appropriate tool for learning the transition and observation model parameters.

While some assumptions are still made for computational tractability, these assumptions can gradually be removed as more advances are made on this topic.

5.3.1 Assumptions

The graphical model presented in this section makes the following assumptions:

1. *Discrete congestion states*: for each day d and each time interval t, the traffic conditions on link l are represented by a *discrete* value, $s^l_{d,t}$, which indicates the level of congestion. There are S discrete levels of congestion.

2. *Conditional independence of link travel times*: conditioned on the state $s^l_{d,t}$ of a link l, the travel time distribution of that link is independent from all other traffic variables.

3. *Conditional independence of state transitions*: conditioned on the states of the spatial neighbors of link l of order n (denoted $\mathbf{N}^l_n$) at time t, the state of link l at time $t+1$ is independent from all other current link states, all past link states and all past travel time observations.

$\mathbf{N}^l_n$ denotes the spatial neighbors of link l of order n, where first order neighbors ($\mathbf{N}^l_1$) are the links sharing an intersection with link l (including link l). The higher order spatial neighbors are defined by the following recursive formula:

$$\mathbf{N}^l_{n+1} = \bigcup_{j \in \mathbf{N}^l_n} \mathbf{N}^j_1 \tag{5.9}$$

Assumption 2 implies that link travel times are not correlated across links, which is an assumption made for computational tractability. Assumption 3 implies that each link is correlated with some (small) subset of neighboring links, but independent of the rest of the network. Neither of these assumptions must hold all of the time in a real traffic network, but some approximation is necessary for computational tractability of the model. The full effect of these assumptions has not been studied to date, but is necessary for full validation of this model.

5.3.2 Graphical Model

Arterial traffic conditions vary over space and time. Given the assumptions in section 5.3.1, the spatio-temporal conditional dependencies of arterial traffic are modeled using a probabilistic graphical model known as a *Coupled Hidden Markov Model* (CHMM) [30]. A *Hidden Markov Model* (HMM) is a statistical model in which the system being modeled is assumed to be a Markov process with unobserved states. CHMMs model systems of multiple interacting processes. In the present case, the multiple processes evolving over time are the discrete *states* (assumption 1) of each link in the arterial network. The discrete state of each link l and time period t is denoted $Z_{l,t}$. Since the state of each link for all times is not directly observed, these processes are considered *hidden*. The travel time distribution on each link is conditioned on its hidden state (assumption 2) from which come sparse observations from probe vehicles traveling through the arterial network. The observations for link l and time period t are denoted by the set $\mathbf{y}_{l,t}$. Assumption 3 gives the *coupled* structure to the HMM by specifying local dependencies between adjacent links of the road network. Figure 5.1 illustrates our 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 state is a discrete quantity defined based on the application (e.g. the possible states could be undersaturated/congested or the number of vehicles in the queue). The forward arrows

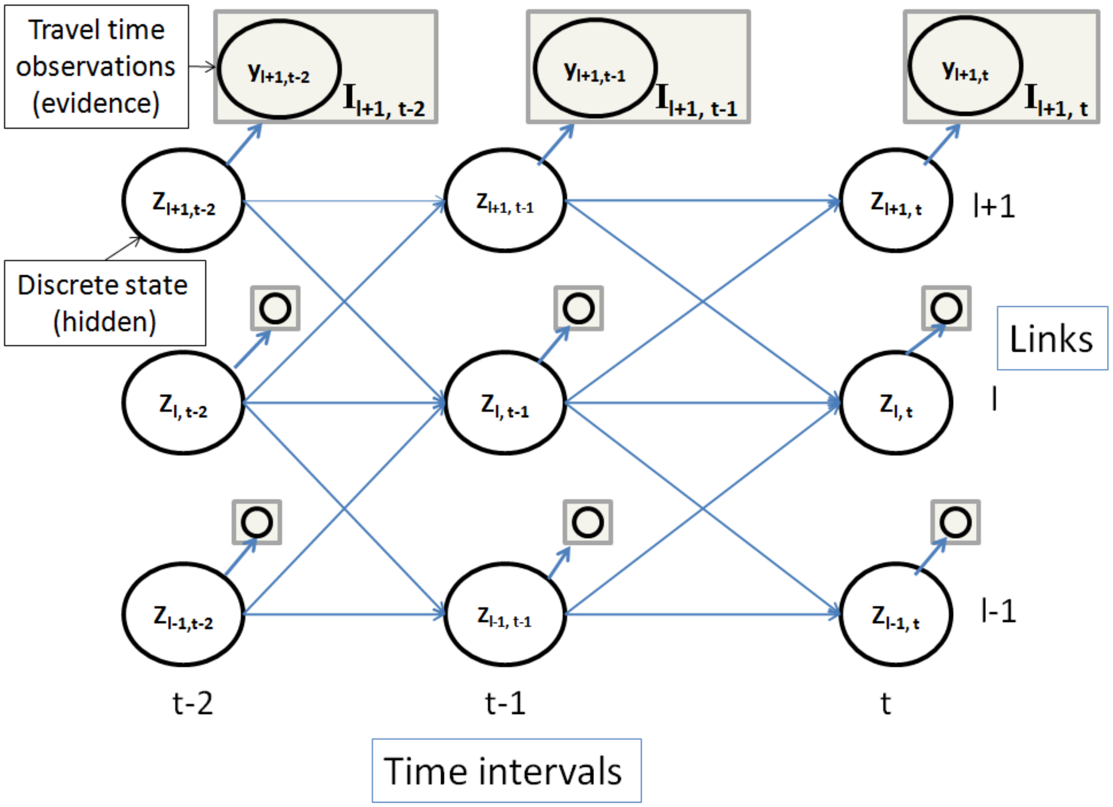


Figure 5.1: Spatio-temporal model of arterial traffic evolution represented as a coupled hidden Markov model. The circular nodes represent the (hidden) discrete state of traffic for each link at each time interval, denoted $Z_{l,t}$. The square nodes represent travel time observations from the distribution defined by the traffic state, denoted $\mathbf{y}_{l,t}$.

indicate the local spatial dependency of links from one time period to the next. Each square node in the graph represents observations on the link to which it is attached (e.g. travel time from probe vehicles, flow data from loop detectors if available, etc.).

To completely specify the CHMM-based model, it is necessary to estimate (i) the initial state probabilities for each link, denoted $\pi_{l,s}$, (ii) the discrete transition probability distribution functions (assumption 3), denoted $A_{l,t}$, and (iii) the distribution of travel time on a link given the state of that link (assumption 2), denoted $g_{l,s,t}$. If data sources other than link travel times are available, then additional observation nodes can be attached to the hidden state nodes. They would be incorporated into the model by defining a probability distribution function for the data source given the hidden states.

For each link l and each time interval t, the probability of link l to be in state s at time $t+1$ given the state of its neighbors at time t is given by the *discrete transition probability distribution* function of link l. It is fully characterized by a matrix of size $S^{\mathbf{N}_n^l} \times S$, denoted

$A_{l,t}$. The element of line r and column s, $A_{l,t}(r,s)$, represents the probability of link l to be in state s at time $t+1$ given that the neighbors of l are in state r at time t.

A simplifying assumption for computational tractability is to assume that for each link l, the state transition matrix $A_{l,t}$ and the conditional travel time distribution function $g_{l,s,t}$ do not depend on time. They are denoted respectively by A_l and $g_{l,s}$ in the remainder of this dissertation. To relax this assumption, one can assume that these functions are piecewise constant in time and estimate them for each period of time during which the stationarity assumption is satisfied. It is also assumed that, given the state of a link, the travel time distribution on that link is independent from all the other random variables. In general, travel time distributions across links are not independent (due to light synchronization, platoons, and other factors). Future work will specifically address the challenge of using correlated distributions, which have the potential to capture more complex dynamics in the arterial road network (see chapter 7).

Chapter 6

Learning and Inference Algorithms for Traffic Estimation

The models presented in chapter 5 were all designed to leverage machine learning tools for solving them. This chapter takes each of the models and provides the detailed algorithms needed for using each one to estimate real-time arterial traffic conditions. These algorithms are the core of the work presented in this dissertation and the primary building blocks for future research on this topic.

6.1 Solution Methods for Regression Models

This section demonstrates how to solve the regression models presented in section 5.1. The solution methods for the logistic regression (section 6.1.1) and STARMA (section 6.1.2) models are given followed by two sets of experimental results (section 6.1.3), one using simulation data and one using GPS data from an experiment in Manhattan, New York. This work is based on a previously published article [54]. A summary of the notation used for the regression algorithms is provided in table 6.1.

6.1.1 Logistic Regression

Consider the estimator based on the logistic model (5.5) to estimate the congestion state $Q_{k,i}$ for a VTL pair i and time interval k. A reminder that the aggregate representative quantity for the incoming data for time interval k on link i using temporal and spatial aggregation levels r and s is denoted $\mathbf{Z}_{k,i}^{r,s}$ (see section 5.1.3 for details). Suppose that $Q_{k,i}$ is binary-valued, that is $Q_{k,i} \in \{0, 1\}$ and $M = 2$. $Q_{k,i} = 1$ (resp. $Q_{k,i} = 0$) corresponds to the VTL pair i during interval k being in the *congested mode* (resp. *undersaturated mode*). The estimator $\hat{Q}_{k,i}$ gives the conditional probability of the $Q_{k,i}$ given the dependent variables:

r	Temporal aggregation level, which is an integer representing how many previous time intervals to include in producing an estimate
s	Spatial aggregation level, which is an integer that indicates what order-level neighboring VTL pairs to include for the previous time intervals in producing an estimate
$\mathcal{N}^s(i)$	The set of s order neighbors for VTL pair i
$Q_{k,i}$	Congestion state of VTL pair i for time interval k
$\mathbf{Z}_{k,i}^{r,s}$	Aggregate representative quantity for VTL pair i during time interval k for temporal and spatial aggregation levels r and s
Z_k	Vector of aggregate travel times for all VTL pairs for time interval k (for STARMA model)
$w_{k,i}^{(n)}$	Spatial weights of order n for $Z_{k,i}$ for VTL pair i and time interval k (for STARMA model)
β_i	Regression parameters for VTL pair i
$f_{\beta_i}(\mathbf{Z}_{k,i}^{r,s})$	Estimation function for $Q_{k,i}$ for VTL pair i and time interval k (for logistic regression model)
$\varphi_i^{(n)}(Z_j)$	Spatially-weighted travel time function (for STARMA model)

Table 6.1: Notation used in the regression algorithms.

$$\begin{aligned}\hat{Q}_{k,i} = \mathbb{E}_{h_i,g_i}[Q_{k,i}|\mathbf{Z}_{k,i}^{r,s}] &= 1 \cdot \mathbb{P}_{h_i,g_i}[Q_{k,i} = 1|\mathbf{Z}_{k,i}^{r,s}] + 0 \cdot \mathbb{P}_{h_i,g_i}[Q_{k,i} = 0|\mathbf{Z}_{k,i}^{r,s}]\\ &= \mathbb{P}_{h_i,g_i}[Q_{k,i} = 1|\mathbf{Z}_{k,i}^{r,s}]\end{aligned}$$

Now using the definition of β_i from equation (5.5), the conditional probability of $Q_{k,i}$ given the aggregate travel time for r temporal and s spatial dependencies is written

$$\mathbb{P}_{h_i,g_i}[Q_{k,i}|\mathbf{Z}_{k,i}^{r,s};\beta_i] = [f_{\beta_i}(\mathbf{Z}_{k,i}^{r,s})]^{Q_{k,i}}[1 - f_{\beta_i}(\mathbf{Z}_{k,i}^{r,s})]^{1-Q_{k,i}}$$

It is assumed that for a VTL pair i, the response process $\{Q_{k,i}\}$ and the covariate process $\{\mathbf{Z}_{k,i}^{r,s}\}$ is available for a number of time intervals $k = 0, \ldots, K$. Introducing the conditional independence assumption that the response variable $Q_{k,i}$ is independent of all other data given $\mathbf{Z}_{k,i}^{r,s}$. Then the joint conditional probability of $\{Q_{k,i}\}$ given $\{\mathbf{Z}_{k,i}^{r,s}\}$ (also known as the conditional likelihood) can be expressed as

$$\mathbb{P}_{h_i,g_i}[\{Q_{k,i}\}_{k=0}^{K}|\{\mathbf{Z}_{k,i}^{r,s}\}_{k=0}^{K};\beta_i] = \prod_{k=0}^{K}[f_{\beta_i}(\mathbf{Z}_{k,i}^{r,s})]^{Q_{k,i}}[1 - f_{\beta_i}(\mathbf{Z}_{k,i}^{r,s})]^{1-Q_{k,i}}$$

For a given training data $\{Q_{k,i}\}_{k=0}^{K}$ and $\{\mathbf{Z}_{k,i}^{r,s}\}_{k=0}^{K}$, the *best* estimate of parameter β_i is obtained by maximizing the logarithm of the conditional likelihood which is stated explicitly as follows:

$$\mathcal{L}(\beta_i; \{Q_{k,i}\}_{k=0}^K, \{\mathbf{Z}_{k,i}^{r,s}\}_{k=0}^K) = \sum_{k=0}^{K} \left(Q_{k,i} \cdot \beta_i^\top \mathbf{Z}_{k,i}^{r,s} - \log \left[1 + \exp \left(\beta_i^\top \mathbf{Z}_{k,i}^{r,s} \right) \right] \right)$$

The optimal estimate so obtained and denoted β_i^*, is called the *maximum likelihood estimate* (MLE). A number of standard iterative methods, all similar to the Newton-Raphson method, can be used to obtain the MLE β_i^*. Examples of such method include Fisher scoring method and the iterative reweighted least squares. The details of the algorithm can be found in [75].

Once the parameters are learned, *validation* can be done on a similar data set as the one used to obtain β_i^*. Validation is done to assess the ability of the learned model to correctly estimate the traffic status (congestion state in this case) on previously unseen data.

6.1.2 STARMA

The STARMA model is a more efficient estimator than the simple linear regression model (5.4). The number of parameters to be estimated for (5.4), given by $r \times |\mathcal{N}^s(i)| + 1$, can increase significantly as the spatial dependency s increases. In order to explain the model, the *spatio-temporal autoregressive* (STAR) model is presented first and subsequently generalized to a full STARMA model.

Following (5.1), the set of n order neighbors ($0 \leq n \leq s$) for a VTL pair i can be expressed as follows

$$\mathcal{N}^s(i) = \left(\bigcup_{n=0}^{s} \mathcal{N}^n(i) \right) \backslash \mathcal{N}^{n-1}(i),$$

where $\mathcal{N}^0(i) \backslash \mathcal{N}^{-1}(i) = \{i\}$ by convention. Now, for the linear regression model (5.4), for any temporal order j, $(k - r \leq j < k)$ and spatial order n, $(0 \leq n \leq s)$, assume that

$$\text{for all } v \in \mathcal{N}^n(i) \backslash \mathcal{N}^{n-1}(i), \quad \beta_i^{j,v} \equiv \beta_i^{j,n}, \tag{6.1}$$

and the definition of *n-th order, spatially-weighted travel time* as

$$\varphi_i^{(n)}(Z_k) = \frac{\sum_{l \in \mathcal{N}^n(i) \backslash \mathcal{N}^{n-1}(i)} w_{k,i}^{(n)} \mathbf{Z}_{k,i}}{\sum_{l \in \mathcal{N}^n(i) \backslash \mathcal{N}^{n-1}(i)} w_{k,i}^{(n)}}, \tag{6.2}$$

where $Z_j = (Z_{j,1}, \ldots, Z_{j,N})$ is the vector of aggregate travel times for all the N VTL pairs during time interval j and $w_{i,l}^{(n)}$ are the pre-defined *spatial weights of order* n for $Z_{j,l}$.

The goal of the STAR model is to predict a future $\mathbf{Z}_{k,i}^{r,s}$ from currently available data. Under the assumption (6.1) and the definition (6.2), the STAR model of *autoregressive* (AR) temporal order r and spatial order s is

$$\mathbf{Z}_{k,i} = \sum_{j=k-r}^{k-1} \sum_{n=0}^{s} \beta_i^{j,n} \varphi_i^{(n)}(Z_j) + \epsilon_{k,i} \tag{6.3}$$

where $\epsilon_{k,i}$ is the normally distributed error term with variance σ^2 with the properties that $\mathbb{E}[\epsilon_{k,i}] = 0$ for all k and $i \in \mathcal{V}$; and for all $i, j \in \mathcal{V}$

$$\mathbb{E}[\epsilon_{k,i}\epsilon_{k+s,j}] = \begin{cases} \sigma^2 & \text{if } s = 0 \\ 0 & \text{otherwise.} \end{cases}$$

The number of parameters to be estimated for the STAR model (6.3), including σ^2, is $r(s+1)+1$ which is (typically) much smaller than $r \times \mathcal{N}^s(i)+1$ for (5.4). The STAR model can now be generalized to STARMA model of autoregressive temporal order r and spatial order s, and *moving average* (MA) temporal order p and spatial order q as[1]

$$Z_{k,i} = \sum_{j=k-r}^{k-1} \sum_{n=0}^{s} \beta_i^{j,n} \varphi_i^{(n)}(Z_j) - \sum_{j=k-p}^{k-1} \sum_{n=0}^{q} \alpha_i^{j,n} \varphi_i^{(n)}(\epsilon_j) + \epsilon_{k,i}, \tag{6.4}$$

where $\epsilon_j = (\epsilon_{j,1}, \dots, \epsilon_{j,N})^\top$.

