# Traffic Estimation and Detection Methods Utilizing Automatic Vehicle Identification Systems

*Sean Lawlor*

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

February 2017

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Doctorate of Engineering.

**2017/05/07**

# Contents

# Abstract

Traffic estimation and detection methods have been used for decades to study the roads in urban environments. These road networks and the traffic patterns present on them have traditionally been studied with point-sensor systems such as inductive loops, which record the number of vehicles on a road segment. However in recent time, advances in automatic vehicle identification (AVI) sensors have allowed for a more advanced sensor deployement on these urban roads. These sensors record a unique identifier for a vehicle at each sensor location allowing the vehicles to be *tracked* over time.

This thesis presents three topics utilizing data from AVI data to perform a series of tasks ranging from convoy detection to estimation of the traffic flow on an urban road network to estimation of the origin-destination (OD) patterns of travellers on a road network. In the first article, we present a method for identifying vehicles which appear to be traveling in dependent patterns through a sensor network deployed in an urban road environment. The next article looks at expanding the model for nominal traffic to allow for time-varying changes in the traffic as the day progresses. Finally we present a method in the last article which recreates an OD matrix from a stream of AVI data into a time-varying mixture model of the OD matrices present in the road network.

The presented methods have applications ranging from law enforcement (for convoy detection), to emergency evacuation management (time-varying traffic pattern estimation), to city planning (estimation of time-varying OD matrices). The collection of methods which are presented in this thesis enrich the field of traffic engineering by allowing models which are only dependent on the data instead of prior biasing information as well as having applications in real-time environments. In the three manuscripts presented, we lay out the analytical methods for detection as well as estimation. We then analyze the algorithms' performance on real and simulated data throughout this work.

# Sommaire

Les méthodes d'estimation et de détection du trafic ont été utilisées pendant des décennies pour étudier les routes en milieu urbain. Ces réseaux routiers et les modes de circulation qui y sont présents ont été traditionnellement étudiés avec des systèmes de capteurs ponctuels tels que des boucles inductives qui enregistrent le nombre de véhicules sur un segment de route. Toutefois, ces derniers temps, les progrès réalisés dans les capteurs d'identification automatique des véhicules (AVI) ont permis un déploiement de capteurs plus avancé sur ces routes urbaines. Ces capteurs enregistrent un identifiant unique pour un véhicule  chaque emplacement de capteur permettant aux véhicules d'être suivis au fil du temps.

Cette thèse présente trois thèmes utilisant des données de données AVI pour effectuer une série de tâches allant de la détection de convoi  l'estimation du flux de trafic sur un réseau routier urbain  l'estimation des profils origine-destination des voyageurs sur un réseau routier. Dans le premier article, nous présentons une méthode pour identifier des véhicules qui semblent se déplacer en motifs dépendants  travers un réseau de capteurs dployé dans un environnement routier urbain. L'article suivant se penche sur l'expansion du modèle pour le trafic nominal afin de permettre des changements dans le temps du trafic mesure que le jour progresse. Enfin, nous prsentons une méthode dans le dernier article qui recrée et la matrice OD d'un flux de données AVI dans un modèle de mélange variant dans le temps des matrices OD présentes dans le réseau routier.

Les méthodes présentées ont des applications allant de l'application de la loi (pour la détection de convoi),  la gestion d'évacuation d'urgence (estimation de la tendance du trafic variable dans le temps),  l'urbanisme (estimation des matrices OD variables dans le temps). La collection de méthodes présentées dans cette thèse enrichit le domaine de l'ingénierie du trafic en permettant des modèles qui dépendent uniquement des données au lieu des informations de polarisation préalable ainsi que des applications en temps réel. Dans les trois manuscrits présentés, nous présentons les mthodes analytiques de dtection et

d'estimation. Nous analysons ensuite la performance des algorithmes sur des donnes relles et simules tout au long de ce travail.

# Acknowledgements

I would like to begin by thanking my advisor, Professor Michael G. Rabbat. For over five years he has diligently supervised both my Master's and Doctoral thesis work. Without his constant guidance, in both research as well as outside that realm, I would not be where I am today. In addition, I would also like to thank the additional members of my Doctoral advising committee, professors Frank Ferrie and Russel Steele. Their guidance has helped lead the work summarized in this thesis to the state it is today, of which I am so proud.

A deep thanks also belongs with Genetec Inc and their founder and CEO Pierre Racz. Not many companys would agree to take a graduate researcher under their supervision. They have treated me so well with access to data and additional resources as well as providing their valuable time and support I will forever be grateful to them.

I would also like to thank my parents, who have always supported my education through my entire life. They started me on the path that allowed me to contribute to my field in ways I can always be proud about.

Lastly I want to thank the love of my life, my wife Tanya. She too has supported my academic aspirations and supported me through the many hours of work required to complete my academic aspirations. I know her support will always continue throughout my future endeavors and I am forever thankful for that.

# Preface and Contribution of Authors

This dissertation is the original intellectual product of the author, and is the result of research undertaken between September 2013 and December 2016, under supervision of Professor Michael G. Rabbat within the department of Electrical and Computer Engineering, McGill University. As a result of this research, a number of manuscripts have been published in peer-reviewed journals and conferences, which are listed below. The three journal articles (numbers 2, 4, and 5) constitute chapters 4, 6, and 8 of this thesis, respectively.

1. S. Lawlor and M.G. Rabbat, "Detecting Convoys in Networks of Short-Range Sensors", 2014 Asilomar Conference on Signals, Systems, and Computers. Pacific Grove, CA, USA. November 2014

2. S. Lawlor, T. Sider, N. Eluru, M. Hatzopoulou, and M.G. Rabbat, "Detecting convoys using License Plate Recognition Data". IEEE Transactions on Signal and Information Processing Over Networks (TSIPN). September 2016

3. S. Lawlor and M.G. Rabbat, "Estimation of Time-Varying Mixture Models: An Application to Traffic Estimation". 2016 Workshop on Statistical Signal Processing (SSP). Palma de Mallorca, Spain. June 2016

4. S. Lawlor and M.G. Rabbat, "Time-Varying Mixtures of Markov Chains: An Application to Traffic Modeling". *submitted* IEEE Transactions on Signal Processing (TSP)

5. S. Lawlor, N. Eluru, M. Hatzopoulou, and M.G. Rabbat, "A Mixture Model Approach to Origin-Destination Matrix Estimation with Routing Information". *submitted* Transportation Research Part B: Methodological.

## Contribution of Non-Supervisory Authors

The authors T. Sider, N. Eluru, and M. Hatzopoulou contributed to work number 2 by providing access to the PTV VISUM simulation data of the city of Montreal, Quebec, Canada and consulted on the use of this data. In work number 5, N. Eluru and M. Hatzopoulou provided access to the PTV VISUM simulation data as well as provided pointers to the literature on traditional estimation techniques and error metrics utilized for OD matrix estimation in the transportation engineering research field. They also made suggestions regarding the writing of the paper, to make it more suitable for submission to the transportation research journal.

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| LPR | License Plate Recognition |
| AVI | Automatic Vehicle Identification |
| GMTI | Ground Moving Target Indicator |
| SMP | Semi-Markov Process |
| DTMC | Discrete Time Markov Chain |
| CTMP | Continuous Time Markov Process |
| KDE | Kernel Density Estimation |
| ML | Maximum Likelihood |
| MLE | Maximum Likelihood Estimate |
| CDF | Cumulative Distribution Function |
| OD | Origin-Destination |

# Chapter 1

# Introduction

## 1.1 Framework

The overlaying theme of this thesis is the estimation of models using automatic vehicle identification (AVI) sensors, and more specifically in much of this work, license plate recognition (LPR) sensors. The following chapter contains examples of other classes of AVI sensors. Automatic vehicle identification sensors capture a unique identifier and metadata about vehicles when they pass within the sensor's detection region. The emergence of lower cost and higher accuracy sensors has allowed for many more urban road networks to be equipped with these sensors. They provide a per-sensor stream of point-measurements of a vehicle which passes by. Traditional sensors deployed in traffic estimation are inductive loops which are buried under the road and only provide a count of the number of vehicles which pass above them. Utilizing AVI data allows for types of signal processing techniques to be applied which were could not be used with only traffic count data from inductive loops. In addition to simply modeling the flow of traffic, a multitude of other estimation and detection tasks can be performed such as identifying patterns or anomalies within the road networks.

This thesis makes use of the information-rich data coming from AVI sensors to perform various statistical analysis on the data. We utilize AVI data to estimate the time-varying traffic flow through a road network as well as detect vehicles which are traveling as convoys through the road network. All of the detection and estimation techniques presented are in a centralized fusion-center style system. The methods presented can be quite complex and utilizing a fusion center style system allows a simpler notation to present these initial methods. We demonstrate that utilization of AVI data can propel the methods traditionally used for convoy detection and traffic estimation to new heights, as well as doing so in an efficient manner.

## 1.2 Motivation

Automatic vehicle identification data is commonly used today for a wide range of applications. Traffic law enforcement forces around the world use LPR sensors to automate the processes of running checks on unpaid registrations. Parking enforcement offices also use LPR to enforce parking restrictions in cities world-wide. Other types of AVI sensors are also used for travel-time estimation at many country border crossings to give travellers an estimate of their wait time before coming to customs and immigration services. However the use of AVI data to estimate traffic flow models has received limited study.

Traffic flow and estimation models are commonly used by a multitude of city service departments. For example, city planners will consider traffic models to determine where new roads need to be added or existing ones expanded. Also police will use these types of models to determine optimal placement of traffic enforcement officers to detect anomolous or target patterns in the traffic flow. In this work we propose techniques to transform AVI data into actionable models which can be used by these various sources.

## 1.3 Thesis Outline and Contributions

This manuscript-based thesis is comprised of three articles and six chapters of supplementary material, including this introduction chapter. Chapter 2 provides background on the field of time-varying mixture models, traffic estimation, and convoy detection. The background includes a brief history of each field addressed in this thesis, as well as the terminology and methodologies used in each field.