Here $\alpha_i^{j,n}$ are the moving average parameters. The total number of parameters (including σ^2) to be estimated for the STARMA model (6.4), denoted as STARMA(r, s, p, q) are $r(s+1)+p(q+1)+1$.

Following [79], assume that the STARMA parameters are the same for all VTL pairs, that is, $\alpha_1^{j,n} = \ldots = \alpha_N^{j,n} \equiv \alpha_{j,n}$ and $\beta_1^{j,n} = \ldots = \beta_N^{j,n} \equiv \beta_{j,n}$. Then model (5.4) can be vectorized for all VTL pairs $i \in \mathcal{V}$ as

$$Z_k = \sum_{j=k-r}^{k-1} \sum_{n=0}^{s} \beta^{j,n} \Phi^{(n)}(Z_j) - \sum_{j=k-p}^{k-1} \sum_{n=0}^{q} \alpha^{j,n} \Phi^{(n)}(\epsilon_j) + \epsilon_k. \tag{6.5}$$

where $\Phi^{(n)}(\cdot) = (\varphi_1^{(n)}(\cdot), \dots, \varphi_N^{(n)}(\cdot))^\top$ and $\epsilon_k = (\epsilon_{k,1}, \dots, \epsilon_{k,N})^\top$.

For given training data $\{Z_k\}$, $(k = 0, \dots, K-1)$, the best estimate of the parameters $A := [\alpha^{j,n}]_{p\times(q+1)}$, $B := [\beta^{j,n}]_{r\times(s+1)}$ and σ^2 is given by maximizing the conditional likelihood expressed as

$$\mathbb{P}(\{Z_k\}_{k=0}^{K-1}; A, B, \sigma^2) = (2\pi)^{-\frac{KN}{2}} \left|\sigma^2 \mathbf{I}_{KN\times KN}\right|^{-\frac{1}{2}} \exp\left(-\frac{S(A,B)}{2\sigma^2}\right) \tag{6.6}$$

[1] More generally, the AR spatial order s (resp. the MA spatial order q) can vary with the temporal order r (resp. p).

where $I_{KN\times KN}$ is the identity matrix, $S(A,B) := (\epsilon_0,\ldots,\epsilon_{K-1})^\top(\epsilon_0,\ldots,\epsilon_{K-1})$ and according to (6.5), ϵ_k is written

$$\epsilon_k = Z_k - \sum_{j=k-r}^{k-1}\sum_{n=0}^{s}\beta^{j,n}\Phi^{(n)}(Z_j) + \sum_{j=k-p}^{k-1}\sum_{n=0}^{q}\alpha^{j,n}\Phi^{(n)}(\epsilon_j).$$

The *maximum likelihood estimate* parameters, denoted A^*, B^*, are obtained by maximizing the logarithm of the conditional likelihood (6.6), and the corresponding σ^* is estimated by

$$\sigma^* = \sqrt{\frac{S(A^*,B^*)}{KN}}.$$

Additional technical details for the STARMA model can be found in [79].

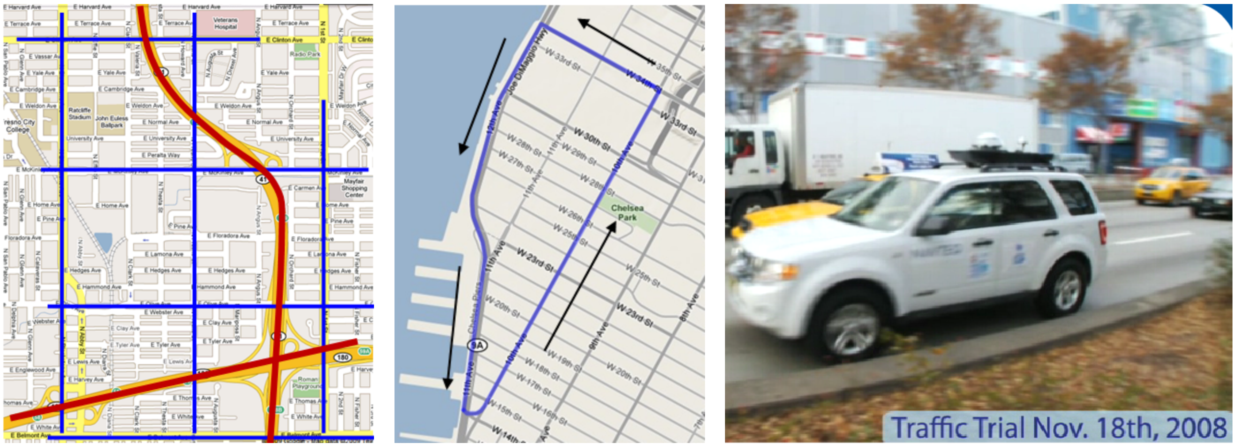
6.1.3 Results

The results from logistic regression-based classification and STARMA-based continuous linear regression are presented here. Each algorithm is implemented and tested on simulation and field experiment data. A framework for quantifying accuracy is introduced. Results are then presented for one-step forecast, followed by multi-step forecast for the STARMA model. Additionally, a study of the effect of the *penetration rate* on the forecast accuracy is presented.

Simulation and Field Experiment Data

There are two data sets used to validate the regression models. The first set was generated from Paramics micro-simulation software. The road network modeled consists of 1,961 nodes, 4,426 links, 210 zones and is based on the SR41 corridor in Fresno, CA. The analysis presented here is for a sub-network that includes 9 arterial roads, 20 signals and 15 stop signs. Paramics simulates every car in the network. From this simulation, the position of every vehicle at one-second time intervals is extracted. This provides detailed information about speed and travel time through the network. The sub-network studied here includes 380 different links, each one of which is characterized with a specific length, a number of lanes, a direction, a speed limit and signal information. 99 VTLs were placed on different links, which corresponds to 156 different pairs of VTLs, in order to capture travel times along links and through intersections.

The second data set was obtained as part of the official *Mobile Millennium* launch demonstration in New York City at the *ITS World Congress.* Twenty drivers, each carrying a GPS equipped cell phone, drove for 3 hours (9:00am to 12:00pm) around a 2.4 mile loop of Manhattan (see figure 6.1). This number of drivers constituted approximately 2% of the total vehicle flow through the road of interest. The experiment was repeated 3 times in order to use two of the experiments as training data for the models and the other to validate



(a) Paramics Map (b) New York Map (c) Test Vehicle

Figure 6.1: **Experiment Design.** (a) Map of the Paramics network in Fresno, CA. (b) Experiment route for New York City field test used to collect the data (arrows represent the direction of traffic of probe vehicles). (c) Test vehicle used for the New York test.

the model results. The operational capabilities of the system were demonstrated at the *ITS World Congress* [58] on November 18, 2008, when live arterial traffic was displayed for conference attendees.

Validation Framework

In order to compute the accuracy of the model, one needs to define the "ground truth" state of traffic. In this set of experiments, travel times are aggregated into a single value per time interval (5 minutes for Paramics, 15 minutes for the New York test). This single value per time interval is considered the true state for the interval. Determining ground truth for the logistic regression method requires classifying each time interval as congested or uncongested. The STARMA method uses the average travel time during each interval as the ground truth value. Both of these methods correspond to choosing appropriate $h_i(\cdot)$ and $g_i(\cdot)$ functions as described in section 5.1.3.

The aggregation function $h_i(\cdot)$ should capture the pattern of change in pace over different intervals to provide an aggregate quantity that is sufficiently representative of the congestion state, thus providing better accuracy in training the model and obtaining the logistic regression parameters. Based on extensive testing and simulation, it is observed that aggregating the travel times based on the entire data available in an interval fails to capture the congestion state due to the high variance of travel times when a link is congested. The probes most effected by congestion should thus have more weight in the aggregation process. A simple yet fairly effective data-driven aggregation method is as follows: given the set of observations for VTL pair i and interval k, $A_{k,i}$ is sorted such that $t_{m_1} < t_{m_2} \implies X_{t_{m_1},i} > X_{t_{m_2},i}$, then

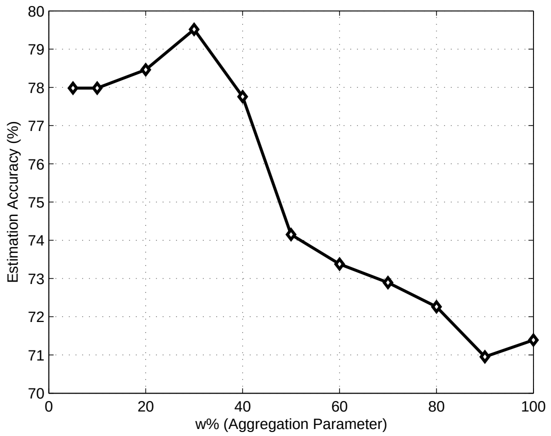


Figure 6.2: Average estimation accuracy vs. aggregation parameter w.

take

$$Z_{k,i} = h_i(\{X_{t_m,i} \,|\, (k-1)t \leq t_m < kt\}) := \frac{1}{w.M_{k,i}} \sum_{m=1}^{\lfloor w.M_{k,i} \rfloor} X_{t_m,i},$$

where $M_{k,i}$ is the number of observations in $A_{k,i}$ and $0 < w \leq 1$ is the fraction of observations used for aggregation. The symbol $\lfloor a \rfloor$ denotes the floor value of a. In words, the aggregate pace is the mean of the $100 \times w\%$ observations with highest pace or equivalently the worst observations. The simulation results for different values of w are shown in figure 6.2. From an application-driven point of view, the w that maximizes estimation accuracy is selected, in the present case $w = 0.3$. At this value, the travel time envelope of the time series of observations is best captured.

The training phase of logistic regression requires as input a congestion threshold along with the aggregate travel times $Z_{k,i}$. Since the congestion threshold should be chosen to be consistent with the choice of aggregate travel times to provide meaningful classification, the congestion threshold, T_i, is defined as the mean of the $100 \times w\%$ observations in D_i with highest travel time where D_i is the set of available observations in all intervals and w is essentially be the same value chosen for aggregation ($w = 0.3$ in this section). This corresponds to choosing

$$Q_{k,i} = g_i(\{X_{t_m,i}|(k-1)t \leq t_m < kt\}) = I(h_i(\{X_{t_m,i} \,|\, (k-1)t \leq t_m < kt\}) > T_i),$$

where $I(\cdot)$ is the indicator function. The STARMA model does not use a g_i function because it forecasts a continuous quantity.

The logistic regression algorithm produces a probability of congestion for each VTL pair studied. If this probability is greater than .5, then the forecasted state is congested. The accuracy of the logistic regression forecasts is defined as the percentage of correctly forecasted states over all intervals and VTL pairs studied. For the STARMA model, the accuracy is defined as the mean percentage error between the forecasted travel time value and the actual travel time value as defined by the h_i function described earlier.

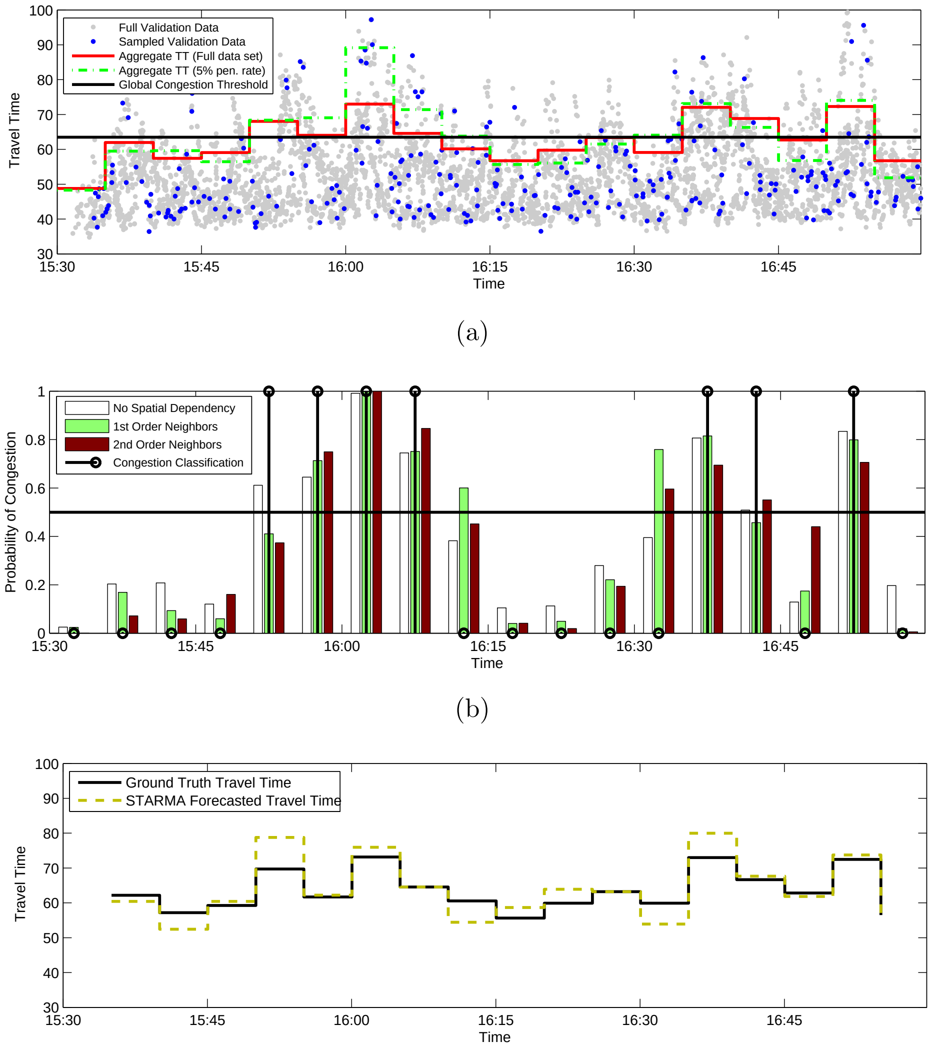
Short-Term Forecast

Both regression methods are designed to do one-step (short-term) forecasts. For each data set (as described in section 6.1.3), the performance of each model was evaluated by dividing the data set into a training set and a validation set. For the Paramics simulation data, the training set consisted of three simulation runs and the validation set consisted of a separate, fourth simulation run. For the New York experiment data, two days of data were used for training and the other day for validation. Through a-priori experimentation, the temporal dependency for the logistic regression model was set to $r = 1$ for the logistic regression, $r = 2$ for the STARMA model. The spatial dependency is varied for comparison in the result figures described in the following paragraph.

The Paramics simulations give information about every vehicle. For testing the methods, only a subset of the data is used for training and inference, corresponding to the penetration rate. This was incorporated into the following analysis by requiring each regression method to produce estimates for the validation data set using only a small percentage of the available travel times. Figure 6.3 displays the one-step forecast results of the logistic regression and STARMA methods on the Paramics validation set respectively, using a penetration rate of 5%. Similarly, figure 6.4 displays the one-step forecast results on the New York validation set.

Penetration Rate Study

The value of 5% for the penetration rate used in the previous subsection was chosen based on the prospects for future adoption of GPS equipped cell phones running traffic information software (such as that provided by *Mobile Millennium*). Therefore, a study of the effect of the penetration rate on results is of interest to quantify the influence of technology adoption on estimation and forecast accuracy. Figure 6.5 shows the one-step forecast accuracy for the logistic regression and STARMA methods as a function of the penetration rate. From these figures, one can infer that 2% penetration rate can give reasonably good results, while 5% and higher give very accurate results. Also note that using spatial neighbors of order 1 (direct neighbors) generally provides better results. One can interpret this as indicating that second order neighbors lead to an overfit model while no neighbors lead to an underfit model.

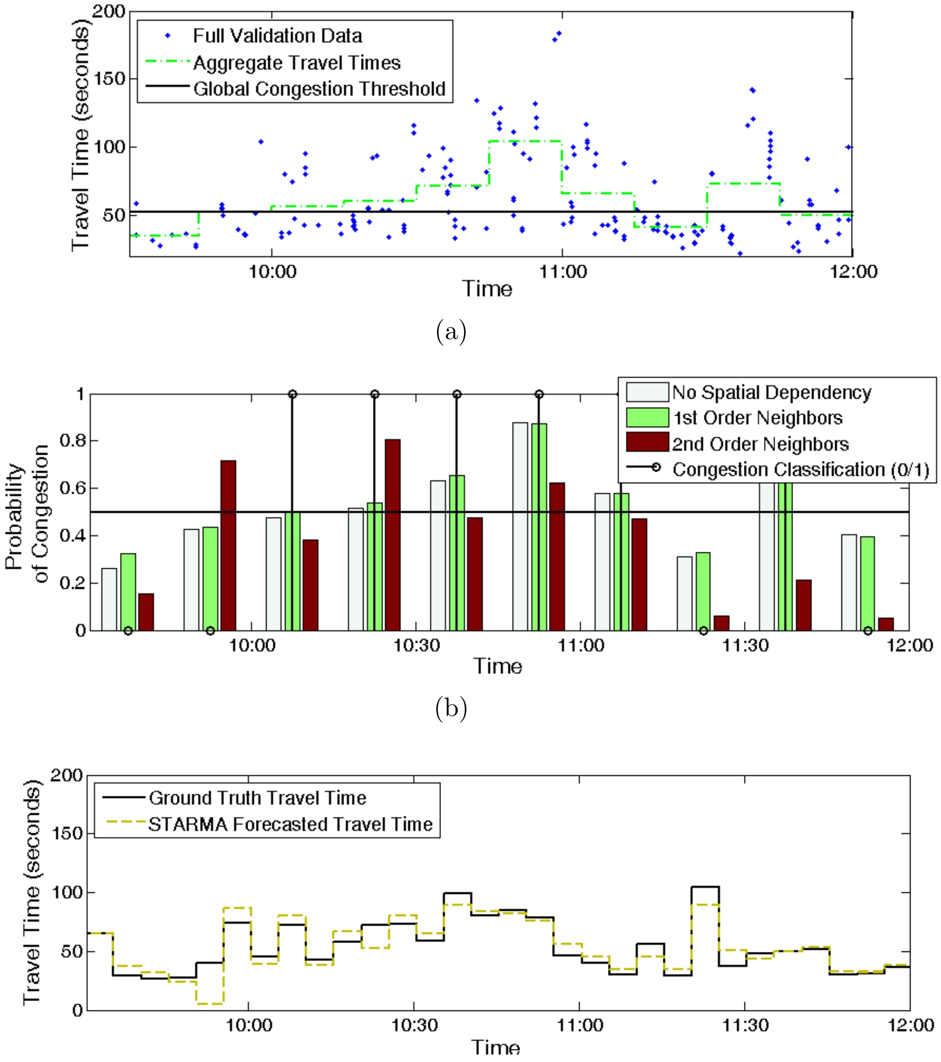


(a)

(b)

(c)

Figure 6.3: **One-step forecast validation results on a given VTL pair of the Paramics simulation network (penetration rate: 5%)**. (a) Travel time data of the VTL pair and its aggregate value on 5 minutes time intervals. Both the data and the aggregate value are shown for the whole data set and for a 5 % penetration rate. (b) One-step forecast of the congestion state produced by the logistic regression algorithm. The bars represent the probability of congestion estimated by the models for different levels of spatial dependency. The real state of congestion is represented with circles. (c) One-step forecast of travel time produced by the STARMA algorithm.

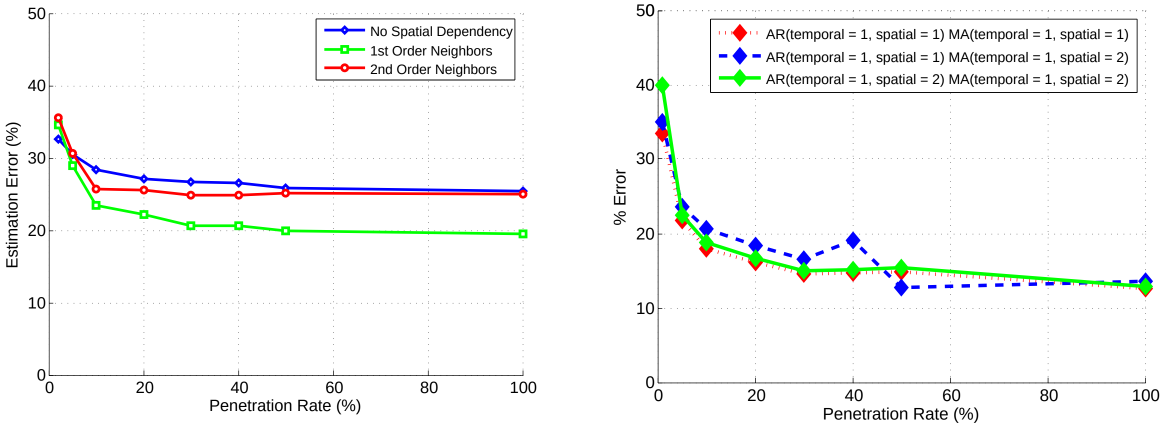


(a)

(b)

(c)

Figure 6.4: **One-step forecast validation results for logistic regression on one VTL pair of the New York network.** (a) Travel time data of the VTL pair and its aggregate value on 15 minutes time intervals. (b) One-step forecast of the congestion state produced by the logistic regression algorithm. The bars represent the probability of congestion estimated by the models for different levels of spatial dependency. The ground truth state of congestion is represented with circles. (c) One-step forecast of travel time produced by the STARMA algorithm.



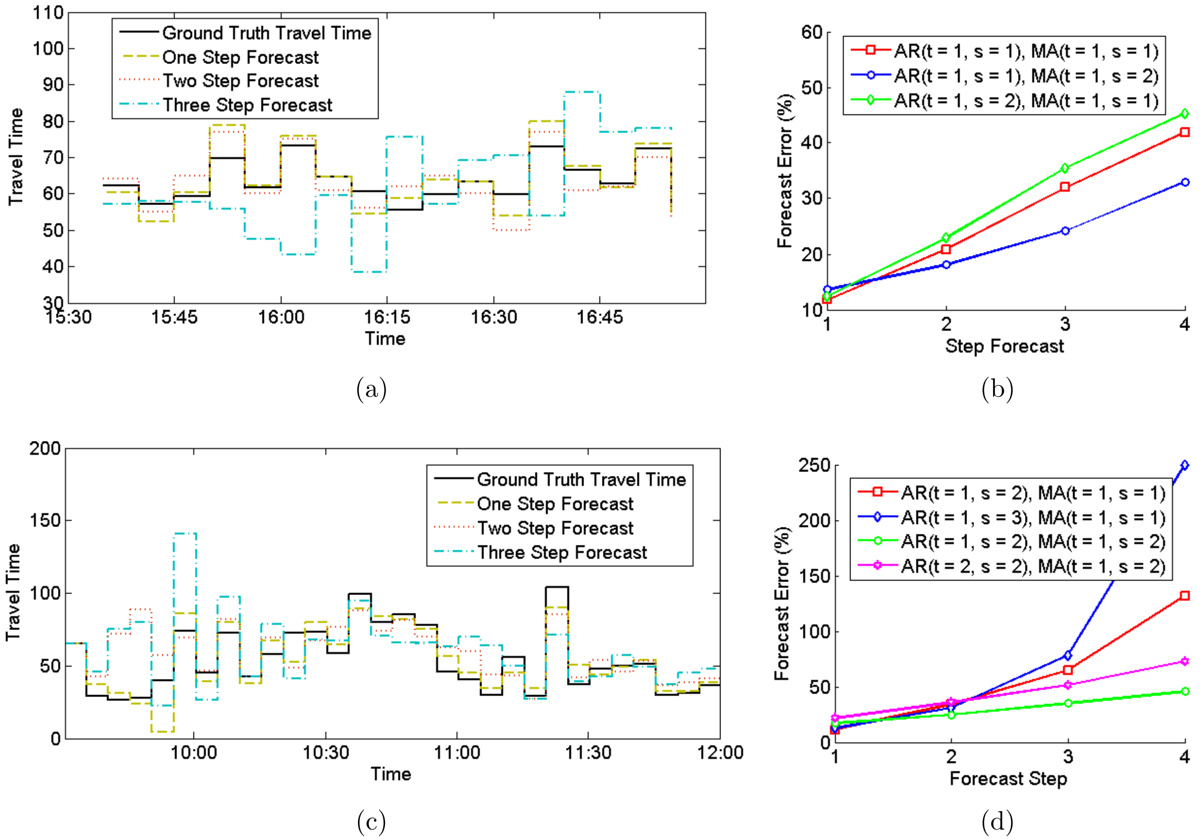
(a) (b)

Figure 6.5: **Average one-step forecast error vs. penetration rate for all VTL pairs in the Paramics dataset.** (a) Logistic Regresion Forecast Classification Error. (b) STARMA Travel Time Forecast Error.

Multi-Step Forecast

The STARMA model is capable of producing forecasts of any number of steps by using the output of the model as input for the next time interval. It is not straightforward to do the same for the logistic regression model since it has an output that is fundamentally different from the input it requires. Therefore, the discrete output of the logistic regression model must be transformed back to a continuous value in order to do forecast in the same way. This avenue was not considered in this work and is left as further research.

In this section, the results of multi-step forecast for the STARMA model are presented. Figure 6.6 shows the forecast results for the New York data set. The best results for the first step forecast are obtained for an autoregressive temporal order of 1, a spatial order of 2, a moving average temporal order of 1 and a spatial of 1. The two plots for which the moving average temporal and spatial orders are both equal to 1 show the best result for the first step forecast, but the error becomes quickly significant when the forecast step increases. On the other hand, the two other plots for which the moving average orders are one temporally and two spatially show a worse result for the first step forecast but considerably better results for more than one step. The choice of the parameters is therefore a very important step and should take into consideration the performance of the forecasting for more than one step ahead. Analysis of a larger data set is necessary to come to a statistically significant conclusion about the best way to chose the spatio-temporal parameters for the STARMA model.



(a)

(b)

(c)

(d)

Figure 6.6: **Forecast accuracy.** (a) and (c): Forecast error for a VTL pair in the Paramics and New York networks, respectively. (b) and (d): Average forecast error as a function of the number of forecast steps into the future, Paramics and New York networks, respectively. One step is 5 minutes. In (b) and (d), t represents the temporal dependency and s represents the spatial dependency.

$\mathcal{L}$	A set of links of the road network
$\mathcal{T}$	A set of historic time periods
t	A historic time period (such as Mondays from 2:00pm-2:15pm)
t_n	A current time interval (such as the current 15 minute period
$\mu_{l,t}$	The historic mean travel time for link l during time period t
$\sigma_{l,t}$	The historic travel time standard deviation for link l during time period t
$\hat{\mu}_{l,t_n}$	The current mean travel time for link l during current time period t_n
$\hat{\sigma}_{l,t_n}$	The current travel time standard deviation for link l during current time period t_n
$\mathbf{P}_t$	The set of probe path observations for time period t
y_p	The travel time for probe path observation $p \in \mathbf{P}_t$
$w_{p,l}$	The proportion of link l driven for path observation $p \in \mathbf{P}_t$
b_l	The minimum travel time for link l
$\mathbf{X}_{l,t}$	The set of travel times allocated to link l for time period t
$\mathbf{N}(y, \mu, \sigma)$	The natural logarithm of the Gaussian distribution for a given travel time y, mean μ and standard deviation σ

Table 6.2: Notation used in the historic and Bayesian real-time algorithms.

6.2 Solution Methods for Bayesian Model

Solving the estimation model presented in section 5.2 is a two-step process. First, the historic parameters of the network are learned (section 6.2.1) and then these parameters are used to perform a Bayesian update for real-time estimation (section 6.2.2). Learning the historic parameters is a process that is intended to be run infrequently, although the more often the historic parameters are re-learned to take into account changes in traffic patterns, the more accurate the real-time estimations will be. The real-time estimation step is structured to take advantage of the more intensive computations required in learning the historic parameters, which means that the real-time estimation step can be solved efficiently online.

In this section, the case where the link travel time distributions are assumed to be Gaussian distributions is considered. The parameters, $Q_{l,t}$ (see section 5.2), are denoted $\mu_{l,t}$ and $\sigma_{l,t}$ for the mean and standard deviation, respectively. The methodology extends to cases beyond the Gaussian distribution, but leads to more difficult optimization problems. The Gaussian case is presented here to show an example of the algorithm from start to finish in complete detail. A summary of the notation used for the algorithms from this section is provided in table 6.2.

6.2.1 Learning Historic Model Parameters

It is computationally challenging to solve the optimization problem in equation (5.7) because it is simultaneously solving for the mean and variance of every link in the network. It is possible to solve this problem directly if using a commercial grade non-linear optimization engine with a lot of computational power. It is assumed that such resources may not be available and an alternative solution strategy is proposed. The basic idea is to decouple the optimization into two separate subproblems, each of which is easier to solve on its own, and then to iterate between these subproblems until converging to an optimal solution. These two subproblems are *travel time allocation* and *parameter optimization*.

The insight into the decoupled solution approach is to realize that if it were known exactly how much each probe vehicle drove on each link of its path (instead of just the total travel time), it would be easy to estimate the mean and standard deviation for each link in the network (by simply taking the mean and standard deviation of all the link travel time observations). However, it is not known exactly how long each probe vehicle spent on each link without high frequency probe data. For sparsely-sampled data, the most likely amount of travel time spent on each link is determined instead. The problem in doing this is that computing the most likely link travel times is dependent upon the link travel time parameters (μ and σ) that need to be estimated. This would appear to a chicken-and-egg sort of problem, but there is a sound mathematical justification for iterating between these two steps. The link parameters are used to determine the most likely travel times and then the most likely travel times are used to update the parameters.

Travel Time Allocation

The travel time allocation problem assumes that estimates of the link parameters are available for all links of the network (which means that $\mu_{l,t}$ and $\sigma_{l,t}$ are fixed for this part of the algorithm). In addition to these link parameters, it is necessary to specify *lower bounds* on the travel time allocated for each link of the network, denoted b_l for any link $l \in \mathcal{L}$. These bounds should be computed conservatively using the maximum speed that is realistically possible for the link, which is likely to be quite a bit higher than the speed limit (on a 25 mph arterial, one might choose $40 - 50$ mph to compute the minimum link travel time). For each observation $p \in \mathbf{P}_t$, the goal is to solve the following optimization problem

$$\begin{array}{rrcl} \arg\max\limits_{x} & \sum\limits_{l \in L_p} \mathbf{N}(x_{p,l}, w_{p,l}\mu_{l,t}, \sqrt{w_{p,l}\sigma^2_{l,t}}) & & \\ \text{s.t.} & \sum\limits_{l \in L_p} x_{p,l} & = & y_p \\ & x_{p,l} & \geq & w_{p,l}b_l, \forall l \in L_p, \end{array} \tag{6.7}$$

where the goal is to obtain the probe vehicle's link travel time ($x_{p,l}$) for each link on that vehicle's path ($l \in L_p$) subject to the constraint that the sum of the link travel times is equal to the observed travel time. There are also lower bounds ($w_{p,l}b_l$) on the link travel time variables to ensure a sensible solution is returned. Algorithm 2 gives the details for how to solve optimization problem 6.7. The optimal solution is primarily contained in lines 11-15, which computes the total expected path variance (V) and the difference between expected and actual travel times (Z). With these two quantities, each link is allocated the expected link travel time adjusted by some proportion of Z, where this proportion is computed using the link variance divided by the total path variance. This procedure can lead to some links being allocated a travel time below the minimum for that link. The set $\mathbf{J}$ is introduced to track the links with initial allocated travel times below the lower bound and the main procedure is repeated by setting the travel times for these links to the lower bound and optimizing with respect to the remaining links. Note that for some path observations, it may not be possible to produce a travel time allocation that satisfies the constraints. This means that the vehicle traveled faster than the bounds dictated was possible. In this case, this observation is considered an outlier and is discarded from the set of observations. The final data structure that is kept from this part of the algorithm is denoted $\mathbf{X}_{l,t}$ which contains all of the individual travel times allocated to link $l \in \mathcal{L}$ for time period $t \in \mathcal{T}$. These travel times are scaled by the proportion of the link traveled, which explains the division by $w_{p,l}$ on line 19 of algorithm 2.

Parameter Optimization

Given $\mathbf{X}_{l,t}$ (a vector of allocated travel times for link l during time period t) from algorithm 2, it is straightforward to optimize the model parameters for each link ($\mu_{l,t}$ and $\sigma_{l,t}$). As was stated in the previous section, time period t is fixed for these computations and the algorithms are repeated for each time period independently. Algorithm 3 provides the detailed procedure for computing new values for these parameters. For this algorithm, $\mathbf{X}_{l,t}(i)$ denotes the ith element of $\mathbf{X}_{l,t}$ from algorithm 2.

Full Historic Arterial Traffic Algorithm

Putting the travel time allocation and parameter optimization pieces together, the entire historic traffic model is presented in algorithm 4. A global parameter is required for this algorithm, which is the maximum number of iterations $M^{\max}$ (to go back and forth between travel time allocation and parameter optimization).

When first running the historic model, it is important to run it iteratively as described in algorithm 4. After this initial run, future observations can be incorporated into the historic model in two ways. The first and most robust way is to run the entire algorithm 4 as before using all previous and newly acquired data. A second way which is less computationally intensive is to do an incremental update of the model parameters, $\mu_{l,t}$ and $\sigma_{l,t}$. Algorithm 5 describes how to do this incremental version. It would be appropriate to do this incremental version to update the historic model weekly. It is ideal to re-run the full version (algorithm 4) every few months.