Chapters 4, 6, and 8 constitute three manuscripts which are either accepted or currently under review of peer-reviewed journals. They are presented in chronological order to most accurately represent the progression of the research presented in this thesis. Chapter 3 begins with an explanation of our interest in the problem of convoy detection and then presents the first manuscript in Chapter 4, titled *Convoy Detection Using License Plate Recognition Data.* The contribution of this chapter is the introduction of the problem of detecting groups of vehicles traveling through an urban road network as a convoy (i.e. in a dependent fashion) as well as the proposition of a statistical detection method and assesing its performance empirically using simulated and real data sources. We formulate a mathematical notion of what it means to be a convoy and then present a method based on a sequential hypothesis test to detect convoys traveling in a streaming-data environment. Chapter 4 ends with a discussion on the transition of research direction to time-varying models for traffic behavior, addressed in Chapter 6.

Chapter 5 introduces the second manuscript in Chapter 6 titled *Time-Varying Mixtures of Markov Chains: An Application to Traffic Estimation*. It is the estimation of time-varying mixture models of Markov chains utilizing a new algorithm, titled the "Automatic Hard EM algorithm". In this contribution, the complex notion of a time-varying mixture-model of Markov chains is presented. This model can be used to model networks which change their structure periodically and need to be remodeled in near real-time. This model allows for the addition, deletion, and movement of components in the mixture model as time progresses which allows much greater power and flexibility in complex network modeling. Traditionally these types of dynamic models require complex estimation techniques, such as Dirichlet process mixture models or complex Expectation-Maximization techniques, which cannot guarantee convergence in a time-frame which might be applicable in real-time applications. The second contribution of this work is then the introduction of the Automatic Hard EM algorithm which provides a method of model order and parameter estimation of time-varying model of mixtures of probabilistic models.

Chapter 7 introduces the final manuscript in this thesis, which is presented in Chapter 8 and titled *Origin Destination Matrix Estimation Utilizing the Automatic Hard EM Algorithm*. This work extends the work presented in Chapter 6 to the field of traffic engineering. In this contribution, an extension of the model presented in Chapter 6 that captures the destination of a vehicle, is provided. This new model then represents both the origin state as well as the destination state of vehicles along with the routes followed between origins and destinations. From this we propose using the Automatic Hard EM algorithm to fit this model, which can be further transformed into an Origin-Destination (OD) matrix commonly used in city planning and traffic research. However the main contribution of the OD matrix estimated using this technique is the automatic association of specific origin-destination pairs and their associated routing information. This gives a wealth of additional information not traditionally encoded in OD matrix estimation techniques.

We conclude this thesis in Chapter 9 with a proposal of possible extensions to the work presented herein. We focus on four proposed extensions. The first is the application of the Automatic Hard EM algorithm to mixtures of other probabilistic distributions, such as Normal distributions. The second extension is a deeper investigation to the implications of using the Kullback-Leibler divergence which is a key component in the presented Automatic Hard EM Algorithm. The third extension is related to the addition of traffic count data to augment poor deployment coverage of AVI sensors in OD matrix estimation. Traffic count sensors, such as inductive loops, are already heavily deployed in traffic networks and are an economical option to augment the data retrieved from AVI sensors. We propose using this type of data to correct errors in OD estimates using only AVI data. Finally we propose an extension which would extend the models to account for errors in the AVI data.

# Chapter 2

# Background

## 2.1 Chapter Introduction

The following sections outline the background in the order of how the research progressed as well as the state of-the-art for each of the topics discussed, beginning with the more detailed introduction of AVI data in Section 2.2. Next in Section 2.3, the problem of convoy detection is introduced, which is detecting groups of vehicles travelling dependently through a network of AVI sensors. I then move to the problem of how does one more accurately estimate background traffic using time-varying mixture models. We conclude with the introduction of the problem of origin-destination (OD) matrix estimation using these newly introduced models for traffic movement. This chapter serves to outline the problems addressed in the main sections of the thesis as well as to provide brief background on these topics.

## 2.2 Automatic Vehicle Identification

Traditionally traffic engineers have deployed sensors of the form of inductive loops, which are deployed below a road segment and capture the number of vehicles passing above them. In this thesis we utilize a new emerging technology which is gaining popularity in traffic research called AVI sensors. Automatic vehicle identification (AVI) sensors are a class of sensor which provide information in the form of *vehicle $V$ was observed at location* $(X, Y)$ *at time $T$*. In this work we only address the sub-class of AVI sensors where their location is static, i.e. the location of the sensor is roughly the location where a vehicle is observed at. Each type of AVI sensor captures a unique identifier for a vehicle which is dependent on the sensor being utilized. Examples of AVI sensors are

1. Bluetooth sensors: These sensors capture the MAC address of the in-vehicle entertainment system or cell phones which are inside the vehicle at the time. When a MAC address is read, the time and location of the capture are associated and recorded.

2. Licence plate recognition: These sensors are cameras which run a specialized object-character recognition algorithm which identifies and records the license place of vehicles which pass in front of them.

3. EZPass readers: EZPass is a technology deployed in regions of the United States, with comparable systems deployed elsewhere around the world, which is a specialized device which allows people to pass through tollbooths without stopping and charges an account instead of paying cash at the booth. EZPass readers can also be deployed in a read-only fashion where the EZPass ID is recorded and the account is *not* charged. Every time an ID is captured, it constitutes a unique identification of the vehicle with the EZPass ID and collections of these reads are a form of AVI data.

Given a collection of AVI data, one can organize the AVI observations into observation sequences by ordering by the unique vehicle identifier. A sequence of observations of a vehicle is an irregularly-sampled time-series of locations where the vehicle was observed. From collections of AVI data in this form, this thesis attempts to solve the problems of traffic estimation and convoy detection as outlined in the following.

## 2.3 Convoy Detection

A convoy of vehicles is two or more vehicles travelling through a network of roads in a dependent fashion. This means that the group of vehicles travelling as a convoy follow similar paths within a small timeframe from the others in the group. Early works studying convoy detection methods utilize the ground moving target indicator (GMTI) data [PPR09, Koc02]. This class of sensor, as previously stated, have a very wide field-of-view and long-range which allows them to provide more continuous information about the vehicles being tracked. They can provide samples of the vehicles within their range continuously until their leave their detection area. These sensors also can provide observation samples of the vehicles with a regular frequency opening the class of convoy detection methods to alternate approaches. Lastly in scenarios utilizing GMTI data, typically only one or a few sensors are deployed to perform the tracking and convoy detection.

In more recent research for convoy detection, trajectory databases have been proposed. These utilize information captured from GPS devices, such as a cell phone or GPS navigation system, for a collection of vehicles. Together a database of vehicle trajectories over time is built. Inside this database, as a post-processing step, convoy detection has been performed using a variety of methods [JLO07, JYZ+08, KMB05, WBH12]. In addition, a decentralized method for these types of trajectory databases has been proposed in [YD12].

Lastly more recent work in convoy detection using a class of AVI sensors, license plate recognition, has been proposed in [HHZ+11]. This method utilizes a database of recorded LPR observations and it thresholds the number of co-occurrences to try and identify convoys of vehicles following exactly the same path. The method proposed in Chapter 4 utilizes the same type of data however utilizes alternate statistical methods to identify vehicles travelling as a convoy inside nominal traffic.

### 2.3.1 Formal Definition of a Convoy

In order to address the problem of a convoy detection, first an understanding of what it means to be a convoy is necessary. In the work presented here, we propose considering a convoy as a group of two or more vehicles which follow a highly dependent route through the road network. In order to determine if vehicles are travelling as a convoy versus simply travelling normally through the network of roads we propose utilizing two models in a hypothesis testing framework. The first model is the null hypothesis which is the model that states vehicle trajectories are governed by a mixture model of discrete-time, first-order Markov chains. Secondly the alternate hypothesis utilizes the distance covered by vehicles travelling similar routes as well as the co-dependent times in which they make the transitions between sensors.

### 2.3.2 Hypothesis Testing in Real-Time Applications

In addition to detecting convoys using AVI data, we would also like to do so in real-time applications. In order to do this, we utilize sequential hypothesis testing [Wal66] techniques for each pair of vehicles being tracked. I only address the problem of detecting pairs of vehicles travelling together because the construction of a larger group from pairwise detections is a trivial problem (as shown in Chapter 4) whereas deriving a method which tracks a varying size convoy group is significantly more complex. As observations are received, the likelihood that the observation comes from the null or alternate hypothesis is updated and a likelihood ratio between the two computed. When this ratio exceeds a high or low boundary a decision for the alternate or null hypothesis can be made. The choice of these boundaries will heavily impact the detection rate as well as the rate of false positive detections. In order to choose the upper and lower boundaries, Wald in [Wal66] demonstrates that these limits can be chosen to guarantee certain false-positive and detection rates. When the likelihood ratio is between these two boundaries then the statistical hypothesis test waits for more data to arrive in order to update its estimate further.

## 2.4 Time-Varying Mixture Models

Time-varying mixtures of probability models are useful tools to estimate complex, time-varying datasets. Use of these complex models is traditionally limited by the ability to estimate the parameters of the distributions using algorithms which take a long time to converge to accurate parameter estimates. In Sections 2.4.1 and 2.4.2 we outline the state of the art estimation methods for these class of models as well as their limitations.

### 2.4.1 History of Time-Varying Mixture Models

Historically time-varying mixture models are viewed in the context of Dirichlet process mixture models [CDD07] which adopt a Bayesian non-parametric approach to the estimation of these mixture models. They state that there are a infinite number of components in the mixture model where only a countable number have a measurable mixture weight. This means that at any point in time only a set number of mixture components are governing the behaviour of the observed data up to that point in time. For example, Stephens [Ste00] proposed a Bayesian representation of a mixture model using a marked point-process.

These models traditionally perform inference utilizing Markov Chain Monte Carlo (MCMC) methods where computational performance guarantees are difficult to achieve due the random sampling nature of MCMC inference methods. In addition the term *curse of dimensionality* is typically attached to MCMC estimation of high dimensional mixture models.

This is because MCMC methods, which sample candidate models such as reversible jump MCMC, sample candidates from the space of possible models and then either accept or reject the candidate. If the parameter space is quite high, for example as with Markov chains in the context studied in this thesis, then it is very difficult to sample good candidate points.

The problem of parameter estimation in high dimensional mixture models also typically uses high-dimensional mixtures of Gaussian distributions where additional properties of the Gaussian distribution can be leveraged to simplify the models and estimation techniques. In the general setting, these types of simplifications are not typically available.

If one views a time-varying mixture model at a single point in time, then the model at that time-point can be estimated as a mixture model with an unknown number of components. Therefore I also provide a review of the literature of mixture model estimation in the non time-varying setting but with an unknown number of components in the mixture model. The problem of automatic model-order selection was initially addressed utilizing model selection criteria such as the Akaike information criterion (AIC) [Aka74], Bayesian information criterion (BIC) [Sch78], and minimum description length (MDL) [Ris78] criterion. To utilize these criterions, multiple candidate models with varying parameters and number of components are estimated and these are used to score the fit of the model. The computation of the class of candidate models can be very time-consuming and computationally expensive when the model order or dimension are high. Therefore these methods are best suited for small models or models with a low dimension.

Moving forward, Corduneanu and Bishop [CB01] utilize another Bayesian representation of a mixture model and utilize variational methods to anneal the number of components starting at a pre-specified maximum value. Taking a similar approach is the CEM$^2$ algorithm introduced in [FJ02]. This work has had high interest and continued on with Chen et al. [CK08] and Huang et al. [HPZ13] which remove the variational estimation schemes of [CB01] for a more traditional EM approach. However in all of these works, it is necessary to specify some maximum number of possible components which are then annealed to the proper number.

The most closely related work to the method presented in this thesis in Chapter 6 is that of Verbeek et al. [VVK03] which attempts to solve the problem of mixture model order selection without specifying a maximum number of mixture model components and instead add components in a greedy nature as necessary. At each iteration, the algorithm in [VVK03] will compute the fit of $k$ specified components and then perform a search over a set of possible candidates to determine if a new candidate model is necessary and if so, add it and refit the model utilizing the EM algorithm. However as is traditionally presented in the Dirichlet process mixture model frameworks, Verbeek et al. utilize Gaussian models for their mixture model components and exploit features of the Gaussian distribution to limit their search over the field of possible candidate models.

### 2.4.2 Estimation of Time-Varying Mixture Models

As Sec. 2.4.1 demonstrates, the problem of time-varying mixture models has received a relatively high degree of study. The work I present in this thesis however assumes a different class of model which to the best of my knowledge has yet to be studied. In order to not consider infinite parameter spaces as is typically done in Dirichlet process mixture models which are applicable to time-varying applications, a fully generative specification of the model dynamics is necessary. I leave the definition of the dynamics to the relevant chapter later, Chapter 6, however a brief overview of time-varying discrete mixture models is helpful.

Consider a mixture model at time $t$ which contains $M(t)$ components. As time progresses from $t$ to $t+1$, the model specified by $M(t)$ components should be updated to contain $M(t+1)$ components given newly observed data. However we still wish to propagate information contained in the model at time $t$. This model is useful for data which arrives in batches or sequentially and can be batched into time-windows and where the assumption that time-windows are independent of each other is not possible. We wish to estimate the parameters of this mixture model as data arrives and update it accordingly. Between time-windows, components in the mixture model can change, die, or new ones can be created. This means that $M(t+1)$ is not necessarily the same as $M(t)$.

## 2.5 Origin-Destination Matrix Estimation

The ultimate goal of the mixture model presented in Chapter 6 and the modified version of this model presented in Chapter 8 is to estimate traffic patterns. These patterns are the observed paths through the network of AVI sensors deployed in an urban environment. Traditionally in traffic engineering, the estimation of traffic flow in a city is done via estimation of the origin-destination (OD) matrix. The OD matrix, also called the trip-table matrix, is a matrix where each cell $(i, j)$ of the matrix denotes the number of vehicles which travelled from a regional zone $i$ to zone $j$.

### 2.5.1 History of Origin Destination Matrices

Estimation of origin destination matrices has been well studied in traffic engineering research. Work in this field has ranged from simple Fratar or growth models [Eva70], which do not differentiate trips by purpose and assign a simple growth factor to account for population growth to an existing OD matrix, to more complex methods which include the gravity-type model and the intervening opportunities model [O'N87]. The gravity model is still the most widely used model for OD matrices and OD matrix estimation techniques. Origin-destination matrix estimation techniques traditionally start with a target OD matrix and then one of several estimation methods is applied to update this target to current demands given traffic count data. A large class of work has been applied to this problem and multiple solutions for the matrix estimation problem have been proposed. Sherali et al. [SNS03] and Peterson [Pet07a] together provide a good overview of this class of OD matrix estimation techniques.

Methods of OD matrix estimation which are seeded with a prior OD matrix typically suffer in that they cannot rapidly adapt to mass changes in traffic patterns. Being based on an informative prior such as a previous OD matrix heavily biases future estimates to appear to be of the form of the provided OD matrix [CMJ08]. In real-world scenarios, short-term traffic pattern estimates are highly volatile. However long-term trends typically demonstrate that patterns hold for long periods, making these classes of estimation techniques appealing. A downside to these techniques is that the monetary as well as physical cost of gathering a prior OD matrix can be exorbitant. They are typically generated from manual surveys which asses where people live and work, their typical mode of transport, as well as a multitude of other factors over an extended period of time. These survey results are then gathered by experts to create a target OD matrix. However this OD matrix is only truly valid for a short period, but the cost of recreating it is very expensive, and therefore they tend to be utilized long after their true validity period. Utilizing only traffic count data it is difficult to see how a better estimate than that proposed using a prior OD matrix would be possible.

In today's road network environments, access to sensors which provide more rich information than simply just traffic counts is available. In many instances, highly accurate forms of AVI sensors are already deployed for city surveillance uses[1]. The use of these class of sensors have been applied to OD matrix estimation in [Zij97, DR05]. Both of these works utilize AVI data in conjunction with traffic counts to improve the performance and reliability of OD matrix estimates given missing information in the prior OD matrix or errors in the arriving traffic count data.

Traditional OD matrix estimation methods do not account for the paths which vehicles take to get from an origin to a destination. This is because of limitations in the information contained in traffic count data. This is because it is often impossible to say which traffic counts influence other traffic counts when there are multiple paths available through the road network. Utilizing AVI type data, we then gain the ability to *track* a vehicle from one point to another by matching the AVI identifications. This is a core property of AVI data which is exploited in the paper in Chapter 8.

[1] An example of a city utilizing city-wide surveillance with license plate recognition is the town of Palm Beach, FL, USA which uses Genetec's AutoVu LPR system deployed across it's city`https://www.genetec.com/solutions/resources/town-of-palm-beach`

# Chapter 3

# Convoy Detection

This thesis begins with a manuscript examining the problem of detecting convoys of vehicles travelling in an urban road network. License plate recognition sensors, which are a type of AVI sensor, are manufactured by the company which sponsored this research, Genetec. Genetec has frequently been asked by their law enforcement and city planning customers if use of their city-wide LPR surveillance systems could be used to detect groups of vehicles travelling together as a convoy. Therefore this type of data is what was utilized to try and solve this problem.

Data from this type of sensor required a unique perspective in order to address issues which became present through initial analysis. Organizing observations of a specific vehicle into the order which they were observed results in an irregularly-sampled time-series of observations of the vehicle. Many methods initially investigated in my Master's thesis, [Law13], were proven infeasible due to the issues with the irregularly sampled nature of the data. An initial version of the convoy detection method was proposed in [Law13] which described the sensors to be considered the states of a Markov chain.

By considering the sensors to be the states of a Markov chain we were able to handle many of the issues which arose from the irregularly sampled data. However [Law13] only considered traffic to be distributed according to a single discrete-time Markov chain and ignored the time it took for a transition from one state to another. The manuscript in Chapter 4 takes the model initially proposed in [Law13] and extends it to a mixture of Markov chains which allows for a more expressive model. In addition to utilizing a mixture of Markov chains, the model presented in the following manuscript includes the time it takes a vehicle to transition from one state to another as well as modelling how transition time affects vehicles travelling as a convoy. It also proposes a novel sequential definition of how a convoy of two or more vehicles transitions through a network of sensors.

# Chapter 4

# Paper: Convoy Detection Using License Plate Recognition Data

[1]*License plate recognition* (LPR) sensors are embedded camera systems that monitor road traffic. When a vehicle passes by a sensor, the vehicle's license plate, the location, and the time of observation are recorded. Given a stream of such observations from a collection of sensors spread around the road network, our goal is to detect convoys: groups of two or more vehicles traveling with highly correlated trajectories. Some of the main challenges with modeling and processing data from LPR sensors include that the data-gathering process is event-driven, thus data are not regularly sampled in time or space. Also, an appropriate definition of convoy should be relative to background traffic patterns which are temporally and spatially varying. This paper proposes novel models for LPR observations of traffic which are well-suited for online convoy detection. Baseline traffic is modeled as following a mixture of semi-Markov processes, and specific models for temporal and spatial correlation of observations of vehicles traveling in a convoy are introduced. These models are used within a sequential hypothesis testing framework to obtain a system for real-time convoy detection. The model of baseline traffic may be of independent interest for forecasting road traffic patterns. Experiments with an extensive simulated dataset illustrate the performance of the scheme and offer insights into the tradeoffs between detection rate, false alarm rate, and the expected number of observations required to detect a convoy.

[1] The contents of this chapter have been pubished in the paper: S. Lawlor, T. Sider, N. Eluru, M. Hatzopoulou, and M. Rabbat, "Detecting convoys using license plate recognition data," *IEEE Trans. on Signal and Information Processing over Networks*, vol. 2, no. 3, pp. 391–405, Sep. 2016.

## 4.1 Introduction

We consider the problem of detecting convoys of vehicles in an urban environment using a collection of *license plate recognition* (LPR) sensors. Each sensor records data of the form “vehicle $X$ was observed at location $Y$ at time $t$”. Given streams of such observations arriving from a collection of sensors, a centralized decision maker must identify which, if any, vehicles are traveling as convoys.

Convoy detection has applications in both law enforcement and the commercial sector. Law enforcement agents may be interested in detecting and tracking convoys for a variety of reasons [1]. In the commercial sector, the approach developed in this paper could be used to identify groups of shipping vehicles traveling along highly correlated routes which may benefit from forming platoons. Recently there has been interest in designing control laws to allow heavy-duty shipping vehicles to maintain platoons over long distances in order to reduce drag on the non-leader vehicles, thereby saving on fuel costs [2,3]. In order to exploit this approach one must first identify potential pairs of vehicles that could form platoons, and the convoy detection approach we propose could be used to automate this process.