Algorithm 2 Travel Time Allocation.

```
Require: $t \in \mathcal{T}$ is fixed to some particular time period.
for $l \in \mathcal{L}$ do
    $\mathbf{X}_{l,t} = \emptyset$ {Initialize allocated travel time sets to be empty.}
end for
for $p \in \mathbf{P}_t$ do {For all probe path observations.}
    if $\sum_{l \in L_p} w_{p,l} b_l > y_p$ then
        Travel time allocation infeasible for this path. This means that the observation represented travel that is considered faster than realistically possible, so the observation is considered an outlier. Remove $p$ from $\mathbf{P}_t$.
    else
        $\mathbf{J} = \emptyset$ {$\mathbf{J}$ contains all links for which the travel time allocation is fixed to be equal to the lower bound.}
        repeat
            $x_{p,l} = w_{p,l} b_l, \forall l \in \mathbf{J}$ {For all links that had an infeasible allocation in the previous pass through this loop, set the allocation to the lower bound.}
            $V = \sum_{l \in L_p \setminus \mathbf{J}} w_{p,l} \sigma^2_{l,t}$ {Calculate the path variance for the links not fixed to the lower bound.}
            $Z = y_p - \sum_{l \in \mathbf{J}} w_{p,l} b_l - \sum_{l \in L_p \setminus \mathbf{J}} w_{p,l} \mu_{l,t}$ {Calculate the difference between expected and actual travel time for the links not fixed to the lower bound.}
            for $l \in L_p$ do {Allocate excess travel time in proportion of link variance to path variance.}
                $x_{p,l} = w_{p,l} \mu_{l,t} + \frac{w_{p,l} \sigma^2_{l,t}}{V} Z$
            end for
            $\mathbf{J} = \mathbf{J} \cup \{l \in L_p : x_{p,l} < w_{p,l} b_l\}$ {Find all links violating the lower bound.}
        until $x_{p,l} >= w_{p,l} b_l, \forall l \in L_p$
        for $l \in L_p$ do
            $\mathbf{X}_{l,t} = \mathbf{X}_{l,t} \cup \left( \frac{x_{p,l}}{w_{p,l}} \right)$ {Add the allocated travel time to $\mathbf{X}_{l,t}$.}
        end for
    end if
end for
return $\mathbf{X}_{l,t}, \forall l \in \mathcal{L}$
```

Algorithm 3 Parameter Optimization.

```
Require: $t \in \mathcal{T}$ is fixed to some particular time period.
Require: Input: $\mathbf{X}_{l,t}, \forall l \in \mathcal{L}$ as returned from the travel time allocation.
for $l \in \mathcal{L}$ do
    if $|\mathbf{X}_{l,t}| < \tilde{n}$ then
        Link has too little data. Keep original $\mu_{l,t}$ and $\sigma_{l,t}$, or use values from other time periods if appropriate. The estimate for this link should be given a very low confidence value. $\tilde{n}$ represents the minimum number of observations to have a reliable estimate of the mean and variance. This should be set to at least 10.
    else
        $\mu_{l,t} = \frac{1}{|\mathbf{X}_{l,t}|} \sum_{i=1}^{|\mathbf{X}_{l,t}|} \mathbf{X}_{l,t}(i)$ {Compute the sample mean.}
        $\sigma_{l,t} = \sqrt{\frac{1}{|\mathbf{X}_{l,t}|} \sum_{i=1}^{|\mathbf{X}_{l,t}|} \left(\mathbf{X}_{l,t}(i) - \mu_{l,t}\right)^2}$ {Compute the sample standard deviation.}
    end if
end for
return $\mu_{l,t}, \sigma_{l,t}, \forall l \in \mathcal{L}$
```

Algorithm 4 Historic Arterial Traffic.

```
Require: $t \in \mathcal{T}$ is fixed to some particular time period.
Initialize the model parameters $\mu_{l,t}$ and $\sigma_{l,t}$ to typical values based on the physical characteristics of the links using guidelines from transportation handbooks such as the Highway Capacity Manual [20].
for $i = 1$ to $M^{\max}$ do
    Use $\mu_{l,t}, \sigma_{l,t}$ for all $l \in \mathcal{L}$ to obtain $\mathbf{X}_{l,t}$ using algorithm 2.
    Use $\mathbf{X}_{l,t}$ to compute new values of $\mu_{l,t}, \sigma_{l,t}$ using algorithm 3.
end for
return $\mu_{l,t}, \sigma_{l,t}, \mathbf{X}_{l,t}, \forall l \in \mathcal{L}$
```

Algorithm 5 Incremental update of historic traffic parameters.

```
Require: $t \in \mathcal{T}$ is fixed to some particular time period.
Require: $\mu_{l,t}, \sigma_{l,t}, \mathbf{X}_{l,t}, \forall l \in \mathcal{L}$ as given by the last run of the historic algorithm.
Run the travel time allocation (algorithm 2) on the new observations $\tilde{\mathbf{P}}_t$ using $\mu_{l,t}, \sigma_{l,t}$ and denote $\tilde{\mathbf{X}}_{l,t}$ the returned values from the algorithm.
$\mathbf{X}_{l,t} = \mathbf{X}_{l,t} \cup \tilde{\mathbf{X}}_{l,t}$
Run the parameter optimization (algorithm 3) using $\mathbf{X}_{l,t}$ to obtain new values for $\mu_{l,t}, \sigma_{l,t}$.
```

6.2.2 Bayesian Real-time Traffic Estimation

The real-time model uses the output of the travel time allocation to perform a Bayesian update of the link parameters. Algorithm 6 provides the details for the complete algorithm to go from path-inferred probe data to travel time distributions for each link. This requires specifying the *current time window* (t_1, t_2). Denote t_2 as the current time and t_1 as some amount of time back in the past that must be specified. Both quantities are defined such that the algorithm is run once every $t_2 - t_1$ amount of time. The frequency at which the model should be run is dependent upon the amount of data coming in real-time. The frequency should be at least as high as the historic model, so if the historic model produces estimates for each 15 minute period of the day, then the real-time model should be run at least once per 15 minutes. If the data volume is large, the model can be run up to every 5 minutes. Running the model more frequently will likely not increase the performance and may lead to estimates that fluctuate too much.

The algorithm works by considering the historic link travel time distributions as *priors* on the real-time distribution. The variance is not updated as part of the real-time algorithm because it is considered to be constant within a historic time period (primarily due to a lack of data to be confident in a real-time estimate of the variance). In algorithm 6, lines 5-6 define the prior parameters (denoted α and β, which are from the *conjugate normal* prior distribution [84] used here) based on the historic travel time distribution. This requires specifying ν, which expresses the precision of the estimate of the historic mean $\mu_{l,t}$. It is chosen based on experimentation to determine how much real-time traffic conditions can deviate from the historic value. The ν parameter allows one to give more or less weight to real-time data. Line 7 computes the average of the current data. Line 8 computes the current estimate of the mean for the link travel time distribution using the formula for performing a Bayesian update [84]. The normal prior is used because it is a computationally efficient (conjugate) prior when the observations (link travel times) are normally distributed with known mean (see [84] for details on the expression of the prior).

6.3 Solution Methods for Probabilistic Graphical Model

In this section, the solution methods for solving the graphical model presented in section 5.3 are introduced. These solution methods employ both particle filtering and the Expectation Maximization algorithm, both of which are introduced in more detail in section 3.2.

The way the graphical model is constructed gives rise to the following paradox. Given the parameters of the model, it is possible to estimate the most likely state of the links given observations and their evolution over time. Similarly, given the state of the links of the network over a period of time, it is possible to estimate the parameters of the model (state transition matrix, and conditional travel time probability distributions). This well

Algorithm 6 One time step of the real-time algorithm.

```
Require: $t_2$ is the current time and $t_1$ is the time of the last estimate.
Select $\mathcal{P}_{t_1,t_2}$, the set of current probe path observations.
Run the travel time allocation algorithm over the set $\mathcal{P}_{t_1,t_2}$ and obtain the link travel time sets $\mathbf{X}_{l,t_2}$, $\forall l \in \mathcal{L}$.
Define $n_{l,t_2} := |\mathbf{X}_{l,t_2}|$, $\forall l \in \mathcal{L}$.
for $l \in \mathcal{L}$ do
    $\alpha = \mu_{l,h(t_2)}$
    $\beta = \nu\sigma_{l,h(t_2)}$
    $\bar{x} = \frac{1}{n} \sum_{x_i \in \mathbf{X}_{l,t_2}} x_i$ {The average of the current observations. $x_i$ is the $i$th element of $\mathbf{X}_{l,t_2}$.}
    $\hat{\mu}_{l,t_2} = \dfrac{\dfrac{\alpha}{\beta} + \dfrac{\bar{x}}{\sigma_{l,h(t_2)}}}{\dfrac{1}{\beta} + \dfrac{n_{l,t_2}}{\sigma_{l,h(t_2)}}}$ {The updated mean travel time corresponds to a weighted average between the historic mean and the real-time mean.}
end for
```

known type of problem is solved using an EM algorithm which iterates between finding the probability of each state for each link of the network and each time interval given some values of the model parameters (E step). Then, the probabilities of each state for each link and each time interval are used to update the value of the parameters by maximizing the log likelihood (M step).

A high-level description of the parameter estimation is presented in algorithm 7 (this work is based on a previously published article [53]). A summary of the notation used for the graphical model algorithms is provided in table 6.3. In general, bold notation refers to the vectorized version of a particular variable (e.g. $\mathbf{a} = \{a_i | \forall i\}$).

6.3.1 State Estimation

The goal of this section is to describe how to perform state estimation given the parameters of the model. This also turns out to be the E-step of the EM algorithm as will be described later.

The challenge of the graphical model approach is that the link travel times are not directly observed since the probe observations received can span several links of the network between two consecutive measurements. This difficulty is addressed by computing the most likely link travel times that make up the path of the probe vehicle (*travel time allocation*). It is possible to have a graphical model representation that does not have this decomposition approach, but it leads to a difficult non-linear parameter optimization (M-step) problem, for which

N	Number of links of the road network
S	Number of congestion states per link
D	Number of days in the training data set
T_d	Number of time intervals for day $d \in D$
$g_{l,s}(\cdot)$	The travel time probability density function for link l when in congestion state s
$P_{l,s}$	The parameters of the probability density function $g_{l,s}(\cdot)$
$z^s_{d,t,l}$	The probability of link l being in congestion state s for time period t on day d
$q^{s,r}_{d,t,l}$	The probability of link l being in congestion state s for time period t on day d given that its neighboring links are in state r
A_l	The state transition probability matrix for link l with respect to its neighbors
$\pi_{l,s}$	The initial probability that link l begins the day in state s
$I_{d,t,l}$	The set of probe observations (after travel time allocation) for day d, time t and link l
$\alpha_{x^l_1,x^l_2}$	The proportion of travel time of link l when driving the partial distance from start location x^l_1 to end location x^l_2.

Table 6.3: Notation used in the graphical model algorithms.

the number of variables increase quadratically in the number of links. This optimization problem would require an approximation technique to solve, which is why a more intuitive decomposition scheme called *travel time allocation* is proposed. For data from fixed location sensors, this is not an issue as one simply needs to define an appropriate parameterized observation distribution for each link. Since all of the available data at the present time is sparse GPS probe data, only this case is analyzed here. Once travel time allocation has been performed, it is straight forward to apply a particle filter for performing real-time estimation.

Travel Time Allocation

An observation consists of a travel time over a path consisting of multiple (partial) links. In order to use the graphical model presented in section 5.3, the total travel time must be decomposed into a travel time for each (partial) link on the path. This can be achieved by maximizing the log-likelihood of the link travel times for each observation given the model parameters. This optimization problem for a single observation is

$$\underset{\mathbf{y}}{\mathbf{argmax}} \left\{ \sum_{l \in P} \ln(\sum_{s=1}^{S} z^s_l g_{l,s}(y_l)) \ : \sum_{l \in P} \alpha_{x^l_1,x^l_2} y_l = \tilde{y} \right\}, \tag{6.8}$$

where P is the set of links on the path, $\mathbf{y} := \{y_l\}_{l \in P}$ is a vector of the travel times assigned to each link on the path, x^l_1 and x^l_2 are the start and end location on link l, $\tilde{y}$ is the observed travel time between the GPS measurements, z^s_l is the probability of link l to be in state s, and $g_{l,s}$ is the travel time distribution for link l when in congestion state s. The values of x_1^l and x_2^l will be equal to the start and end of the link for all intermediate links and will only have non-trivial values for the first and last link of the path (where the actual GPS observations are). The values of z_l^s are obtained from the E-step of the EM algorithm, except in the first iteration where they have been initialized with reasonable values (see algorithm 7). The optimization problem in equation (6.8) has a number of variables equal to the number of links of the path between consecutive GPS measurements, which is always a relatively small number. This makes the optimization problem easy to solve using numerical methods.

To compute α_{x_1,x_2}, the proportion of the full link travel time to use, the method proposed in [55] is used (see the section on density estimation).

Particle Filtering

On small networks, it is possible to do exact inference in the CHMM by converting the model to an HMM (with a number of links-dimensional state vector) [85]. However, the transition matrix is a $S^N \times S^N$ matrix (N is the number of links in the network), which is intractable for traffic network with more than a few dozen links. Instead, an approximation based on particle filtering [48] is used. Each particle represents an instantiation of the time evolution of the network. Each particle has a weight proportional to the probability of having this instantiation of the state evolution of the network given the available data. It is necessary to simulate a high number of particles that evolve through the graphical model. These particles are used to estimate the probabilities of the state of each link and each time interval and the probabilities of transition between the state of the neighbors of link l at time $t-1$ and the state of link l at time t. See section 3.2.2 for more details on particle filtering.

6.3.2 Parameter Estimation: the EM Algorithm

As described earlier, the EM algorithm iterates between finding the most likely state of the network given the model parameters and then uses those state estimates to find the new most likely model parameters. Section 3.2.3 provides additional details about the EM algorithm. The details of how to apply the EM algorithm for the graphical model are presented here.

E step

As stated earlier, the E-step of the EM algorithm is identical to the state estimation section just discussed (section 6.3.1).

M step

For each link and each state, it is assumed that the travel time distribution $g_{l,s}$ is parameterized by a set of parameters $P_{l,s}$ and the set of all parameters is denoted $\mathbf{P} = (P_{l,s})_{l,s}$. To update these parameters, the expected complete log-likelihood is maximized given the probability $(z^s_{d,t,l})$ that each link l is in state s at time t and day d and the probability $(q^{s,r}_{d,t,l})$ of link l to be in state s given that the neighbors of link l are in state r at time $t-1$ and day d. The updates of the transition matrices A_l and the initial state probabilities π_l for each link of the network corresponds to optimizing on the set of parameters $\mathbf{A} = (A_l)_l$ and $\pi = (\pi_{l,s})_{l,s}$. The expected complete log likelihood is

$$\begin{aligned}
&\Lambda(Y|\,\mathbf{z}, \mathbf{q}, \mathbf{P}, \mathbf{A}, \pi) = \\
&\sum_{l=1}^{N}\sum_{s=1}^{S}\sum_{d=1}^{D}\sum_{t=1}^{T_d} z^s_{d,t,l}\left(\sum_{i=1}^{I_{d,t,l}} \ln(g_{l,s}(y_i))\right) + \\
&\sum_{l=1}^{N}\sum_{d=1}^{D}\sum_{t=2}^{T_d}\sum_{s=1}^{S}\sum_{r=1}^{S^{N_n^l}} q^{s,r}_{d,t,l}\ln(A_l(r,s)) + \\
&\sum_{l=1}^{N}\sum_{d=1}^{D}\sum_{s=1}^{S} z^s_{d,0,l}\ln(\pi_{l,s}),
\end{aligned} \tag{6.9}$$

where $I_{d,t,l}$ is the set of travel time observations for day d, time interval t, and link l as provided by the travel time allocation method presented in section 6.3.1.

The typical optimization problem (as described in section 3.2.3) is modified to take into account the varying number of observations for each link and each time interval. The optimization problem is stated as

$$\max_{\mathbf{P},\mathbf{A}} \quad \Lambda(Y|\,\mathbf{z}, \mathbf{q}, \mathbf{P}, \mathbf{A}, \pi)\ :\ \begin{cases} \sum_{s=1}^{S} A_l(r,s) = 1, \forall\, l, r \\ A_l(r,s) \in [0,\, 1], \forall l, r, s \\ \sum_{s=1}^{S} \pi_{l,s} = 1, \forall\, l \\ \pi_{l,s} \in [0,\, 1], \forall l, s \end{cases} \tag{6.10}$$

The updates of the transition probabilities A_l and of the initial state probabilities π_l are straightforward. The update of the travel time distributions depends on the type of distribution used in the model. Due to the travel time allocation, the optimization problem on all the parameters $\mathbf{P}$ of the network decouples in $S \times N$ smaller optimization problems, one for each state and link of the network. For state s and link l, the optimization problem is

$$\max_{p_{l,s}} \quad \sum_{d=1}^{D}\sum_{t=1}^{T_d} z^s_{d,t,l}\left(\sum_{i=1}^{I_{d,t,l}} \ln(g_{l,s}(y_i))\right), \tag{6.11}$$

where $p_{l,s}$ represents the parameters of the travel time distribution $g_{l,s}$. Decoupling the optimization problem 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 whole network resulting in a large non-linear optimization problem that does not scale well.

Algorithm 7 Estimation of the historical distribution of travel time and state transition probability matrices.

```
Initialize the parameters $P_{l,s}$ of the distributions, the state transition probability matrices $A_l$, the initial state probabilities $\pi_{l,s}$, and the state probabilities $z^s_{d,t,l}$

EM-algorithm with travel time allocation:
while The algorithm has not converged do
    Travel time allocation (section 6.3.1)
    $y_l \leftarrow$ Allocated travel times given the parameters $P_{l,s}$ and the state probabilities $z^s_{d,t,l}$

    E Step (section 6.3.1): compute the expected state probabilities $z^s_{d,t,l}$ and transition probabilities $q^{r,s}_{d,t,l}$ given $(y_l)_l$, $(P_{l,s})_{l,s}$ and $(A_l)_l$
    $z^s_{d,t,l} \leftarrow E(z^s_{d,t,l}|y_l,\, P_{l,s},\, A_l)$
    $q^{r,s}_{d,t,l} \leftarrow E(q^{r,s}_{d,t,l}|y_l,\, P_{l,s},\, A_l)$

    M Step (section 6.3.2): maximize the expected complete log-likelihood, given the state probabilities $z^s_{d,t,l}$ and the transition probabilities $q^{r,s}_{d,t,l}$.
    $(P_{l,s}, A_l, \pi_l) \leftarrow \mathbf{argmax}_{\mathbf{P},\mathbf{A},\pi}\Lambda(Y|\,\mathbf{z},\mathbf{q},\mathbf{P},\mathbf{A},\pi)$
end while
```

6.3.3 Results

The model was tested using probe data from a fleet of about 500 taxis in San Francisco as provided to us by the Cabspotting project [1], which has been integrated into *Mobile Millennium* system through a data feed. Each taxi provides a measurement of its location approximately once every minute (generally between 40 and 100 seconds), which falls into the category of sparse GPS probe data (as described in section 2.1.9). In addition to its location, the taxi also reports whether or not it is carrying a customer. This information allows for filtering out the points when a taxi is loading or unloading a passenger. This data is sent to the *Mobile Millennium* traffic system, where it is processed and visualized in real-time.