Defining a concrete notion of what it means to be a convoy is not as straightforward as it may seem. Intuitively a convoy comprises two or more vehicles traveling together. While it may be tempting to particularize this definition to say that a convoy is two or more vehicles traveling along the same route over a given distance (e.g., for more than 500 consecutive meters) or for a minimum amount of time (e.g., at least 5 minutes), without separating by more than a particular distance (e.g., 50 meters), such a threshold-based approach has a number of limitations and drawbacks. Setting the thresholds too tight does not allow for situations where the convoy vehicles take slightly different routes (e.g., deviating for a few city blocks before rejoining). Similarly, in dense urban environments or along stretches of highway during rush hour it may be expected that arbitrary vehicles will be seen near each other for a relatively long distance and/or time even if they are not traveling as a convoy, simply because of the dense traffic.

Similar to problems of unsupervised novelty/anomaly detection [4–6], defining what it means to be a convoy is not straightforward. One may expect convoys to be relatively rare events. Still it is not straightforward to obtain a sample of traffic that is guaranteed to contain no convoys, and it is also not straightforward to obtain labeled examples of convoys for training. Intuitively, two vehicles may be called a convoy if their trajectories are more correlated in space and time than two typical vehicles in normal traffic. The challenge is in making precise what is “more correlated” and what are “typical vehicles in normal traffic”.

Another challenge is due to the fact that measurements arrive at irregular times. Existing LPR sensors use cameras in conjunction with computer vision algorithms to identify and extract vehicle license plates. Consequently, LPR sensors have a short range, and measurements are obtained in an event-driven manner, when a vehicle passes within the field of view of the camera. Thus, measurement times are arbitrary, and measurements of any particular vehicle are not obtained at regular sampling intervals, either in time or space.

The aim of this work is to develop algorithms that detect convoys in real-time. Our approach is based on sequential hypothesis testing [7], and the main contribution of this work is the modeling of observations from such a network of LPR sensors. Under the independent (non-convoy) hypothesis, vehicle movement is modeled as following a mixture of Markov chains, and under the convoy hypothesis a novel leader/follower observation model is developed.

### 4.1.1 Previous Work

The majority of previous work on convoy detection and tracking in the information fusion literature [8–10] focuses on sensors with a wide field of view, such as *ground moving target indicator* radar. Data is collected from one or a few sensors and provides a tracking indicator based on the physical characteristics of a vehicle. Each sensor regularly scans and gathers measurements about the vehicles in its field of view over an extended period of time and over a large geographic region. In contrast, the setting considered in this paper is such that any individual sensor only measures a vehicle when it is nearby the sensor, and individual vehicles are thus only measured intermittently (and irregularly) over time when they pass by a sensor.

Threshold-based approaches have been studied for off-line identification of convoys in trajectory databases [11–14]. Such methods are applicable when entire vehicle trajectories are available (e.g., all vehicles carry GPS units and regularly report their location to a central office, as is commonly the case with taxis and shipping trucks). In such a setting, when a vehicle is also aware of which other vehicles are nearby, convoys can be detected using decentralized methods [15]. In contrast to this previous work, the present paper deals with partially-observed trajectories, sampled when the vehicle passes by an LPR sensor.

In addition, the previous work mentioned above does not take into account the underlying traffic patterns and structure of the road network.

Convoy detection based on LPR sensors (a.k.a., automatic number plate recognition systems) is considered in [1], where a heuristic approach to detecting vehicle convoys in a database of LPR records is proposed. The approach, similar to [12], is based on counting and thresholding co-occurrences of vehicles observed nearby each other. The convoy model considered in [1] requires that the vehicles in a convoy follow precisely the same path, and the method is designed for post-processing of database records rather than real-time/sequential detection.

### 4.1.2 Contributions and Organization

We address the problem of convoy detection using tools from the statistical signal processing toolbox. Specifically, the contributions of this work are: 1) posing the problem of convoy detection using short-range LPR sensors in the framework of sequential hypothesis testing, and 2) developing models for LPR observations under convoy and non-convoy hypotheses. In the non-convoy setting we model vehicle movement using a mixture of Markov models. In the convoy setting, a novel leader/follower measurement model is developed. The convoy model is flexible and does not require all vehicles in the convoy to travel along precisely the same route; rather they should travel in the same general direction (following the leader), and the leader may change over time. The extent to which their routes deviate can be specified in the model, so that the scenario where all convoy vehicles follow precisely the same route is a special case. We evaluate the performance of the proposed approach using simulated data based on a detailed model of road traffic in Montréal, Canada.

The rest of the paper is organized as follows. Section 4.2 provides the problem formulation. Generative models for observations of convoys and independent vehicles are described in Section 4.3. The proposed sequential hypothesis testing framework, including implementation details, is described in Section 4.4. The results of the experimental performance evaluation are reported in Section 4.5. Additional issues are discussed in Section 4.6, and we conclude in Section 4.7.

## 4.2 Problem Description

This section takes steps towards formalizing the problem of convoy detection. We describe characteristics of the measurement system that make the problem challenging. Then we discuss assumptions made and describe performance metrics that will be used to evaluate convoy detection methods.

### 4.2.1 License Plate Recognition Data

Consider a system of urban roads instrumented with license plate recognition sensors. When a vehicle passes by the sensor it records the license plate as well as the time and location of the event. The sensors have a very short range of detection (e.g., 10 meters). The measurements from many of these sensors, at different locations in the road network, report their measurements to a fusion center whose goal is to detect groups of vehicles that are driving together as a convoy.

Formally, we consider a collection of $C$ sensors, indexed using the first $C$ natural numbers, $1, \ldots, C$, and let the set of sensor indices be $\Omega = \{1, \ldots, C\}$. In this paper we focus on detecting convoys composed of two vehicles; the extension to convoys of more than two vehicles is discussed in Section 4.6. Let $(x_1, r_1), (x_2, r_2), \ldots,$ denote a sequence of observations of one vehicle, where $x_i \in \Omega$ is the index of the sensor making the $i$th observation and $r_i \in \mathbb{R}_+$ is the time of the $i$th observation.[2] Similarly, let $(y_1, s_1), (y_2, s_2), \ldots,$ denote the sequence of observations of a second vehicle.

### 4.2.2 Measurement System Characteristics and Assumptions

Fig. 4.1 shows a sample path of the measurement process as a function of time. The horizontal axis corresponds to time; the labels along the bottom of the figure show observation times for each vehicle ($r_i$ and $s_i$), and the labels along the top of the figure show global observation times. The vertical axis gives the index of the camera making the measurement; this index should be treated as a categorical variable since the ordering is arbitrary and does not necessarily reflect, e.g., the geography of the sensors.

Note that the observation times are not necessarily equally spaced, and the number of observations is not necessarily the same for each vehicle. This is because vehicles can

[2] Without loss of generality we take all times to be non-negative and denote by $\mathbb{R}_+$ the set of non-negative real numbers.

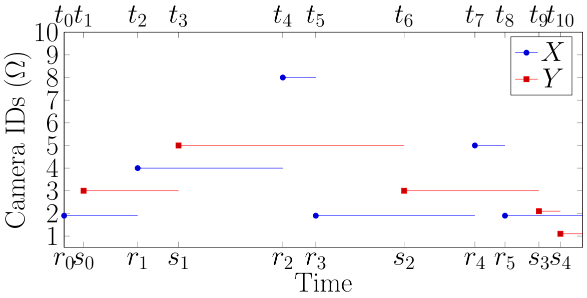


**Fig. 4.1** Example measurements of two vehicles over time.

leave the observation area, people drive at different speeds, and traffic patterns and road conditions vary over time.

Traffic patterns in a large urban environment may also be quite complex. For example, during rush hour there may be a significant flow of vehicles heading from the suburbs into the city and, at the same time, from the city out to the suburbs. This motivates the need for models that can capture these subtle aspects of traffic flows and not just the average or majority flow over the network.

We make the following assumptions about the measurements. First, the sensors are synchronized so that the timestamps from different sensors are directly comparable. This is justified since existing LPR cameras are typically equipped with GPS receivers that can provide reliable and accurate synchronization.

Second, we assume that no two vehicle observations are recorded at precisely the same time instant; this ensures that the two time sequences $\{r_i\}_{i\geq 1}$ and $\{s_i\}_{i\geq 1}$ can be uniquely ordered. This is justified when timestamps at each sensor use a sufficiently high resolution.

Third, we assume that a sensor always records vehicles that pass by the road segment it is monitoring and that the sensor does not produce any spurious measurements. Thus, there is no "noise" in the measurement sequences (missed observations or erroneously injected observations), and the main source of uncertainty is in the vehicle trajectories. While it is certainly of interest to allow for such additional noise sources, we leave this as an extension for future work.

Fourth, we assume that the sensors are static and that their locations are known to the fusion center. Thus, the fusion center can make use of related information, such as the distance between sensors, when making a decision.

Finally, we assume that the sensors transmit their measurements to the fusion center over a reliable, delay-free channel; i.e., we consider a traditional centralized decision making setup. This is reasonable since each individual measurement can be encoded in a small number of bits (e.g., much smaller than the size of a typical Ethernet packet) and the inter-observation time for a given vehicle (i.e., the time between when it is observed at one sensor and next observed at a different sensor) is large relative to the time it takes to transmit such a measurement using contemporary communication technologies.

### 4.2.3 Sequential Testing and Performance Metrics

In this work we consider a typical sequential hypothesis testing setting [7] where the observations $(x_i, r_i)$ and $(y_i, s_i)$ arrive successively at the fusion center. Under the null hypothesis, $H_0$, the vehicles are independent (not a convoy), and under the alternative hypothesis, $H_1$, the vehicles are moving as a convoy. After receiving an observation the decision maker must choose from one of three options: 1) declare that the pair of vehicles is a convoy (i.e., reject the null), 2) declare that the pair is not a convoy (i.e., fail to reject the null), or 3) wait to receive additional observations. As discussed in Section 4.1, defining what it means to be a convoy is difficult. Ultimately, the precise definition of convoy adopted in this work is implicit in the models described in Section 4.3.

The objective is to make accurate decisions without deferring for too long. Accuracy is measured using the standard metrics for hypothesis testing: the probability of detection and probability of false alarm. We also study the average number of observations required to make a decision. Ideally a system should have high probability of detection, low probability of false alarm, and a low average number of observations required to make a decision.

## 4.3 Modeling

Our aim is to formulate the problem of convoy detection in the sequential hypothesis testing framework. The main task is one of modeling; i.e., to define appropriate distributions for the observations under the hypotheses that $(H_1)$ the two observed vehicles are a convoy, or $(H_0)$ the vehicles are not a convoy. First we describe a simple Markov model for the observations of individual vehicles. Then we build on this to develop models for observations of pairs of vehicles under each hypothesis.

### 4.3.1 Single-Vehicle Markov Model

To begin, we define a model for the observations of a single vehicle, $\{(x_i, r_i)\}_{i=1}^n$. Our model can be viewed as a semi-Markov process [16], where the sequence of sensors where the vehicle is observed, $x_1, x_2, \ldots,$ follows a discrete-time Markov chain, and the inter-observation times $r_i - r_{i-1}$, $i = 2, \ldots, n$, are mutually independent and are conditionally independent of the other variables given the states $x_{i-1}$ and $x_i$.[3]

Let $(\pi_x)_{x\in\Omega}$ denote the initial state distribution, with

$$\sum_{x\in\Omega} \pi_x = 1,$$

and let $P_{x_{i-1},x_i} = \Pr(x_i|x_{i-1})$ denote the transition distribution of a Markov chain, satisfying

$$\sum_{x_i\in\Omega} P_{x_{i-1},x_i} = 1, \qquad \forall x_{i-1} \in \Omega \; .$$

Furthermore, let $f(r_i - r_{i-1}|x_{i-1}, x_i)$ denote the density of the $i$th inter-observation time given that a vehicle was observed at sensor $x_{i-1}$ and then at $x_i$. We require that $f(\cdot|x_{i-1}, x_i)$ has support on $\mathbb{R}_+$ for all $x_{i-1}, x_i \in \Omega$.

Under a semi-Markov model, the likelihood of the observations $\{(x_i, r_i)\}_{i=1}^n$ is

$$\mathrm{p}(\{(x_i, r_i)\}_{i=1}^n) = \pi_{x_1} \prod_{i=2}^{n} P_{x_{i-1},x_i} f(r_i - r_{i-1}|x_{i-1}, x_i) \; .$$

To capture richer, more complicated traffic patterns, we modify the model on the sequence of sensors $x_1, \ldots, x_n$ which observe the vehicle to be a mixture of Markov chains. Let $M$ be a positive integer. For $m = 1, \ldots, M$, let $\pi_x^{(m)}$ denote the initial state distribution of the $m$th mixture component and let $P_{x_{i-1},x_i}^{(m)}$ denote the transition probabilities of the $m$th component. Also let $\theta^{(1)}, \ldots, \theta^{(M)}$ be the mixture parameters, satisfying $\theta_m \geq 0$ for all $m = 1, \ldots, M$ and $\sum_{m=1}^{M} \theta^{(m)} = 1$.

We associate a latent variable $m$ with each vehicle, taking values in the set $\{1, \ldots, M\}$, indicating which mixture component governs the vehicle's path. The trajectory of any

[3] If the inter-observation times were assumed to follow an exponential distribution then the semi-Markov process is equivalent to a continuous-time Markov chain. In general, the inter-observation times of a semi-Markov process may follow an arbitrary distribution with support on the positive real numbers.

particular vehicle is governed by only one component of the mixture model; i.e., each vehicle is a realization of this process and the particular component governing its trajectory is a multinomial random variable with parameters $\theta^{(1)}, \ldots, \theta^{(M)}$. Then the likelihood of the observations $\{(x_i, r_i)\}_{i=1}^n$ in the mixture model is given by

$$\begin{aligned} &\mathrm{p}(\{(x_i, r_i)\}_{i=1}^n) \\ &= \left( \sum_{m=1}^{M} \theta^{(m)} \pi_{x_1}^{(m)} \prod_{i=2}^{n} P_{x_{i-1},x_i}^{(m)} \right) \prod_{i=2}^{n} f(r_i - r_{i-1} | x_{i-1}, x_i) \; . \end{aligned} \tag{4.1}$$

Note that the mixture model only applies to the sequence of states, and the conditional distribution of the inter-observation times $r_i - r_{i-1}$ given the states $x_{i-1}$ and $x_i$ are independent of the mixture component $m$. In a transportation network this implies that the time to travel from $x_{i-1}$ to $x_i$ is independent of the process determining the route the vehicle is following.

In order to evaluate the likelihood (4.1) given observations $\{(x_i, r_i)\}_{i \geq 1}$ we need to specify the form of the inter-observation time density and we need to provide values for the parameters $\{\theta^{(m)}, \pi_x^{(m)}, P_{x,x'}^{(m)} \colon x, x' \in \Omega, m = 1, \ldots, M\}$ of the Markov chain mixture model. As mentioned above, the inter-observation time density $f(r_i - r_{i-1} | x_{i-1}, x_i)$ can be any density with support on the positive real numbers. Examples of potential choices include the truncated Gaussian, inverse-Gaussian, and gamma distributions. Each of these distributions has additional parameters which would need to be fit from data. In practice, we fit these parameters and the parameters of the Markov chain mixture model using data from a training period taken before the sequential hypothesis test for convoys goes online. We describe this training procedure in more detail in Section 4.4.

### 4.3.2 Notation for Observations of Two Vehicles

Recall that, in the convoy detection problem, we have two observation sequences $\{(x_i, r_i)\}_{i \geq 1}$ and $\{(y_i, s_i)\}_{i \geq 1}$ of the two vehicles, which we will refer to as $X$ and $Y$, where $x_i \in \Omega$ is the identifier of the sensor that observes vehicle $X$ at time $r_i$, and where $\Omega = \{1, \ldots, C\}$ denotes the collection of sensor indices. Also recall that the times $r_i$ and $s_i$ do not coincide; i.e., the observation times are not regularly sampled. Towards developing models and a sequential hypothesis test involving this data, we introduce notation to allow for simultaneously indexing the observations of both vehicles.

For a given time $t \in \mathbb{R}_+$, let

$$n_x(t) = \max\{i \colon r_i \leq t\}$$

denote the number of observations of vehicle $X$ that have been collected at time $t$, let

$$n_y(t) = \max\{i \colon s_i \leq t\}$$

denote the number of observations of vehicle $Y$ that have been collected at time $t$, and let

$$n(t) = n_x(t) + n_y(t)$$

denote the total number of observations of either vehicle that have been collected at time $t$. Let

$$T(t) = \{r_i\}_{i=1}^{n_x(t)} \cup \{s_i\}_{i=1}^{n_y(t)}$$

denote the set of all times when either vehicle is observed. Based on the assumption that no two observation events occur simultaneously, the cardinality of $T(t)$ is $n(t)$ and we can write

$$T(t) = \{t_0, t_1, \ldots, t_{n(t)-1}\},$$

where $t_k < t_{k+1}$, $k = 1, \ldots, n(t) - 1$; i.e., $T(t)$ can be viewed as the sequence of observation event times.

We assume that one of the two cases,

$$\{t_0 = r_1 \text{ and } t_1 = s_1\} \text{ or } \{t_0 = s_1 \text{ and } t_1 = r_1\} \ ,$$

holds; i.e., the test begins with one observation of each vehicle.

At each observation time $t_k$, exactly one vehicle is observed. It will be useful to define the extended observation sequences,

$$\begin{aligned} \mathbf{x}_k &= \left(\widetilde{x}_k, \widetilde{r}_k\right) \in \Omega \times \mathbb{R}_+ \\ \mathbf{y}_k &= \left(\widetilde{y}_k, \widetilde{s}_k\right) \in \Omega \times \mathbb{R}_+ \ , \end{aligned}$$

for $k = 1, \ldots, n(t) - 1$, where

$$\widetilde{x}_k = x_{n_x(t_k)} \quad \text{and} \quad \widetilde{r}_k = r_{n_x(t_k)}$$

are the sensor and time where vehicle $X$ was last seen as of observation time $t_k$, and

$$\widetilde{y}_k = y_{n_y(t_k)} \quad \text{and} \quad \widetilde{s}_k = s_{n_y(t_k)}$$

are the sensor and time where vehicle $Y$ was last seen as of observation time $t_k$. For example, $t_k$ is a time when vehicle $X$ is observed then $\widetilde{r}_k = t_k$, and $\widetilde{s}_k$ $(< \widetilde{r}_k)$ is the most recent time prior to $t_k$ when vehicle $Y$ is observed. We define the extended observation starting only from time $t_1$ (not $t_0$) so that both vehicles have been observed. Finally, let $\mathbf{x}_{1:n} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ denote the $X$-observation sequence at the first $n$ joint observation times, and let $\mathbf{y}_{1:n}$ be defined in a similar manner.

Note that there is an equivalence between the extended observation sequence $(\mathbf{x}_{1:n(t)-1}, \mathbf{y}_{1:n(t)-1})$ and the per-vehicle observation sequences, $\{(x_i, r_i)\}_{i=1}^{n_x(t)}$ and $\{(y_i, s_i)\}_{i=1}^{n_y(t)}$, in the sense that one can always construct the extended observation sequence given the per-vehicle observation sequences, and the per-vehicle sequences can be uniquely extracted from the extended observation sequence. Hence, the two representations convey precisely the same information.

### 4.3.3 Two-Vehicle Likelihood Factorization

We assume that under both of the hypotheses, $H_j$ with $j \in \{0, 1\}$, the joint likelihood of the extended observation sequences $\mathbf{x}_{1:k}$ and $\mathbf{y}_{1:k}$ is first-order Markov; i.e.,

$$\begin{aligned} &\mathrm{p}(\mathbf{x}_{1:n(t)-1}, \mathbf{y}_{1:n(t)-1} | H_j) \\ &\quad = \pi(\mathbf{x}_1, \mathbf{y}_1) \prod_{k=2}^{n(t)-1} \mathrm{p}(\mathbf{x}_k, \mathbf{y}_k | \mathbf{x}_{k-1}, \mathbf{y}_{k-1}, H_j) \; . \end{aligned} \tag{4.2}$$

This makes it possible to recursively calculate the log-likelihood ratio, simplifying the implementation of the sequential hypothesis test which is discussed further in Section 4.4. We also assume that the initial distribution $\pi(\mathbf{x}_1, \mathbf{y}_1)$ is independent of the hypothesis. In the following subsections we describe the proposed transition distribution $\mathrm{p}(\mathbf{x}_k, \mathbf{y}_k | \mathbf{x}_{k-1}, \mathbf{y}_{k-1}, H_j)$

under each hypothesis $j \in \{0, 1\}$.