In the case study, data from November 25, 2009 through February 27, 2010 was used, focusing on weekdays from 3pm-8pm in the subnetwork of San Francisco depicted in figure 6.7. This subnetwork contains 322 links (where a link is defined as the road between two signals)

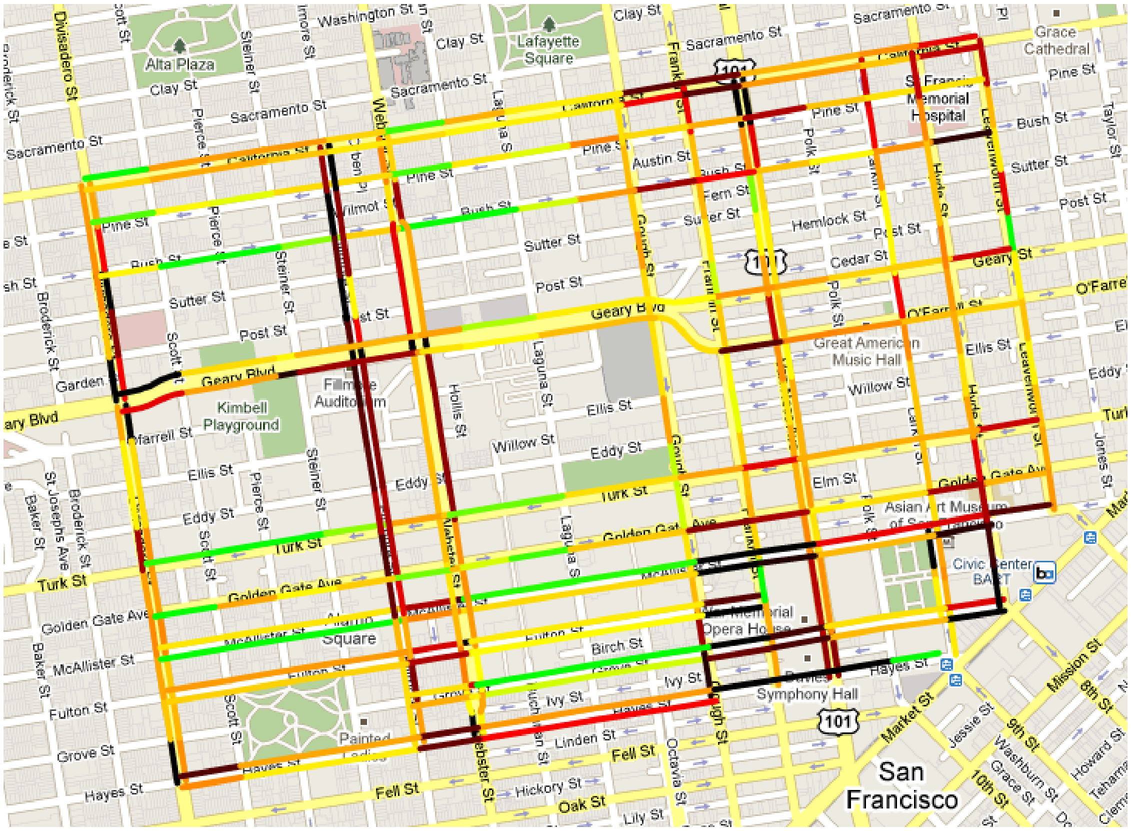


Figure 6.7: Real-time traffic estimation for a subnetwork of San Francisco. The color scale represents the estimated travel time divided by the speed limit travel time. Green is for values close to 1 (travel time is about the same as driving at the speed limit) and black indicates values around 5 (travel time is 5 times slower than driving at the speed limit).

and has an average of 600 observations per half hour time interval for the whole network. The time interval used was 30 minutes (half an hour) for the graphical model. These results are for the case where the observation probability distribution functions g (section 5.3.2) are independent Gaussians. In general, the choice of a Gaussian distribution restricts the flexibility of the model to capture unique traffic characteristics, but it is also far more tractable to solve in practice. While this section has described how to solve this model using the distribution introduced in section 4.3, no experimental results have been generated to date.

The approach requires a training period (section 6.3.2) before it can be used to make predictions in real-time. Data from November 25, 2009 through February 19, 2010 was used for the training period. Only Tuesdays, Wednesdays and Thursdays were used to train the model, which totalled 18 training days (after removing holidays and days with system malfunctions that prevented data collection). The model was then tested by running it over all Tuesdays, Wednesdays and Thursdays between February 20, 2010 and February 27, 2010,

Model	RMSE (sec)	MPE
Graphical model	46	30.1%
Baseline model	63	44.4%

Table 6.4: Experimental results comparison between the proposed graphical model and the baseline model.

which totalled 3 days.

The traffic density parameters (see the travel time allocation algorithm of section 6.3.1) for each hour of the day from 3pm to 8pm, where each hour period is assumed to have its own characteristics in terms of the average density on a link. The EM algorithm (section 6.3.2) was run over all the training data, with the assumption that the transition matrix **A** and the Gaussian distributions for each link are stationary over the study period. Once the parameters were learned through the EM algorithm, a particle filter was used to compute the most likely state of each link given real-time data on a test day. Figure 6.7 shows a map of the subnetwork of San Francisco with each link colored according to its level of congestion, defined as the mean travel time divided by a reference free flow travel time. The free flow travel time is computed as the travel time experienced when traveling at the speed limit and accounting for an expected delay (due to traffic signals) under light traffic conditions.

To quantify the validity of the estimates, the actual travel time of an observed path is compared to the estimate obtained by summing over the mean travel time for all links of the path. Table 6.4 shows the root mean squared error (RMSE) and mean percentage error (MPE) of our travel time predictions as compared to a baseline approach. The baseline approach computes the average speed for each observation and assigns it to each link along its path. Then all of the speeds recorded on each link are averaged to give a historical average speed for each link. The real-time version of this approach also computes the average speed on each link within the real-time estimation interval and then takes a weighted average between the historical and the real-time speed to give a speed estimate for each link of the network, which can be used to estimate travel times.

These results were computed on the data obtained between February 20 and February 27, 2010. The data was split into two sets, one for computing the real-time traffic estimates and one for computing the error metrics. This was done to ensure an unbiased comparison of the proposed graphical model and the baseline model. Approximately 70% of the data was used for computing the real-time traffic estimates with the other 30% used for computing the error metrics.

Chapter 7

Conclusions and Future Directions

This dissertation presented a novel approach to arterial traffic estimation relying only on GPS probe data and without any fixed-location sensor information. This work leverages the growth of the GPS-enabled smartphone industry as well as GPS data from other sources to provide real-time traffic information to drivers and transit agencies. Through the *Mobile Millennium* system, real-time traffic information can be sent to a number of targets including individual cell phones for route guidance while driving.

Several different traffic estimation models were presented and studied. The work began by looking at two different types of regression models (sections 5.1 and 6.1), one for classifying discrete states of traffic (logistic regression) and the other for estimating and forecasting a continuous representation of the traffic state (STARMA using link travel times as the state). The key challenges to address in the regression models were to process the raw data available into a format that the standard regression algorithms expect. This was accomplished through the development of custom aggregation functions (denoted $g_i(\cdot)$ and $h_i(\cdot)$ for the input representative quantity and the discrete classification quantity, respectively), which were constructed using traffic engineering knowledge combined with empirical data analysis. In summarizing the regression approaches, the most important lesson learned was that the quality of the aggregation functions was ultimately the determining factor in the success of the regression algorithms. This fact means that regression approaches could potentially be very successful in estimating traffic conditions, but that the combination of relatively high data requirements (the inability to handle missing data) and the need to build custom aggregation functions means that these models were not as robust as desired.

The second type of traffic estimation model presented was based on decomposing the traffic estimation process into a historical patterns step and a Bayesian update step for real-time estimation (sections 5.2 and 6.2). The historical model works by iterating between two basic operations, travel time allocation and parameter estimation. When using this model with certain classes of link travel time distributions, the travel time allocation problem is efficient even for large amounts of data (such distributions include the “traffic derived” distribution of section 4.3 as well as standard distributions like Gaussian, Log-Normal, Gamma and others). The parameter estimation problem is also efficient for the same set of distributions just listed. This decomposition scheme resulted in a huge performance improvement over possible direct approaches to solving a difficult nonlinear optimization problem. In summarizing this model, the travel time allocation algorithm was a key development in the progress made on arterial traffic estimation research. A naïve approach to travel time allocation is to distribute the travel time in proportion to the length of each link in the path, but this has proven to give inferior results. As a result of this development, this model was able to perform well on large-scale networks (in particular, the entire city of San Francisco was used for testing). Once the historic model parameters were learned, the real-time step of performing a Bayesian update was relatively straight forward. The key decisions to be made in this case were to empirically determine the proper parameters for how much weight to give the learned historic model parameters. To do this properly, a thorough set of experiments would need to be performed using cross validation, although these results were not processed as part of this dissertation. In conclusion, the combined historic/Bayesian update model had several advantages and disadvantages:

Advantages:

- The historic model leverages a vast amount of data collected over a long period of time. This feature enables the real-time model to have an accurate sense of what the typical traffic conditions are like at a particular day and time.

- The travel allocation component of the algorithm enables it to be efficient and capable of running on large networks.

- Leveraging large amounts of historical data provides a robust estimate of what the general distribution of traffic patterns are. This traffic pattern information is highly useful for businesses who need to plan routes for a large fleet of drivers (e.g. UPS or FedEx).

Disadvantages:

- The majority of the input data spans several links per observation. This degrades the quality of the information being provided to the algorithm. While travel time allocation performs well at splitting the data onto the links of the path, it is no substitute for directly measuring link travel times. When some parts of the network are sparsely covered by data (even when aggregating over all data collected), the model is unable to learn the correct travel time distribution.

- Decomposing the problem into the travel time allocation and parameter estimation components represents an approximation to the real optimization problem that is desired to be solved. The approximation scheme presented is not guaranteed to achieve a global optimum and there are currently no metrics for computing the optimality gap built into this algorithm.

The final traffic estimation model presented used a probabilistic graphical model as the starting point for modeling arterial traffic (sections 5.3 and 6.3). This model was a significant step forward in that it considered arterial traffic in a full spatio-temporal state evolution setting. It was a natural choice for a model, given that traffic conditions vary continuously over space and time. The continuous space-time domain of the problem is simplified into discrete spatial units (links) and discrete time units (time periods) that make sense from a traffic information systems perspective. The graphical model was solved using the expectation maximization algorithm for learning the model parameters and using a particle filter for performing real-time estimation. Just as in the previous model, this model breaks down into a historical and real-time components, but instead of using a simple time period by time period aggregation method for the historic and a Bayesian update for the real-time components, this model leverages the complete evolution of the state of traffic over many days to identify the transitions in traffic regimes as well as the link distribution parameters. The ability to learn the typical patterns of how traffic states evolve through the network was a key development in the success of this algorithm.

After studying and testing all of the models presented, the conclusion of this dissertation is that the model and algorithm presented in sections 5.3 and 6.3 provide a fundamental framework for using machine learning techniques for traffic estimation. The benefits of the model and algorithm are:

1. Combines the benefits of machine learning with physics of the road modeled using traffic theory.

2. Has the ability to learn real road network parameters.

3. Is independent of the choice of travel time distribution.

4. Provides efficient large-scale traffic monitoring using GPS data with the potential for including other types of data available.

These positive aspects of the model presented provide a sound foundation for continuing further research on this topic. The key areas of possible improvement are:

1. How can the current model make use of cloud computing and parallel computing technology that is increasingly becoming available?

2. How can the current model be used to identify places in the road network where transit agencies should focus? Questions could include how to determine where the worst bottlenecks regularly occur or where the most poorly timed traffic signals are.

3. The model requires that a parametric form for the travel time distributions be specified. How can the form of the distribution also be learned by the model? This work began to analyze this topic, but it can be further developed.

4. How can the optimal number of particles for the particle filter be determined?

This dissertation also contributed to the advancement of traffic information systems by designing and implementing the *Mobile Millennium* system. The structure of this system provides the foundation needed for any generic traffic information system. A number of research activities will directly use this system as the basis to perform further research on traffic estimation.

There were a number of important considerations that went into the design of the system. The primary design element of the system is the use of a database-centric architecture. This approach allowed the system to be modular with each module only needing to interface with a single core module (which handles all interactions with the database). The benefits of this design for the project were enormous. By having each module read input and write output to the database, it was easy to track and analyze the data flow through the system by being able to query any set of data at any step in its progress. The second key design element was to build a visualization system capable of providing quick access to every step of the data flow process from visualizing raw data on the map to the final outputs of the estimation models. Without the visualization component, the numerous bugs that arise in any complex code (which are inevitable given the complexity of the models) would have never been found. This same point was the motivation behind another key design element, which was the evaluation framework. This component of the system was built to allow for a standard set of comparison metrics for comparing a wide variety of data, including raw, filtered, and model output data. Having a set of benchmark metrics to quickly assess the performance of the system is crucial to successful execution.

The core module that represents the interface to the database has a number of important features that were critical to the successful operation of the system. The most important piece of the core module is a simplified representation of the road network that any other module can access quickly. Creating this simplified representation of the road network was perhaps the single most important development for the system due to the fact that every other module relies upon it. While it is relatively simple to describe the simplified representation of the network (see section 2.2.4), building it was a challenge due to the complexity of the raw road network data provided by Navteq. Sifting through the complexity to extract the necessary components was an important achievement. In addition to the network representation, it was necessary to build map matching functionality into the core module. Every data source, whether from fixed location sensors or mobile GPS devices, requires that it be placed on the map correctly for it to be used in a traffic estimation algorithm. The map matching component was built to be robust, scalable, and fast.

The most important lessons learned from developing the *Mobile Millennium* system were that the core functionality is absolutely essential to the success of the traffic estimation algorithms. Given that fact, it is important to invest heavily in terms of developer time, computer resources, and systems managers. A quick and easy solution to any of the problems of network representation or map matching lead to the estimation models providing poor results. Additionally, without the visualization, evaluation and monitoring capabilities provided by the system, the time to develop and implement a great traffic estimation algorithm goes up by an order of magnitude.

With these lessons in mind, the final conclusions of this dissertation are that the systems development and the estimation model development are two complimentary pieces that have to work together to provide accurate traffic information. It is the hope that the system will continue to be developed to provide even higher accuracy information to drivers and continue to add new features such as traffic forecast and route guidance.

Bibliography

[1] Cabspotting. http://www.cabspotting.org.

[2] California Center for Innovative Transportation. http://www.calccit.org.

[3] California Department of Transportation. http://www.dot.ca.gov/.

[4] CCIT/CalFrance Program. http://www.calccit.org/resources/CalFrance.html.

[5] CCTV information. http://www.cctv-information.co.uk/i/An_Introduction_to_ANPR.

[6] CITRIS, Center for Information Technology Research in the Interest of Society. http://www.citris-uc.org/.

[7] Finish Funding Agency for Technology and Innovation. http://www.tekes.fi/.

[8] Freeway Performance Measurement System. http://pems.eecs.berkeley.edu/Public/.

[9] NAVTEQ Inc. http://www.navteq.com.

[10] Next generation simulation. http://ngsim-community.org/.

[11] Nokia Inc. http://www.nokia.com.

[12] NSF, National Science Foundation. http://www.nsf.gov/.

[13] PostGIS extensions for PostgreSQL. http://postgis.refractions.net/.

[14] PostgreSQL database. http://www.postgresql.org/.

[15] Sensys Networks. http://www.sensysnetworks.com.

[16] UC Berkeley Center for Future Urban Transport. http://www.its.berkeley.edu/volvocenter/.

[17] UCTC, University of California Transportation Center//. http://www.uctc.net/.

[18] United States Department of Transportation. http://www.dot.gov.

[19] VTT. http://www.vtt.fi/.

[20] *Highway Capacity Manual.* TRB, National Research Council, Washington, D.C., 2000.

[21] Berkeley Transportation Systems (BTS), PeMS User Guide, Version 5.2. http://pems.eecs.berkeley.edu, 2004.

[22] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on signal processing*, 50(2), 2002.

[23] X. Ban, L. Chu, and H. Benouar. Bottleneck identification and calibration for corridor management planning. *Transportation Research Record*, 1999:40–53, 2007.

[24] X. Ban, R. Herring, P. Hao, and A. Bayen. Delay pattern estimation for signalized intersections using sampled travel times. In *Proceedings of the 88th Annual Meeting of the Transportation Research Board*, Washington, D.C., January 2009.

[25] X. Ban, R. Herring, J. Margulici, and A. Bayen. Optimal sensor placement for freeway travel time estimation. *Proceedings of the 18th International Symposium on Transportation and Traffic Theory*, July 2009.

[26] X. Ban, Y. Li, A. Skabardonis, and J. Margulici. Performance evaluation of travel time methods for real time traffic applications. In *Proceedings of the 11th World Congress on Transport Research (CD-ROM)*, 2007.