To simplify the notation, in the sequel we write $\mathrm{p}_j(\mathbf{x}_k, \mathbf{y}_k | \mathbf{x}_{k-1}, \mathbf{y}_{k-1})$ for the transition dynamics under hypothesis $j \in \{0, 1\}$.

### 4.3.4 Model for Vehicles Traveling Independently ($H_0$)

The null hypothesis ($H_0$) states that the two vehicles are traveling through the network independent of each other. The likelihood of the observed paths of the two vehicles under this null hypothesis is simply the product of the two individual likelihoods from the previous section,

$$\begin{aligned} &\mathrm{p}_0(\mathbf{x}_{1:n(t)-1}, \mathbf{y}_{1:n(t)-1}) \\ &\quad = \mathrm{p}_0\left(\{(x_i, r_i)\}_{i=1}^{n_x(t)}, \{(y_i, s_i)\}_{i=1}^{n_y(t)}\right) \\ &\quad = \mathrm{p}\left(\{(x_i, r_i)\}_{i=1}^{n_x(t)}\right) \mathrm{p}\left(\{(y_i, s_i)\}_{i=1}^{n_y(t)}\right), \end{aligned}$$

where the individual likelihood of each vehicle is given by (4.1).

### 4.3.5 Markov Model for Convoys ($H_1$)

As discussed in the introduction, giving a precise definition of a convoy is not straightforward. We seek a method where the notion of a convoy encompasses the following elements:

1. At any point in time, one vehicle is following the other, and which vehicle is leading a convoy may change at any point in time.
2. The vehicles in a convoy need not take precisely the same route, but they should remain near each other (e.g., within a prescribed distance threshold).
3. The distance between the vehicles in a convoy is roughly proportional to the speed at which they are traveling, so if the vehicles were to follow exactly the same route then the time between consecutive observations of each vehicle at the same sensor would be roughly constant.

Initially a pair of vehicles is observed close together (possibly by the same camera or a nearby camera, and near in time) in order to trigger the initialization of a hypothesis test.

Then the subsequent observations of the pair can be used to update the likelihood of the convoy.

Consider two vehicles, $X$ and $Y$, moving through the network as a convoy. Initial observations for both vehicles will be set to the same likelihood under $H_1$ as under $H_0$. Specifically, we take the initial state distribution to be equal under both hypotheses. This is

$$\mathrm{p}_j(\mathbf{x}_1, \mathbf{y}_1) = \max_m \left\{\pi_{x_1}^{(m)} \pi_{y_1}^{(m)}\right\}, j \in \{0, 1\}$$

which selects the maximum likelihood mixture component for the initial distribution for the vehicles.

Under the convoy hypothesis, $H_1$, we model the sequence of states where the two vehicles are observed as being generated by the same mixture component in the mixture of Markov chain model. Given that this is mixture component $m$, the likelihood is given by

$$\begin{aligned}&\mathrm{p}_1(\mathbf{x}_{1:k}, \mathbf{y}_{1:k} | m)\\ &\quad = \pi_m(x_1)\pi_m(y_1) \times \prod_{i=2}^{n(t_k)-1} \mathrm{p}_1(\mathbf{x}_i, \mathbf{y}_i | \mathbf{x}_{i-1}, \mathbf{y}_{i-1}, m).\end{aligned}$$

Due to the assumption that exactly one observation is made at any time instant, at any observation time $t_k$ either the observation is of $X$, in which case $\widetilde{r}_k > \widetilde{s}_k$, or the observation is of $Y$, in which case $\widetilde{s}_k > \widetilde{r}_k$. Note that if the observation at time $t_k$ was of $X$ (respectively, of $Y$), then $\mathbf{y}_k = \mathbf{y}_{k-1}$ (respectively, $\mathbf{x}_k = \mathbf{x}_{k-1}$), and thus

$$\begin{aligned}&\mathrm{p}_1(\mathbf{x}_k, \mathbf{y}_k | \mathbf{x}_{k-1}, \mathbf{y}_{k-1}, m)\\ &\quad = \begin{cases} \mathrm{p}_1(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{y}_{k-1}, m) & \text{if } \widetilde{r}_k > \widetilde{s}_k \\ \mathrm{p}_1(\mathbf{y}_k | \mathbf{x}_{k-1}, \mathbf{y}_{k-1}, m) & \text{if } \widetilde{s}_k > \widetilde{r}_k, \end{cases}\end{aligned} \tag{4.3}$$

and so we must define the likelihood function for the two cases in (4.3).

Based on the description of a convoy given in Section 4.2, we desire a model where, under $H_1$, one vehicle will be leading and the other will be following at any point in time. If, for example, $X$ is leading, then $X$ transitions first and $Y$ transitions to a state afterwards which depends on $X$'s new location. We do not require that the same vehicle lead the entire time. The leader can switch shortly after $X$ and $Y$ have been observed near each other; the idea is that to switch between leading and following roles the follower must pass the leader.

Let $\mathrm{dist}(x, x')$ denote the geographic distance between two sensors $x, x' \in \Omega$, and let $L > 0$ be a given proximity threshold. To capture the different possible observation scenarios under this model, we split the description of the transition distribution (4.3) into six cases:

1. $X$ and $Y$ are close at time $t_{k-1}$ (no clear leader) and $X$ is observed next;
2. $X$ and $Y$ are close at time $t_{k-1}$ (no clear leader) and $Y$ is observed next;
3. $X$ is leading and $X$ is observed next;
4. $Y$ is leading and $Y$ is observed next;
5. $X$ is leading and $Y$ is observed next;
6. $Y$ is leading and $X$ is observed next.

Which case applies at a given point in time can be determined by examining the following three quantities:

i) The distance between the last observations of $X$ and $Y$, $\mathrm{dist}(\widetilde{x}_{k-1}, \widetilde{y}_{k-1})$, relative to the threshold $L$;

ii) Which vehicle was observed at time $t_{k-1}$,

$$\begin{cases} X & \text{if } \widetilde{r}_{k-1} > \widetilde{s}_{k-1}, \\ Y & \text{if } \widetilde{s}_{k-1} > \widetilde{r}_{k-1}; \end{cases}$$

iii) Which vehicle was observed at time $t_k$,

$$\begin{cases} X & \text{if } \widetilde{r}_{k} > \widetilde{s}_{k}, \\ Y & \text{if } \widetilde{s}_{k} > \widetilde{r}_{k}. \end{cases}$$

If $X$ and $Y$ were last observed close together (i.e., $\mathrm{dist}(\widetilde{x}_{k-1}, \widetilde{y}_{k-1}) < L$) then there is no clear leader, so the next vehicle to be observed may do so independently of the last observed location of $Y$. For example, if the vehicles were seen near each other at time $t_{k-1}$ (i.e., the distance between the two observing cameras is less than the proximity threshold $L$), then if the observation at time $t_k$ is of $X$ and the vehicles are following mixture component $m$ we take

$$\mathrm{p}_1(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_{k-1}, m) = P^{(m)}_{\widetilde{x}_{k-1},\widetilde{x}_k} f(\widetilde{r}_k - \widetilde{r}_{k-1}|\widetilde{x}_{k-1}, \widetilde{x}_k),$$

where $P^{(m)}_{x,x'}$ and $f(\cdot|x, x')$ denote the same transition and inter-observation time distributions used in the model under $H_0$.

If $\mathrm{dist}(\widetilde{x}_{k-1}, \widetilde{y}_{k-1}) \geq L$, then $X$ and $Y$ were not last seen close together, and one of the two vehicles is leading. If the previous observation was of $X$ (i.e., $\widetilde{r}_{k-1} > \widetilde{s}_{k-1}$) then $X$ is leading, and if the previous observation was of $Y$ then $Y$ is leading. In this case we further check whether the most recent observation, at time $t_k$, was of the leader or of the follower.

It can happen that the leader vehicle is observed multiple times between consecutive observations of the follower. For example, if $X$ is leading, $X$ and $Y$ are already separated by a distance larger than $L$, and $X$ is observed again, then $X$ is moving further away from $Y$. This is the case if $\mathrm{dist}(\widetilde{x}_{k-1}, \widetilde{y}_{k-1}) \geq L$ and $\widetilde{r}_{k-1} > \widetilde{s}_k$ (i.e., $X$ is observed again before $Y$, so we have at least two observations of $X$ since the last observation of $Y$). In this scenario we again model $X$'s transition as being independent of the last observation of $Y$,

$$\mathrm{p}_1(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_{k-1}, m) = P^{(m)}_{\widetilde{x}_{k-1},\widetilde{x}_k} f(\widetilde{r}_k - \widetilde{r}_{k-1}|\widetilde{x}_{k-1}, \widetilde{x}_k).$$

The model is similar if $Y$ is leading and it is observed multiple times between consecutive observations of $X$.

If $\mathrm{dist}(\widetilde{x}_{k-1}, \widetilde{y}_{k-1}) \geq L$ and the observation is of the follower, then we expect the location and time of the observation to depend on the last observation of the leader. Towards modeling dependence of the observed locations, we define

$$\delta_k = \frac{\mathrm{dist}(\widetilde{x}_{k-1}, \widetilde{y}_{k-1}) - \mathrm{dist}(\widetilde{x}_k, \widetilde{y}_k)}{\mathrm{dist}(\widetilde{x}_{k-1}, \widetilde{y}_{k-1})}. \tag{4.4}$$

Observe that $\delta_k$, which takes values in the interval $(-\infty, 1]$, measures the relative change in distance between the leader and follower at time $t_k$. If $\delta_k > 0$ then the follower was observed closer to the leader. We model the distribution over where the follower is observed using $\delta_k$ in the following manner. Suppose that, at time $t_k$, $X$ is leading and an observation is made of $Y$. Then

$$\mathrm{p}_1(\widetilde{y}_k|\widetilde{x}_{k-1}, \widetilde{y}_{k-1}, m) \propto \begin{cases} 1+\delta_k & \text{if } \delta_k > -1 \\ 0 & \text{otherwise,} \end{cases} \tag{4.5}$$

where[4] the constant of proportionality is chosen to ensure we have a valid distribution. Note that the transition distribution does not depend on the mixture component $m$ in this case. If $\delta_k \leq -1$ then the distance between the leader and follower has more than doubled since the last observation of the follower. This means that the leader and follower are travelling further apart from each other. In this case, the model above will set the likelihood of a convoy (hypothesis $H_1$) to zero, and the hypothesis test will declare that the pair of vehicles is not a convoy.

When there is a clear follower (i.e., the distance at time $t_{k-1}$ is greater than $L$) and the follower is observed, we also expect the inter-observation times of the leader and follower to be dependent. Suppose that $X$ is leading and, at time $t_k$, we observe $Y$. Consider the quantity $\widetilde{s}_k - \widetilde{r}_k$ which is strictly positive and gives the time between this observation of $Y$, the follower, and the last observation of $X$, the leader. We postulate that this distribution should be such that values of $\widetilde{s}_k - \widetilde{r}_k$ closer to zero are more indicative of the pair being a convoy. A simple way to capture this idea is to model the leader-follower inter-observation time as following the half-normal distribution with parameter $\sigma^2 > 0$,

$$f_{HN}(\widetilde{s}_k - \widetilde{r}_k) = \frac{\sqrt{2}}{\sqrt{\pi\sigma^2}} \exp\left( \frac{-(\widetilde{s}_k - \widetilde{r}_k)^2}{2\sigma^2} \right) \tag{4.6}$$

where $\widetilde{s}_k - \widetilde{r}_k > 0$.

To summarize, the forms of the transition distribution (4.3) for each of the six cases mentioned at the beginning of this subsection are shown in Table 4.1.

In the convoy model there are still $M$ mixture components in the terms involving the Markov transition matrices $P^{(m)}_{x,x'}$. Therefore the likelihood that a pair of vehicles are traveling as a convoy becomes

$$\mathrm{p}_1(\mathbf{x}_{1:k}, \mathbf{y}_{1:k}) = \max_m \left\{ \mathrm{p}_1(\mathbf{x}_{1:k}, \mathbf{y}_{1:k}|m) \right\}.$$

[4] Note that the above equation is valid even though $\delta_k$ in the right-hand side depends on $\widetilde{x}_k$, which appears to be missing from the arguments on the left-hand side. This is because, for the situation considered where the observation at time $t_k$ is of $Y$, we have $\widetilde{x}_k = \widetilde{x}_{k-1}$, so $\delta_k$ is still computable.

**Table 4.1** Value of the transition distribution $\mathrm{p}_1(\mathbf{x}_k, \mathbf{y}_k | \mathbf{x}_{k-1}, \mathbf{y}_{k-1}, m)$ for the different cases considered under $H_1$. Here $\mathbf{1}\{\cdot\}$ denotes the 0/1-valued indicator function, and the $\propto$ refers to the constant of proportionality from (4.5).

| Vehicle observed at $t_k$ | No Clear Leader ($\mathrm{dist}(\widetilde{x}_{k-1}, \widetilde{y}_{k-1}) < L$) | Clear Leader ($\mathrm{dist}(\widetilde{x}_{k-1}, \widetilde{y}_{k-1}) \geq L$) | |
|---|---|---|---|
| | | $X$ leading ($\widetilde{r}_{k-1} > \widetilde{s}_{k-1}$) | $Y$ leading ($\widetilde{s}_{k-1} > \widetilde{r}_{k-1}$) |
| $X$ ($\widetilde{r}_k > \widetilde{s}_k$) | $P^{(m)}_{\widetilde{x}_{k-1},\widetilde{x}_k} f(\widetilde{r}_k - \widetilde{r}_{k-1} \vert \widetilde{x}_{k-1}, \widetilde{x}_k)$ | $P^{(m)}_{\widetilde{x}_{k-1},\widetilde{x}_k} f(\widetilde{r}_k - \widetilde{r}_{k-1} \vert \widetilde{x}_{k-1}, \widetilde{x}_k)$ | $\propto (1 + \delta_k) f_{HN}(\widetilde{s}_k - \widetilde{r}_k) \mathbf{1}\{\delta_k > -1\}$ |
| $Y$ ($\widetilde{s}_k > \widetilde{r}_k$) | $P^{(m)}_{\widetilde{y}_{k-1},\widetilde{y}_k} f(\widetilde{s}_k - \widetilde{s}_{k-1} \vert \widetilde{y}_{k-1}, \widetilde{y}_k)$ | $\propto (1 + \delta_k) f_{HN}(\widetilde{s}_k - \widetilde{r}_k) \mathbf{1}\{\delta_k > -1\}$ | $P^{(m)}_{\widetilde{y}_{k-1},\widetilde{y}_k} f(\widetilde{s}_k - \widetilde{s}_{k-1} \vert \widetilde{y}_{k-1}, \widetilde{y}_k)$ |

This, as in the independent model, denotes the likelihood of a convoy as the highest likelihood of a convoy for any individual chain.

### 4.3.6 Convoy Example

Fig. 4.2 shows an example convoy scenario where two vehicles, $X$ and $Y$, transition through a network. The observations of each vehicle are shown in the table. In this example $X$ is leading from times 0 to 4, then $Y$ leads from times 7 to 10, and $X$ leads again from time 14 until the end of the example. The routes taken by the two vehicles are highly correlated but not identical. In addition, the vehicles are not always observed by exactly the same sensors. Thus the example illustrates some of the subtleties we aim to capture in our definition of a convoy.

## 4.4 Convoy Detection via Sequential Hypothesis Testing

Next we discuss our approach to detecting convoys in streams of license plate reads. We consider a typical sequential hypothesis testing setting [7] where the observations arrive successively at the fusion center, ordered by the times $r_i$ and $s_i$, and after receiving an observation the decision maker must choose from one of three options: 1) declare that the pair of vehicles is a convoy, 2) declare that the pair of vehicles is not a convoy, or 3) wait to receive additional observations. The aim is to make accurate decisions without deferring too long.

For the models described in the previous section, which involve mixtures of Markov chains, to perform testing in a sequential manner we use the sequential generalized likelihood ratio test [17]. The test statistic after $k+1$ total observations is

$$\Lambda(\mathbf{x}_{1:k}, \mathbf{y}_{1:k}) = \frac{\max\limits_{m} \{\mathrm{p}_1(\mathbf{x}_{1:k}, \mathbf{y}_{1:k}|m)\}}{\max\limits_{m} \{\mathrm{p}_0(\mathbf{x}_{1:k}, \mathbf{y}_{1:k}|m)\}}. \tag{4.7}$$

The test statistic can be updated in a recursive manner since the individual likelihoods $\mathrm{p}_0(\mathbf{x}_{1:k}, \mathbf{y}_{1:k}|m)$ and $\mathrm{p}_1(\mathbf{x}_{1:k}, \mathbf{y}_{1:k}|m)$ factorize according to (4.2). Thus, $M$ likelihood statistics need to be stored and updated for each hypothesis, $H_0$ and $H_1$.

Two decision thresholds, $\eta_0$ and $\eta_1$, are applied so that the decision after each update is given by the well-known rules:

$$\begin{array}{rclll} & & \Lambda(\mathbf{x}_{1:k}, \mathbf{y}_{1:k}) & < \eta_0 & \text{decide } H_0 \\ \eta_0 & \leq & \Lambda(\mathbf{x}_{1:k}, \mathbf{y}_{1:k}) & < \eta_1 & \text{decide ``need more data''} \\ \eta_1 & \leq & \Lambda(\mathbf{x}_{1:k}, \mathbf{y}_{1:k}) & & \text{decide } H_1. \end{array}$$

According to Wald [7], approximate decision regions for the sequential likelihood ratio test can be derived given specific performance criteria: the desired probability of false detection, $P_F \leq \alpha$, and the desired probability of detection, $P_D \geq \beta$, by taking

$$\eta_0 \geq \frac{1-\beta}{1-\alpha} \quad \text{and} \quad \eta_1 \leq \frac{\beta}{\alpha} \, . \tag{4.8}$$

Using these expressions, with equality, for $\eta_0$ and $\eta_1$ results in upper and lower bounds on $P_D$ and $P_F$. This can be used to set the desired performance limitations on the system. Normally in sequential hypothesis testing these bounds will be computed for i.i.d. samples of the two probability densities however the only requirement to achieve these bounds on the sequential test's performance are that the likelihood ratio be able to be decomposed into components which are only dependent on the current sample and the previous likelihood. In a Markov setting the "current" sample is a joint sample of the actual current sample and the previous sample. Therefore since this test can still be decomposed into individual components this analysis still holds.

To evaluate the likelihood models described in this section, parameters of the Markov chain mixture model need to be estimated or configured. These issues are discussed next.

### 4.4.1 Estimation of Markov Chain Mixture Model Parameters

In order to use a mixture of discrete Markov chains to more accurately describe the network, the model parameters must be estimated from training data. We use the Expectation Maximization (EM) algorithm [18] for this purpose. Previous work for estimation of a mixture of Markov chains using EM addressed the problem in the setting where each observation is an individual transition that may come from a different mixture component [19]. For the observations considered here, we assume that each vehicle's entire trajectory is associated with a single (latent) mixture component (rather than each observed transition of each vehicle potentially coming from a different mixture component). The number of mixture components can be determined using standard measures for goodness of fit in model order selection, such as the *Bayes Information Criterion* (BIC) [20].

### 4.4.2 Comments on the Leader-Follower Inter-Observation Time Distribution Under $H_1$

The parameter $\sigma^2$ of the half-normal distribution appearing in (4.6), used in the likelihood model under $H_1$, also needs to be specified. To consider a pair of vehicles to be driving as a convoy, one would like that the vehicles do not drift too far away from each other. We take $\sigma^2 = 30$ in the experiments, roughly corresponding to a maximum allowable time separation of 100 seconds between observations the leader and follower under $H_1$. To see this correspondence, note that integrating the half-normal pdf from 0 to 100 is close to 1 when $\sigma^2 = 30$.