[27] B. Bartin, K. Ozbay, and C. Iyigun. A clustering based methodology for determining the optimal roadway configuration of detectors for travel time estimation. *Transportation Research Record*, 2000:98–105, 2007.

[28] R. Bellman and S. Dreyfus. *Applied Dynamic Programming.* Princeton University Press, 1962.

[29] P. Bickel, C. Chen, J. Kwon, J. Rice, E. Van Zwet, and P. Varaiya. Measuring traffic. *Statistical Science*, 22(4):581–597, 2007.

[30] M. Brand. Coupled hidden Markov models for modeling interacting processes. Technical report, The Media Lab, Massachusetts Institute of Technology, Boston, MA, 1997.

[31] E. Charniak. *Statistical Language Learning.* MIT Press, Cambridge, Massachusetts, 1993.

[32] C. Chen, J. Kwon, J. Rice, A. Skabardonis, and P. Varaiya. Detecting errors and imputing missing data for single-loop surveillance systems. *Journal of Transportation Research Board*, 1981:160–167, 2003.

[33] C. Chen, K. Petty, A. Skabardonis, and P. Varaiya. Freeway performance measurement system: mining loop detector data. In *80th Annual Meeting of the Transportation Research Board*, Washington, D.C., January 2001.

[34] L. Chu, X. Liu, and W. Recker. Using microscopic simulation to evaluate potential intelligent transportation system strategies under nonrecurrent congestion. *Transportation Research Record*, 1886:76–84, 2004.

[35] C. Claudel and A. Bayen. Lax-Hopf based incorporation of internal boundary conditions into Hamilton-Jacobi equation. Part I: theory. *IEEE Transactions on Automatic Control*, 55(5):1142–1157, 2010. doi:10.1109/TAC.2010.2041976.

[36] C. Claudel and A. Bayen. Lax-Hopf based incorporation of internal boundary conditions into Hamilton-Jacobi equation. Part II: Computational methods. *IEEE Transactions on Automatic Control*, 55(5):1158–1174, 2010. doi:10.1109/TAC.2010.2045439.

[37] C. Claudel, M. Nahoum, and A. Bayen. Minimal error certificates for detection of faulty sensors using convex optimization. In *Proceedings of the 47th Annual Allerton Conference on Communication, Control, and Computing*, Allerton, IL, Sep. 2009.

[38] B. Coifman. Estimating travel times and vehicles trajectories on freeways using dual loop detectors. *Transportation Research A*, 36(4):351–364, 2002.

[39] C. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research B*, 28(4):269–287, 1994.

[40] C. de Fabritiis, R. Ragona, and G. Valenti. Traffic estimation and prediction based on real time floating car data. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 197–203, 2008.

[41] J. Butler et. al. Mobile Millennium final report. Technical report, University of California, Berkeley, To appear in 2011.

[42] L. C. Evans. *Partial Differential Equations.* Graduate Studies in Mathematics, V. 19. American Mathematical Society, Providence, RI, 1998.

[43] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter.* Springer-Verlag, Berlin Heidelberg, 2007.

[44] C. Furtlehner, J. Lasgouttes, and A. de la Fortelle. A belief propagation approach to traffic prediction using probe vehicles. In *Proceedings of the IEEE 10th International Conference on Intelligent Transportation Systems*, pages 1022–1027, 2007.

[45] H. Gault and I. Taylor. The use of output from vehicle detectors to access delay in computer-controlled area traffic control systems. Technical Report Research Report No. 31, Transportation Operation Research Group, University of Newcastle upon Tyne, United Kingdom, 1977.

[46] N. Geroliminis and A. Skabardonis. Prediction of arrival profiles and queue lengths along signalized arterials by using a Markov decision process. *Transportation Research Record*, 1934(1):116–124, May 2006.

[47] J. F. Gimpel. A theory of discrete patterns and their implementation in snobol4. *Commun. ACM*, 16:91–100, February 1973.

[48] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F Radar and Signal Processing*, 140(2):107–113, 1993.

[49] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, New York, NY, corrected edition, July 2003.

[50] B. Hellinga, P. Izadpanah, H. Takada, and L. Fu. Decomposing travel times measured by probe-based traffic monitoring systems to individual road segments. *Transportation Research Part C: Emerging Technologies*, 16(6):768 – 782, 2008.

[51] J. Herrera and A. Bayen. Traffic flow reconstruction using mobile sensors and loop detector data. In *Proceedings of the 87th Transportation Research Board Annual Meeting*, Washington, D.C., January 2008.

[52] 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. *Transportation Research Part C: Emerging Technologies*, 18(4):568–583, August 2010.

[53] R. Herring, A. Hofleitner, P. Abbeel, and A. Bayen. Estimating arterial traffic conditions using sparse probe data. In *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems*, Madeira, Portugal, September 2010.

[54] R. Herring, A. Hofleitner, S. Amin, T. Abou Nasr, A. Abdel Khalek, P. Abbeel, and A. Bayen. Using mobile phones to forecast arterial traffic through statistical learning. In *Proceedings of the 89th Annual Meeting of the Transportation Research Board*, Washington D.C., 2010.

[55] A. Hofleitner, R. Herring, and A. Bayen. A hydrodynamic theory based statistical model of arterial traffic. *Technical Report UC Berkeley*, August 2010.

[56] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J. Herrera, and A. Bayen. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *The Sixth Annual International conference on Mobile Systems, Applications and Services (MobiSys 2008)*, Breckenridge, U.S.A., June 2008.

[57] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Enhancing security and privacy in traffic-monitoring systems. *IEEE Pervasive Computing*, 5(4):38–46, March 2006.

[58] The *Mobile Millennium* Project. http://traffic.berkeley.edu.

[59] T. Hunter, R. Herring, A. Hofleitner, A. Bayen, and P. Abbeel. Trajectory reconstruction of noisy GPS probe vehicles in arterial traffic. *In progress*, 2010.

[60] F.V. Jensen and T.D. Nielsen. *Bayesian Networks and Decision Graphs*. Springer, New York, NY, 2nd edition, June 2007.

[61] Z. Jia, C. Chen, B. Coifman, and P. Varaiya. The PeMS algorithms for accurate, real-time estimates of g-factors and speeds from single-loop detectors. In *4th IEEE Conference on Intelligent Transportation Systems*, Oakland, CA, August 2001.

[62] 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 *Proceeding of IEEE ITS Annual Meeting*, pages 536–541, 2001.

[63] Y. Kamarianakis and P. Prastacos. Space-time modeling of traffic flow. *ERSA Conference Papers*, June 2006.

[64] A. Krause, E. Horvitz, A. Kansal, and F. Zhao. Toward community sensing. In *Proceedings of ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, St. Louis, MO, April 2008.

[65] J. Krumm. Inference attacks on location tracks. In *Proceedings of the Fifth International Conference on Pervasive Computing*, Toronto, Ontario, Canada, May 2007.

[66] K. Kwong, R. Kavaler, R. Rajagopal, and P. Varaiya. Arterial travel time estimation based on vehicle re-identification using wireless magnetic sensors. *Transportation Research Part C: Emerging Technologies*, 17(6):586–606, December 2009.

[67] Lawrence L. Larmore and Baruch Schieber. On-line dynamic programming with applications to the prediction of RNA secondary structure. In *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pages 503–512, San Francisco, California, United States, 1990. Society for Industrial and Applied Mathematics.

[68] M. Lighthill and G. Whitham. On kinematic waves. II. a theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 229(1178):317–345, May 1955.

[69] H. Liu, A. Danczyk, R. Brewer, and R. Starr. Evaluation of cell phone traffic data in minnesota. *Transportation Research Record*, 2086:1–7, December 2008.

[70] H. Liu and W. Ma. A virtual probe approach for time-dependent arterial travel time estimation. *Presented at the 87th Annual Conference on Transportation Research Board, and Submitted for publication*, 2008.

[71] H. Liu, H. van Zuylen, H. van Lint, and M. Salomons. Predicting urban arterial travel time with State-Space neural networks and kalman filters. *Transportation Research Record*, (1968):99–108, 2006.

[72] L. Mimbela, L. Klein, P. Kent, J. Hamrick, K. Luces, and S. Herrera. *Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems*. Federal Highway Administration's (FHWA) Intelligent Transportation Systems Program Office, August 2007.

[73] 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 *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, pages 1–6, 2009.

[74] G.F. Newell. *Theory of Highway Traffic Signals*. Institute of Transportation Studies, University of California, Berkeley, CA, 1988.

[75] A. Ng. Lecture notes. CS 229: Machine learning. *Stanford University*, 2003.

[76] C. Oh. *Anonymous Vehicle Tracking for Real-Time Traffic Performance Measures*. PhD thesis, University of California, Irvine, Irvine, CA, 2003.

[77] C. Oh and S. Ritchie. Anonymous vehicle tracking for real-time traffic surveillance and performance on signalized arterials. In *Proceedings of the 82nd Annual Meeting of the Transportation Research Board (CD-ROM)*, 2003.

[78] T. Park and S. Lee. A Bayesian approach for estimating link travel time on urban arterial road network. In *Computational Science and Its Applications ICCSA 2004*, pages 1017–1025. Perugia, Italy, May 2004.

[79] P. Pfeifer and S. Deutsch. A three-stage iterative procedure for space-time modeling. *Technometrics*, 22(1):35–47, February 1980.

[80] L. Rabiner. A tutorial on hidden markov models and selected applications inspeech recognition. *Proceedings of the IEEE*, 77(2):257286, 1989.

[81] J. Rice and E. Zwet. A simple and effective method for predicting travel times on freeways. *IEEE Transactions on Intelligent Transportation Systems*, 5(3):200–207, 2004.

[82] P. Richards. Shock waves on the highway. *Operations Research*, 4(1):42–51, February 1956.

[83] S. Ritchie, S. Park, S. Jeng, and A. Tok. Anonymous vehicle tracking for real-time freeway and arterial street performance measurement. Technical Report Research Report, UCB-ITS-PRR-2005-9, California PATH, 2005.

[84] C. Robert. *The Bayesian choice: a decision-theoretic motivation.* Springer-Verlag, 1994.

[85] S. Russell and P. Norvig. *Artificial Intelligence - A Modern Approach.* Prentice-Hall, Inc, Englewood Cliffs, NJ, 1995.

[86] V. Sisiopiku and N. Rouphail. Travel time estimation from loop detector data for advanced traveler information system applications. Technical report, Illinois University Transportation Research Consortium, 1994.

[87] A. Skabardonis and R. Dowling. Improved speed-flow relationship for planning applications. *Transportation Research Record: Journal of the Transportation Research Board*, 1572:18–23, 1997.

[88] A. Skabardonis and N. Geroliminis. Real-time estimation of travel times on signalized arterials. In *Proceedings of the 16th International Symposium on Transportation and Traffic Theory*, University of Maryland, College Park, MD, July 2005.

[89] A. Skabardonis and N. Geroliminis. Real-time estimation of travel times on signalized arterials. In *16th International Symposium on Transportation and Traffic Theory*, pages 387–406. College Park, MD, 2005.

[90] A. Skabardonis and N. Geroliminis. Real-time monitoring and control on signalized arterials. *Journal of Intelligent Transportation Systems*, 12(2):64–74, March 2008.

[91] X. Sun, L. Munoz, and R. Horowitz. Mixture Kalman filter based highway congestion mode and vehicle density estimator and its application. In *Proceedings of the 2004 American Control Conference*, pages 2098–2103, Boston, MA, 2004.

[92] 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 ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Berkeley, CA, November 2009.

[93] TTI. Texas Transportation Institute: Urban Mobility Information: 2007 Annual Urban Mobility Report. http://mobility.tamu.edu/ums/, 2007.

[94] M. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. Technical report, Dept. of Statistics, September 2003. Published: Technical Report 649.

[95] P. D. Wasserman. *Neural computing: theory and practice*. Van Nostrand Reinhold Co., New York, NY, USA, 1989.

[96] J. Wasson, J. Sturdevant, and D. Bullock. Real-time travel time estimates using mac address matching. *Institute of Transportation Engineers Journal*, 78(6):20–23, 2008.

[97] Laurence A. Wolsey and George L. Nemhauser. *Integer and Combinatorial Optimization*. Wiley-Interscience, New York, NY, 1 edition, November 1999.

[98] D. Work, S. Blandin, O. Tossavainen, B. Piccoli, and A. Bayen. A distributed highway velocity model for traffic state reconstruction. *Applied Research Mathematics eXpress (ARMX)*, 1:1–35, April 2010.

[99] D. Work, O. Tossavainen, S. Blandin, A. Bayen, T. Iwuchukwu, and K. Tracton. An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 5062–5068, Cancun, Mexico, December 2008.

[100] X. Xie, R. Cheu, and D. Lee. Calibration-free arterial link speed estimation model using loop data. *Journal of Transportation Engineering*, 127(6):507–514, 2001.

[101] H. Xiong and G. Davis. Travel time estimation on arterials. In *Proceedings of the 87th Annual Meetings of Transportation Research Board (CD-ROM)*, 2008.

[102] H. Zhang. A link journey speed model for arterial traffic. *Transportation Research Record*, 1676:109–115, 1998.

Appendix A

Optimal Sensor Placement for Highways

Intelligent Transportation Systems (ITS) applications rely on various types of data to characterize traffic such as flow or speed. The data is usually collected from fixed-location traffic sensors. For example, freeway travel time estimation often requires speeds measured at specific locations. Traditionally, a large portion of traffic sensors were deployed on a case by case basis by practitioners without a systematic study of the quantity and locations of sensors needed. Since traffic sensors are limited resources with high installation, operation, and maintenance costs, determining optimal placement strategies maximizes the value of this resource. [25]

This chapter studies the dependency of travel time estimates on the locations of the sensors (specifically fixed-location sensors). The goal is to minimize the error in our estimate of vehicle travel times. The travel time application is selected because travel time estimates are one of the most useful roadway traffic metrics for both traffic management agencies and the driving public. First, travel time is a crucial and direct measure of traffic conditions and system performance. Second, travel times represent information that is easy to understand and process. Furthermore, relevant traffic information enables travelers to make educated choices about their itinerary, departure time or even transportation mode, which may result in a form of "system self-management." This work was performed at the California Center for Innovative Transportation [2].

A.1 Problem Statement

The problem studied in this chapter can be formally stated as follows: given a freeway segment (called *route* r) and a given number of fixed-location sensors (such as loop detectors), where should these sensors be placed so that their deployment is "optimal" in terms of providing travel time estimates? Here we assume that the number of sensors is given (denoted as K), which may often be determined by budget constraints. If this is not the case, one can always solve the problem for different numbers of sensors and pick the one with the desired performance. The efficiency of our proposed algorithm makes solving the problem multiple times (with different values of K) tractable.

Similar to other engineering problems, the answer to this optimal sensor placement problem depends on several factors. First, there are numerous methods available to compute travel times and sensors can usually provide multiple types of data (such as aggregated and disaggregated speeds, flow, occupancy, etc). Therefore, determining optimal sensor placement is dependent upon the travel time estimation method and the sensor data type. This section discusses assumptions made to address these concerns, most of which are consistent with what is currently used in practice.

A.2 Travel Time Estimation Methods

To be consistent with current practice, we assume that travel times are calculated based on aggregated sensor speeds (say every $\Delta T = 30$ seconds). Speeds can be obtained directly from double loop detectors and other types of fixed location sensors or estimated from single loop detectors [62]. We further assume that every sensor has a spatial "influence area", called a *link*. The sensor speed represents the (uniform) speed of the entire link associated with the sensor. There are a number of ways to define how a sensor is associated with its link (for example, PeMS [8] defines a link as the segment between the middle points of two sensors. [21] We assume that a sensor is always in the middle of its corresponding link [1]. Different link definitions lead to slightly different ways of interpreting sensor speeds, which in turn might result in small variations in travel time calculation. These variations however should not be significant.

Following this convention, the to-be-deployed K sensors divide the study route r into K links, and the route travel time is the summation of all link travel times. We recognize that such a definition will effectively eliminate certain travel time estimation methods based directly on routes (e.g., [81]). However, it is widely used in practice (see for example [21], pp. 3-23). More importantly, the DP model presented does not depend on how link travel times are calculated. This implies much flexibility regarding which travel time method to use in the model.

We focus on two specific travel time computation methods: the *instantaneous* method [26] and *Coifman* method [38]. The instantaneous method assumes traffic conditions remain

[1] One may argue that restricting sensors to be only in the middle of its link can potentially filter out better solutions. This issue was considered previously by some researchers. For example, a probability distribution was assumed in [27], which describes the probability that a sensor will be deployed to each discretized section of a link . However, in practice, after sensor are deployed based on the results from specific optimization models, practitioners need a straightforward way to define the link associated with each sensor to compute travel times. If sensors are allowed to be deployed at any arbitrary location within a link, the link boundary will have to be recorded to compute travel times. We argue that this is highly impractical in reality.

unchanged from the time a vehicle enters a route until it leaves the route. Therefore, travel time of the route can be computed by summing the travel times of the constituent links at the time a vehicle enters the route. This method is "naive" in the sense that traffic condition changes are not considered at all; however, it is probably the mostly widely used method in practice due to its simplicity and the fact that it can be used in real time (i.e., no future information or prediction is required). The second method, originally developed in [38], is a more sophisticated algorithm for calculating link travel times. The method constructs vehicle trajectories from sensor speeds using traffic flow theory, from which link travel times can be extrapolated. The reader is referred to [38] for detailed discussions of how the algorithm works.

We use the instantaneous method to illustrate the DP model and the solution algorithm. However, we discuss how the Coifman method can also be considered in the model and solution algorithm. In fact, our algorithm can be used with any travel time calculation method that estimates link travel times using one sensor per link and then adds them up the link travel times to compute the route travel time. Our algorithm will not work for travel time estimation methods that do not have this separability property.

A.3 Quantifying Travel Time Estimation Quality

The quality of a single travel time estimate could be calculated in a number of different ways. One could look at the difference between the estimate and the actual median travel time realized among all vehicles traversing the route during some given time window. One could also calculate the mean squared error of vehicle travel times with respect to the estimate for each vehicle's travel time. Here we define the objective function that is optimized in our DP model. For this purpose, we assume that we have trajectories of a certain number of vehicles (assumed to be M). We first denote $\hat{\tau}_k^m$ and τ_k^m the estimated and actual travel times of the m-th vehicle ($1 \leq m \leq M$) traveling link k ($1 \leq k \leq K$), respectively. The travel time estimation error for the m-th vehicle on link k, denoted as e_k^m, can be expressed as:

$$e_k^m = \hat{\tau}_k^m - \tau_k^m. \tag{A.1}$$

We use the same objective function as that in [27], which is defined as follows:

$$\hat{E} = \frac{\sum_{m=1}^{M} \sum_{k=1}^{K} (e_k^m)^2}{M} = \sum_{k=1}^{K} \hat{E}_k. \tag{A.2}$$

Here $\hat{E}$ represents the objective function. $\hat{E}_k$ is the *Mean Squared Error* (MSE) of the travel time estimation for all M vehicles for link k, defined as:

$$\hat{E}_k = \frac{\sum_{m=1}^{M}(e_k^m)^2}{M}. \tag{A.3}$$

The objective defined in (A.2) focuses on estimation errors of all individual links, instead of only on the entire route. The reason for this is that we want to generate sensor locations that can provide "good" estimates of all link travel times, not only in terms of the entire route. If attention is only put on the entire route, it is possible that the resulting sensor locations may underestimate travel times for certain links and overestimate for other links, but as a whole, they cancel out each other and provide good estimation. This type of sensor placement is not desirable. It is easy to see that the objective function we use here can effectively eliminate such sensor deployment strategies since they will lead to large objective values using equation (A.2).

A.4 Dynamic Programming Formulation

An overview of our solution method to the sensor placement problem is presented in figure A.1. The objective is to minimize the summation of link MSEs, so we investigate the MSE of any link k. For this purpose, we apply a scheme to discretize both time and space. We first divide the given route r into small segments, called *sections*. The premise is that if the length of a section is sufficiently small, we can reasonably assume that speed does not change within the section and it does not matter where to place a sensor within the given section. We thus only need to determine where to deploy the given K sensors to these small sections. Assume that the length of each section is Δx and that the given route r can be divided into N sections. We use $n = 1, \ldots, N$ to index a given section. A link then contains one or more sections, and the link boundaries are at the section boundaries. Also, since we assume a sensor is always in the middle of its link, the sensor deployment problem is now converted to determine the optimal starting and ending indices of all the K links that comprise the study route. In the time domain, it is natural to divide (evenly) the time into *intervals* with the interval length $\Delta T = 30$ seconds. This is because we assume that sensors can only provide 30-sec average speeds. In particular, assume the entire study period can be divided into H time intervals and $h = 1, \ldots, H$ is used to index a given interval. For simplicity, we assume route r starts with $x = 0$ and time starts with $t = 0$.

This space and time discretization is illustrated in Figure A.2. It is clear from the figure that the two-dimensional $x - t$ space is divided into a grid of *sensor boxes*. Each sensor box represents a data collection unit (speeds in this article) at a specific location (section), which is only active for the designated time period (30-sec long). The average speed of each sensor box can be computed via available vehicle trajectories. In particular, it is defined as the average speed of all vehicles that pass the sensor at the designated time period. This mimics the way loop detectors collect average speeds in practice. Calculating the average speed for any sensor box (n, h) with $n = 1, \ldots, N, h = 1, \ldots, H$ of the route will result in the *speed*

Algorithm for determining optimal sensor placement based on known vehicle trajectories

Input/Data:

1. Trajectories of Individual Vehicles
2. Desired number of sensors to place (denoted K)
3. Segment Length (i.e. what granularity should the roadway be discretized?)
4. Travel Time estimation method (e.g. instantaneous or Coifman)
5. (optional) Minimum spacing between sensors
6. (optional) Locations of any existing sensors

Desired Output:

1. Optimal locations for K sensors as defined by objective A.2
2. Estimated travel time for each vehicle given the optimal sensor locations and associated sensor data (where sensor data is calculated based on the vehicle trajectories)

Figure A.1: Optimal Sensor Placement Algorithm.

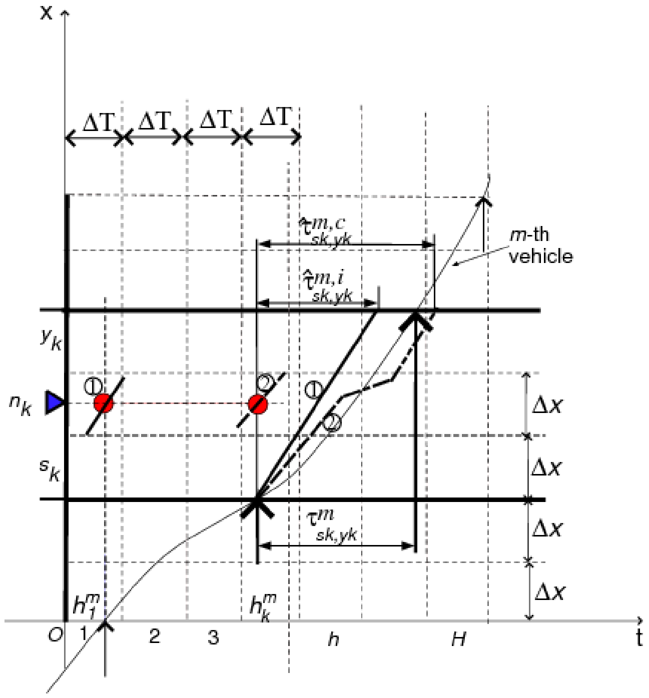


Figure A.2: Actual and Estimated Travel Times.

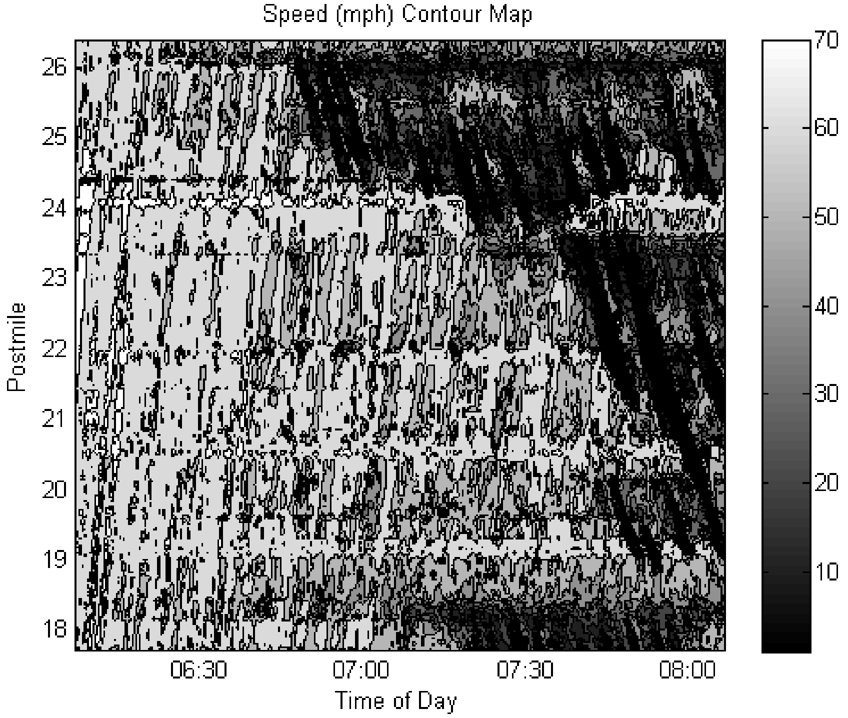


Figure A.3: Speed Contour Map.

field (also called *speed contour map*, see [23]) of the study route for the study period. Figure A.3 depicts the speed field of the micro-simulation data.

Notice that if all vehicle trajectories are available and can cover all sensor boxes, the speed field can be calculated; otherwise, we may have “blank” sensor boxes for which there is no vehicle passing by. For blank sensor boxes, we estimate the corresponding average speeds using surrounding sensor boxes, which is called *imputation* [32]. In this article, we adopt a simple imputation method: the speed of a blank sensor box is the average speed of all its surrounding sensor boxes whose speeds are already available.

We assume that the speed field is given by the above discretization scheme. Furthermore, assume the k-th link starts at section s_k and ends at section $y_k \geq s_k$. Both s_k and y_k are integers to represent a section. Note that the starting and ending sections are both inclusive, i.e., link k starts at the starting location of section s_k and ends at the ending location of section y_k. This is illustrated in Figure A.2.

To calculate the MSE of link k as expressed in equation (A.3), we focus on the given M vehicles. For any m-th vehicle, Figure A.2 depicts, in a solid thin line, the trajectory of the vehicle. In the figure, we denote $\tau^m_{s_k,y_k}$ the actual travel time of the vehicle traversing link k (i.e. from the starting location of section s_k to the ending location of section y_k). The corresponding estimated travel times are denoted as $\hat{\tau}^{m,i}_{s_k,y_k}$ (instantaneous) and $\hat{\tau}^{m,c}_{s_k,y_k}$

(Coifman). It can be seen that $\tau^m_{s_k,y_k}$ can be expressed as:

$$\tau^m_{s_k,y_k} = t^m_{y_k\Delta x} - t^m_{(s_k-1)\Delta x}, \tag{A.4}$$

where t^m_x denotes the time when the m-th vehicle passes location x. Suppose a sensor is deployed on this k-th link. Based on our assumptions, the sensor will be in the middle of the link. Denote n_k the section that the sensor on link k is located, we have:

$$n_k = \lfloor (s_k + y_k)/2 \rfloor. \tag{A.5}$$

Here $\lfloor \cdot \rfloor$ denotes the *rounding* operator. Assume the m-th vehicle enters route r at time interval h^m_1 and it enters section s_k, the starting section of link k, at time interval h^m_k. Then according to the definitions of the instantaneous travel time, the average speed of the sensor box (n_k, h^m_1), denoted as v_{n_k,h^m_1}, will be used for computing the instantaneous travel time of link k. This is shown as the solid bold line in Figure A.2, which is marked as "1". Noticing that $(y_k - s_k + 1)\Delta x$ is the length of link k, we can compute the instantaneous travel time as follows:

$$\hat{\tau}^{m,i}_{s_k,y_k} = \frac{(y_k - s_k + 1)\Delta x}{v_{n_k,h^m_1}}, \tag{A.6}$$

If Coifman method is used instead for the link travel time, the vehicle trajectory will be first estimated by a piece-wise linear curve using traffic flow theory [38]. This is shown as the bold dash line in Figure A.2 (marked as "2"). The Coifman link travel time for link k, $\hat{\tau}^{m,c}_{s_k,y_k}$, does not have a close-form expression. However, it is clear that $\hat{\tau}^{m,c}_{s_k,y_k}$ only depends on the starting and ending sections of link k provided speeds of all sensor boxes are given, and the entrance time is assumed to be $t^m_{(s_k-1)\Delta x}$.

Denote $\hat{E}^i_k$ and $\hat{E}^c_k$ the MSE of travel time estimation for link k for instantaneous and Coifman travel times, respectively. Following equation (A.3), they are both functions of (s_k, y_k) and can be expressed as:

$$\hat{E}^i_k(s_k, y_k) = \frac{\sum_{m=1}^M \left(\hat{\tau}^{m,i}_{s_k,y_k} - \tau^m_{s_k,y_k}\right)^2}{M} = \frac{\sum_{m=1}^M \left(\frac{(y_k-s_k+1)\Delta x}{v_{n_k,h^m_1}} - t^m_{y_k\Delta x} + t^m_{(s_k-1)\Delta x}\right)^2}{M}, \tag{A.7}$$

$$\hat{E}^c_k(s_k, y_k) = \frac{\sum_{m=1}^M \left(\hat{\tau}^{m,c}_{s_k,y_k} - \tau^m_{s_k,y_k}\right)^2}{M}. \tag{A.8}$$

The above procedures for calculating link MSE (instantaneous or Coifman) show that link MSE only depends on the starting and ending sections of the link, i.e., s_k and y_k. In particular, the calculation is independent of how the $(k-1)$ sensors for the previous $(k-1)$ links are deployed once s_k and y_k are known. This motivates us to formulate the optimal sensor placement problem using dynamic programming, as will be shown in the next section.

Denote $\hat{E}^i$ and $\hat{E}^c$ the objective functions for instantaneous and Coifman travel times respectively. We will have, according to (A.2):

$$\hat{E}^i = \sum_{k=1}^{K} \hat{E}^i_k(s_k, y_k), \tag{A.9}$$

$$\hat{E}^c = \sum_{k=1}^{K} \hat{E}^c_k(s_k, y_k). \tag{A.10}$$

We can see that the objective functions for instantaneous and Coifman travel times are algorithmically similar, and the only difference is which link MSE to use. Therefore, in the remainder of this section, we only use instantaneous travel time to illustrate the proposed DP model.

Given the objective function, the optimal sensor location problem can be stated as follows: find the optimal values of $(s_k, y_k), k = 1, \dots, K$ such that (A.9) can be minimized. That is, one needs to solve the following optimization problem:

$$\min_{1 \le s_k, y_k \le N, k=1,\dots,K} \quad \sum_{k=1}^{K} \hat{E}^i_k(s_k, y_k). \tag{A.11}$$

Subject to constraints (A.12) - (A.15) below.

The above optimization model is a linear integer program since $\hat{E}^i_k(s_k, y_k)$ is computable for any given (k, s_k, y_k), and (s_k, y_k) are integer-valued. However, directly solving the model may not be tractable if the dimension of the problem is large.

We thus divide the problem into stages: at each stage, the optimal location of one sensor is obtained, which can be achieved by finding the optimal starting and ending locations of its associated link. For this purpose, we assign the starting location (section) of link k (i.e., s_k) as the state variable. Accordingly, the ending location of link k (i.e., y_k) is the decision variable. Once s_k and y_k are given, the sensor location (section) can be achieved through equation (A.5).

We first look at the constraints for s_k and y_k. Clearly, we have

$$s_1 = 1, \tag{A.12}$$

$$y_K = N. \tag{A.13}$$

This means that the first link must start at section 1 and the last link (link K) must end at section N. Also, we have the state transfer function as

$$s_{k+1} = y_k + 1. \tag{A.14}$$

That is, knowing the ending section of link k (y_k), the starting section of link $(k+1)$ must be the next section (y_k+1).

Furthermore, since every link contains at least one section, we have

$$k \leq s_k \leq y_k \leq N - K + k. \tag{A.15}$$

The first inequality holds since there are $k-1$ links before link k, which contain at least $k-1$ sections. Similarly, the last inequality holds since there are $K-k$ links after link k, which contain at least $K-k$ sections. Furthermore, equations (A.12) - (A.15) show that there is only one possible state for stage 1 as $s_1 = 1$, but multiple states for stage $k \geq 2$. In particular, equation (A.15) means that the possible states for any stage $k \geq 2$ is from k to $N-K+k$, i.e., the total number of states is $N-K+1$.

At any stage k, the cost of deploying a sensor is assumed to be the link MSE $\hat{E}_k^i$, which is consistent with the objective function (A.9) and (A.2). Since $\hat{E}_k^i$ is only a function of (s_k, y_k), the optimal value of y_k can be obtained by minimizing $\hat{E}_k^i$ if s_k is known. In particular, if we denote $F_k(s_k)$ as the total cost from stage k (including stage k) to the last stage (i.e. stage K), a recursive formulation for $F_k(s_k)$ can be given as:

$$F_1(s_1) = F_1(1) = \min\nolimits_{1 \leq y_1 \leq N-K+1} \left\{ \hat{E}_1^i(1, y_1) + F_2(y_1+1) \right\}, \tag{A.16}$$

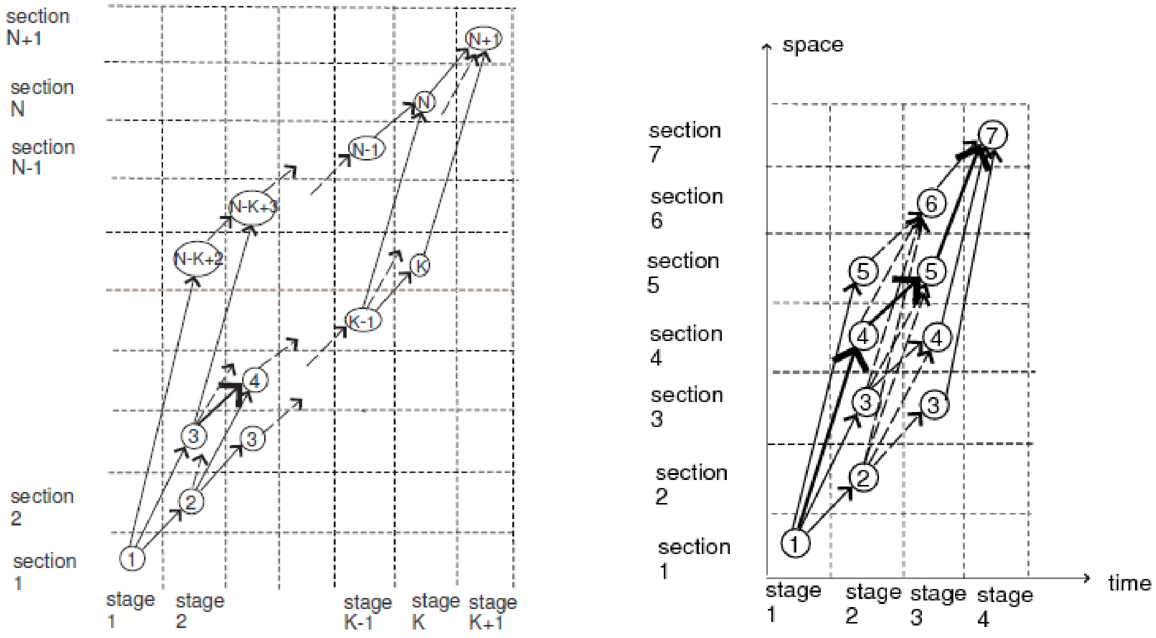
$$F_k(s_k) = \min\nolimits_{s_k \leq y_k \leq N-K+k} \left\{ \hat{E}_k^i(s_k, y_k) + F_{k+1}(y_k+1) \right\}, 2 \leq k \leq K-1, \tag{A.17}$$

$$F_K(s_K) = \hat{E}_K^i(s_K, N). \tag{A.18}$$

The above three equations are for stage 1, stage $k \in \{2, \ldots, K-1\}$, and stage K respectively. First, due to (A.12), we have $F_1(s_1) = F_1(1)$ for stage 1, which is a summation of the cost of stage 1 (i.e. $\hat{E}_1^i(1, y_1)$) and that from stage 2 to stage K (i.e. F_2). For stage $2 \leq k \leq K-1$, the cost F_k is a function of the state variable s_k, which is also the summation of the cost of the current stage k and that from stage $k+1$ to the last stage. Note that in both equations, the starting location of the next stage (i.e. stage 2 or $k+1$) is the *immediate next* section of the ending location of current stage (i.e. y_1+1 and y_k+1 respectively) due to (A.14). For the last stage, since the ending location must be N, $F_K(s_K)$ is automatically computable given s_K.

We can easily observe that (i) all constraints (A.12) - (A.15) are satisfied in the above three equations and there are no extra constraints introduced, (ii) $F_1(1) = \sum_{k=1}^{K} \hat{E}_k^i(s_k, y_k)$. Therefore, solving (A.11) is equivalent to solve (A.16) - (A.18). Furthermore, from these recursive equations, we can see that if $(s_k^*, y_k^*), 1 \leq k \leq K$ is an optimal solution, $(s_k^*, y_k^*), k_1 \leq k \leq k_2$ must be an optimal solution from stage $k_1 \geq 1$ to stage $k_2 \leq K$[2]. This illustrates

[2] Otherwise, suppose $(s_k', y_k'), k_1 \leq k \leq k_2$ is the optimal solution instead for stage $k_1 \geq 1$ to $k_2 \leq K$. Then it is clear that $(\bar{s}_k, \bar{y}_k)$, for $\bar{s}_k = s_k^*, 1 \leq k \leq k_1 - 1$ or $k_2 + 1 \leq k \leq K$, $\bar{s}_k = s_k', k_1 \leq k \leq k_2$; $\bar{y}_k = y_k^*, 1 \leq k \leq k_1 - 1$ or $k_2 + 1 \leq k \leq K, \bar{y}_k = y_k', k_1 \leq k \leq k_2$ will produce smaller objective than $(s_k^*, y_k^*), 1 \leq k \leq K$. This is a contradiction.



(a) Graph Representation of DP Model.

(b) An Illustrative Example.

that the *optimality principle* [28] holds for the model (A.16) - (A.18). Therefore, the model is a Dynamic Programming (DP) model.

The above DP model is for both the instantaneous and Coifman travel times due to the calculations of their link MSEs as described at the beginning of this section. In fact, it is easy to see that the proposed DP model can be used for any other link travel time method as long as the method only depends on the starting and ending locations of the link.

A graph representation of the DP model is depicted in figure A.4(a). In the figure, stages are listed horizontally and sections are listed vertically. Since we deploy one sensor per stage, we associate each link with a stage as well. Based on the above DP model, the state of a stage represents the starting section of the link associated with the stage. In this figure, all possible states of a stage are represented as *nodes*. In other words, a node represents a section of the roadway, and the node number is the section number. For example, the node at stage 2 and Section 2 represents that the starting location of link 2 could be section 2. As mentioned before (especially equations (A.12) - (A.15)), there is only one state in stage 1 ($s_1 = 1$) and $(N - K + 1)$ states (from k to $N - K + k$) for stage $k = 2, \ldots, K$. We further create a fake stage as stage $K + 1$ that has only one fake state $N + 1$.

A connection, denoted as an *arc*, may be created from a node in stage k to another node in the immediate next stage $k + 1$ if the latter node has a higher node number. Each arc actually represents a possible roadway link by defining the link's starting and ending sections. That is, an arc from node s_k in stage k to node s_{k+1} in stage $k + 1$ represents one possible configuration for link k: it starts at section s_k and ends at section $s_{k+1} - 1$ because the next link starts at s_{k+1}. Therefore, we must have $s_{k+1} > s_k$ in order to construct the arc. For example, the arc from node 2 in stage 2 to node 4 in stage 3 (marked in bold line) in Figure A.4(a) means that one possible configuration for link 2: it starts at node 2 and ends at node 3 (both are inclusive). Therefore, there should be no arc from node 4 in stage 2 to node 4 or lower in stage 3. Furthermore, there are no arcs between any two stages that are not adjacent to each other. We also associate a cost with each arc in Figure A.4(a). For the arc from node s_k in stage k to node s_{k+1} in stage $k+1$, the arc cost is $\hat{E}^i_k(s_k, s_{k+1} - 1)$ as computed in (A.7). In other words, the cost of an arc is the MSE of travel time estimation for its corresponding roadway link.

It is easy to check that the graph constructed in the above manner enumerates all possible states in each stage (1 to K) and all possible configurations (i.e., the starting and ending locations) of each link. It also incorporates all the constraints of the model shown in equations (A.12) - (A.15). More importantly, each path from node 1 in stage 1 to node $N+1$ in stage $K+1$ contains exactly K arcs, each of which represents a possible configuration of a particular roadway link (i.e. its starting and ending sections). In other words, each path represents a potential sensor deployment scenario. Therefore the optimal sensor locations can be achieved by finding the minimum-cost path from node 1 in stage 1 to node $N+1$ in stage $K+1$. Since all arc costs are positive, the DP model can be solved by a shortest-path search algorithm.

Figure A.4(b) depicts an illustrative example to show how the graph can be constructed. In this figure, we deploy 3 sensors on a segment with 6 sections, i.e. $K = 3, N = 6$. Then we have 4 stages and 7 sections. Stage 1 has one state at the first section, stage 2 has four states (sections 2, 3, 4, and 5), and stage 3 also has four states (sections 3, 4, 5, and 6). The fake stage 4 has only one state at the fake section 7. Arcs can only be added from nodes in stage 1 (or 2 or 3) to nodes in stage 2 (or 3 or 4) and from lower-numbered nodes to higher-numbered nodes. The cost of an arc from node s_k at stage k to node s_{k+1} at stage $k+1$ is $\hat{E}^i_k(s_k, s_{k+1} - 1)$ as defined in (A.9). The nodes, arcs, and the costs associated with arcs complete the graph corresponding to the DP model. In this graph, any path from node 1 in stage 1 to node 7 in stage 4 contains three arcs. Each arc actually corresponds to a physical roadway link, and the path represents one possible link configuration (i.e. sensor deployment strategy). For example, we highlight in bold line the path 1 → 4 → 5 → 7. The first arc starts at node 1 and ends at node 4, implying that the associated roadway link starts at section 1 and ends at section 3 (the next link starts at section 4). The second arc starts at node 4 and ends at node 5, meaning the second link contains only section 4. Similarly, the third link contains sections 5 and 6. It is easy to check that the graph enumerates all possible paths and the optimal strategy is represented as the shortest path from node 1 in stage 1 to node 7 in stage 4.

The shortest path in a directed acyclic graph demonstrates the polynomial complexity of the algorithm (see [25] for a formal proof). This graph representation also allows us to solve variants of the problem where practical restrictions can be modeled by removing edges from the graph. One such example is the consideration of existing sensors. [25]

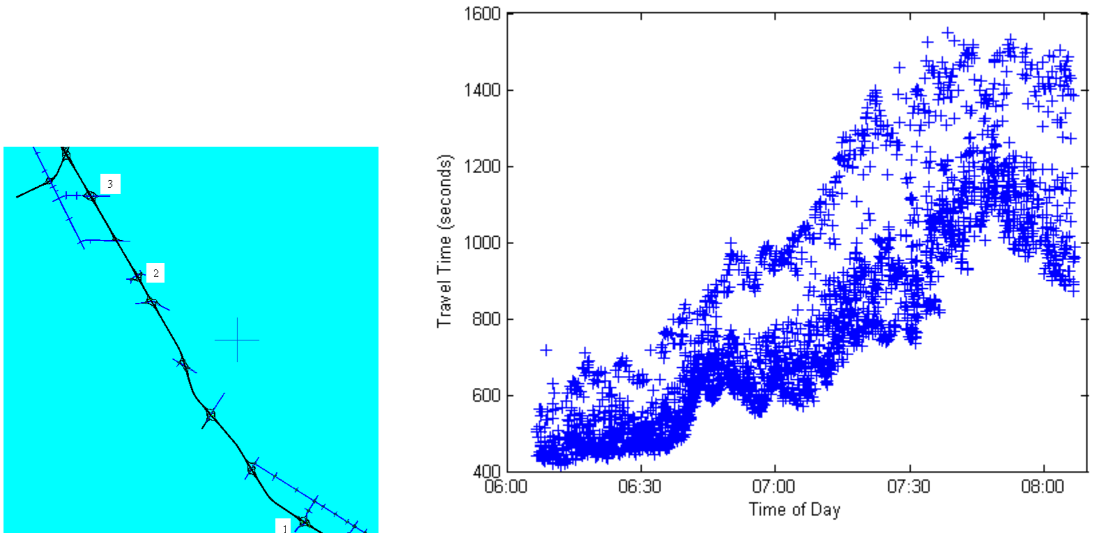
A.5 Results

In this section, we illustrate the proposed DP model and solution algorithm using a case study focused on data obtained from micro-simulation, which provides an ideal analysis framework since all individual trajectories are known. This allows us to investigate for example how the sampling rate (i.e. the percent of trajectories that are available to run the DP algorithm) will impact the sensor location quality.

The micro-simulation data set is for a freeway segment that is roughly 8.7 miles in length from postmile (PM) 17.7 to 26.4. Figure A.4(c) provides an overview of the network in Paramics, in which "1" and "3" indicates respectively the origin and destination of the route. We ran the simulation for 2 hours 30 minutes for morning peak hours from 5:30 am to 8:00 am. We then chose the last 2 hours as the study period, making the total number of 30-second time intervals equal to 240. We divide the freeway segment into 100-foot sections, resulting in $N = 459$. The "representative" vehicles are selected as those who traveled the entire segment and started their trips within the 2-hour study period. There are 3,586 such vehicles, i.e., $M = 3586$. The average travel time of the vehicles is 796 seconds and standard deviation is 227 seconds. Some basic characteristics of the travel times for this network are shown in Figure A.4(d). We can see immediately that for a given time, the range of realized travel times is large. In particular, if we use the average travel time at a 30-second interval as the base, the mean absolute variation of travel times is about 15%. This means that even if the instantaneous method or Coifman method were able to perfectly predict the average travel time for each 30-second interval, we would still see about 15% error due to this variation. Since neither method is error free, we might consider the error above 15% the true error of the method.

We solved the DP model for values of $K = 3, \ldots, 25$ by solving the shortest path problem in the graph described in section A.4. Figure A.5 shows how the objective function changes as we vary the number of sensors. We can see substantial improvement for the first 10 sensors, after which the marginal benefit of each additional sensor becomes small. We also note that the Coifman method clearly outperforms the instantaneous method for calculating travel times for all numbers of sensors. However, both methods lead to similar sensor placements, which is evidence that our model is insensitive to the exact travel time calculation method.

Figure A.6 shows how the optimal sensor locations change as we increase the number of sensors. We can see that when the number of sensors is small, the optimal sensor locations are where the bottlenecks tend to occur (see figure A.3). As the number of sensors increases, we see that these same locations are still used and the additional sensors are placed to "help out" the ones that were originally there (when the number of sensors was small). This is evidence that our model progressively chooses the best sensors as the number of sensors to place increases. This means that if there is a budget for 5 sensors now and in the future the



(c) Paramics Simulation Network.

(d) Vehicle Travel Times as a Function of Time Entering the Study Segment.

Figure A.4: Simulation Overview.

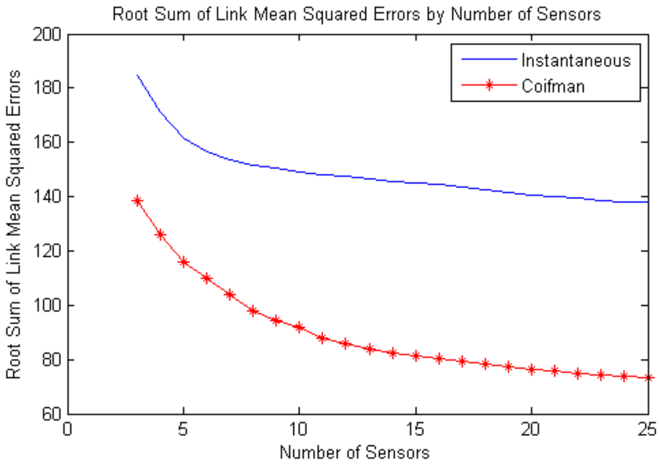


Figure A.5: Total Error (seconds) as a Function of Number of Sensors.

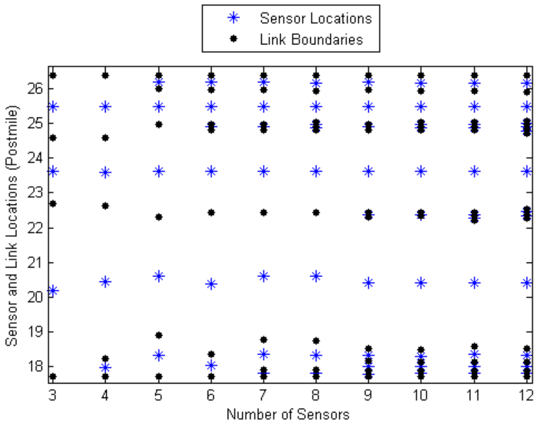


Figure A.6: Evolution of Optimal Sensor Locations Through the Optimization Procedure.

budget is increased to another 3 sensors, the original 5 will still be optimal and the additional 3 will optimally supplement the existing ones.

In [25], we also study trajectories from a case study where the individual trajectories come from vehicles equipped with GPS-enabled cellular phones (known as the *Mobile Century* experiment [52, 99]). The results from this case study led to similar conclusions. See [25] for full details.

A.6 Summary and Possible Extensions

The research work done so far has achieved a specific number of goals. The results that we have obtained lead to conclusions consistent with intuition about data needs on highways and also provide a tool to practitioners to aid in the deployment of actual sensors. The following is a summary of our accomplishments:

1. Development of a modeling framework for solving the optimal sensor placement problem as a shortest path problem on a directed acyclic graph, which can be solved via a polynomial dynamic programming algorithm and can therefore be applied to large-scale scenarios.

2. Analysis of the model led to the conclusion that bottleneck areas (where congestion begins) require more densely deployed sensors than do areas that are generally in free flow.

3. Sensitivity analysis of the DP algorithm led to the conclusion that sensor configurations determined by our algorithm are considerably more stable and predictable compared to evenly spaced or randomly spaced sensor configurations.

4. Adapted the modeling framework to incorporate existing sensors into the model with little increase in the complexity of the algorithm (still polynomial).

5. Established that the model is flexible enough to handle cases with additional requirements such as minimum sensor spacing.

6. Characterized the tradeoffs between number of sensors and information quality, from which we can determine a small range for the optimal number of sensors to place.

7. Validated our results using two different case studies.

The DP model and solution algorithm are have opened numerous avenues to fully understand the problem of sensor placement. In the process of performing this study, we have identified several issues that remain unanswered:

1. How sensitive is the DP model to different travel time computation methods? We have only tested our model using instantaneous and Coifman travel time methods, so we cannot apply our conclusions to all travel time calculation methods. Furthermore, we assume that travel times are computed on a link by link basis using one sensor per link. That means that our model cannot be used with more sophisticated travel time methods.

2. How sensitive is the model to different sets of vehicle trajectories? We used a single simulation run and a single experiment to test our algorithm. Clearly, the quality of a sensor configuration depends on its performance over many days, not just a single one. Therefore, to fully evaluate a sensor configuration, one needs to study its performance over a long period of time.

3. How can we account for sensor errors? Sensors frequently provide inaccurate data. Our model does not take this fact into account. The value of a sensor configuration is dependent upon the ability to obtain good traffic data even when some subset of the sensors give inaccurate data. Additionally, different types of sensors will have different errors and this needs to be accounted for.