### 4.4.3 Other System Parameters

For practical reasons, tracks of pairs of vehicles are only started when two vehicles are first seen close together in distance ($< L$) and in time. This threshold, which is also used in the statistical test, controls how far apart vehicles can drive in parallel routes while still being considered a convoy. It also controls how close together vehicles need to get in order to start the statistical test. We introduce two additional time threshold parameters, $T_s$ and $T_d$. The parameter $T_s$ is used to determine when to begin tracking a given pair of convoy vehicles (i.e., running the sequential hypothesis test for the given pair). A test is started if the vehicles are observed at locations at most a distance of $L$ apart within $T_s$ time units. The choice of $T_s$ will only control when tests start. A logical choice for this parameter might be related to the choice of the 95% confidence interval of the half-normal distribution. For example, $\sigma^2 = 30$ results in approximately 100 as the maximum value for the pdf of the half-normal in the 95% area. Therefore a logical choice to mimic the convoy sequential test might be 100 seconds. Setting this value very large would trigger the start of a lot of unnecessary tests, tracking pairs of vehicles, which would likely terminate after a few observations are made. The parameter $T_d$ is introduced for practical reasons, to also limit the number of consecutive sequential likelihood ratio tests being evaluated; if $T_d$ time units have elapsed and no new observation of either of the vehicles considered in a test has been received, then that track is terminated. This is the same as the track of the vehicles getting lost since they likely have travelled outside the field of view of the sensor network or at least one vehicle has parked and therefore will not be observed by the network.

## 4.5 Experimental Evaluation

### 4.5.1 Data Description

Next we study the performance of the proposed sequential hypothesis test using the models described in Section 4.4 against simulated data. A regional traffic assignment model for the Montreal metropolitan area is described in Sider et al. [21]. The model takes as an input the 2008 Origin-Destination (OD) trip data for the Montreal region provided by Montreal's *Agence Métropolitaine de Transport* and assigns it on the network using a stochastic assignment in the VISUM platform [22]. The regional network consists of 127,217 road links and 90,467 nodes associated with over 1500 traffic analysis zones. It also contains various road characteristics such as the type, length, speed limit, capacity, and number of lanes [21]. Note that this model has been validated using both traffic counts [23] and speed data collected using GPS [24].

Output from the traffic assignment simulations consists of an array that contains a detailed description of all paths connecting pairs of origin-destination zones for every hour of the day. Using this load information, we simulate a population of 2 million vehicles (roughly the number of registered vehicles in the greater Montreal region). These vehicles are sent randomly from zone to zone at random times during each hour along the paths from the Sider et al. [21] dataset, with the number of vehicles per path chosen to match the prescribed loads.

Sensors are placed at the 75 locations shown in Fig. 4.3(a). Each sensor records the identification number (license plate) of the vehicles as they pass by the sensors' locations. The data recorded by these sensors constitutes the baseline, normal traffic used in our experiments.

Two datasets were then simulated on this sensor network. Each simulation results in 24 hours of data and contains approximately 500,000 observed vehicles. The first of these two simulations was used for training, to fit the parameters of the mixture of Markov chains as well as the parameters to the distribution describing the time transitions. The training resulted in a two-component estimated mixture model. The second dataset was then used as a test dataset in which convoys of varying types were injected along with vehicles traveling independently. Performing a cursory analysis on each dataset we note that each vehicle is observed nine times, on average. This means that any detections which will occur only have access to a limited amount of data from each vehicle in the timespan the vehicle is present in the data. This dataset is the basis for the performance analysis reported later this section.

### 4.5.2 Estimated Transition Matrices

To fit the transition model parameters used in the simulations, multiple iterations of the EM algorithm were run while varying the number of mixture components in order to estimate the Markov transition matrices and initial distributions. The Bayesian Information Criteria (BIC) [20] was used for model order selection. More specifically, for each possible number of mixture components in the range $\{1, 2, \dots, 5\}$, the EM algorithm was executed from fifty different random initializations. We did not try more than 5 mixture components since we noted after multiple trials that the BIC for mixtures with more components got worse, rapidly. For this network, the model with the best BIC across all $50 \times 5$ random initializations is a mixture with 2 components. The two estimated transition matrices are visualized in Figs. 4.3(b) and 4.3(c). Each of the estimated components exhibits essentially the same behavior on the highway between exits. This is reasonable, since a vehicle traveling down the highway without a possible exit will continue traveling in the same direction. The differences in the transition matrices can be more aptly visualized on the side-roads off the highway. We can see that different traffic patterns are captured in these small offshoots from the highway.

### 4.5.3 Inter-Observation Time Distribution Under $H_0$

In addition to the Markov chain mixture model, the distribution governing the inter-observation times needs to be specified. As mentioned in Section 4.3.1, a valid distribution for inter-observation times should have support on $\mathbb{R}_+$. For this work we estimate a time transition based on the starting state using various exponential family models. Using the dataset described above, the normal distribution, inverse-Gaussian, and gamma distributions were fit to the data. Using the BIC as a measure of goodness, the heavy-tailed nature of the inverse-Gaussian distribution provided the best fit to the training data. Thus, we take $f(\tau|x, x')$, the likelihood that the time between two consecutive observations of a vehicle is $\tau$ time units given it was observed at sensor $x$ and then at sensor $x'$ (after $\tau$ time units), to be the inverse-Gaussian distribution,

$$\begin{aligned} &f_{IG}(\tau; \mu_{x,x'}, \lambda_x) \\ &\quad = \left[\frac{\lambda_x}{2\pi\tau^3}\right]^{1/2} \exp\left[\frac{-\lambda_x(\tau - \mu_{x,x'})^2}{2\mu_{x,x'}^2\tau}\right] \mathbf{1}\{\tau \geq 0\}, \end{aligned}$$

where $\mathbf{1}\{\cdot\}$ is the 0/1-valued indicator function, $\mu_{x,x'}$ is the mean time to transition from state $x$ to state $x'$, and $\lambda_x$ is the shape parameter associated with trajectories departing state $x$. When viewed as a generalized linear model [25], the inverse-Gaussian distribution has link function

$$\begin{aligned} \frac{1}{\mu_{x,x'}^2} &= \alpha_x + \text{dist}(x, x')\beta_x \\ \mu_{x,x'} &= \frac{1}{\sqrt{\alpha_x + \text{dist}(x, x')\beta_x}} \end{aligned}$$

where, now, $\alpha_x$, $\beta_x$, and $\lambda_x$ are the parameters to be estimated, and $\text{dist}(x, x')$ is the distance between states $x$ and $x'$. These parameters are estimated from the training data using Fisher scoring [25].

### 4.5.4 Simulating Convoys

The simulated dataset described in Section 4.5.1 is intended to represent normal background traffic. While we cannot guarantee there are no instances of convoys in this dataset, the appearance of any is unintentional. In order to evaluate the performance of the proposed sequential hypothesis testing approach, we inject convoys into the background data. Simulation of convoys involves determining two main factors: 1) the trajectories that will be taken by the vehicles, and 2) how the spacing between them will evolve over time. We consider two possibilities for each of these factors.

For the trajectories, in one case we simulate a convoy where the leader remains fixed for the entire trajectory and the follower takes exactly the same trajectory as the leader, where the leader's trajectory is sampled from one of the Markov chain mixture components. Alternatively, to allow for the leader and follower to take slightly different paths, we also simulate convoys where the follower's trajectory is sampled using the model described in Section 4.3.5, e.g., using (4.5).

To determine the timing between when the leader and follower are observed, we also consider two possibilities. In one case, the follower is always observed exactly one second after the leader. At a typical highway speed of 100 km/h, separation of 1 second corresponds to a distance of 27.8 meters between the vehicles, or 5–6 car lengths. Alternatively, we also simulate convoys where the follower's observation times are sampled from the half-normal distribution with parameter $\sigma^2 = 30$, following the model proposed in Section 4.3.5. The value $\sigma^2 = 30$ was chosen to allow an approximate maximum of 100 seconds of separation between vehicles in a convoy. If one solves the equation

$$1 = \int_{0}^{100} f_{HN}(y|\sigma^2)dy \tag{4.9}$$

for $\sigma^2$, one gets a value of approximately $\sigma^2 \approx 30$. This is a parameter to be chosen which allows for a target allowed maximum separation time between vehicles which the detection method will be sensitive to.

Taking all possible combinations of the two trajectory models and timing models described above leads to four ways in which convoys may be simulated. These four scenarios are summarized in Table 4.2, and all four are considered in the simulation results discussed below. Convoys of the varying types are simulated for approximately 18 observations (9 of each vehicle) and last anywhere from a few seconds to about 30 minutes, depending on the road segment they were randomly started on. Although we simulate such longer-lasting convoys, in Section 4.5.8 we study the average number of observations required by the sequential hypothesis testing procedure to make a decision to better understand how many observations are required and how this number depends on the performance criteria $P_F$ and $P_D$.

**Table 4.2** Simulated Convoy Configurations

| | Time separation between $X$ and $Y$ | Discrete Transition Model |
|---|---|---|
| Scenario 1 | Constant separation of 1 second | $X$ strictly followed by $Y$ |
| Scenario 2 | Constant separation of 1 second | $X$ and $Y$ following model in Section 4.3.5 |
| Scenario 3 | $\sim$ HalfNormal($\sigma^2 = 30s$) | $X$ strictly followed by $Y$ |
| Scenario 4 | $\sim$ HalfNormal($\sigma^2 = 30s$) | $X$ and $Y$ following model in Section 4.3.5 |

To simplify the presentation, for the rest of this section we only present and discuss results for convoys simulated according to Scenario 4. Results for the other three scenarios, which are included in the appendix, are qualitatively very similar.

### 4.5.5 Probability of Detection

To assess the probability of detection of the proposed sequential test, we simulate 1000 convoys for each of the four scenarios described in Table 4.2, and we evaluate empirical probability of detection as a function of the decision thresholds $\eta_0$ and $\eta_1$. Fig. 4.4 shows the probability of detection at the time of the first decision for Scenario 4. Varying the threshold $\eta_0$ has relatively little effect, especially for $\ln(\eta_0) < -5$. Setting the thresholds according to (4.8) with design criteria $\alpha = 0.0111$ and $\beta = 0.9999$ gives $\ln(\eta_0) \approx -9.20$ and $\ln(\eta_1) \approx 4.50$, for which the resulting probability of detection is $P_D = 0.9332$.

### 4.5.6 Probability of False Detection

We next simulate 1000 pairs of vehicles traveling through the road network independently. Each pair is simulated according to the same mixture component in the mixture of Markov chains and are sampled, as with the convoy case, for 18 observations (9 of each vehicle). The spacing of these observations in time depends on the random starting location and the network links traveled. We use these to study the probability of false detection for different values of $\ln \eta_0$ and $\ln \eta_1$. Fig. 4.5 shows the empirical probability of false detection as $\ln(\eta_0)$ and $\ln(\eta_1)$ are varied. As can be seen, the probability of false detection quickly drops to an almost negligible amount with a small increase in $\ln \eta_1$. Using the same decision bounds mentioned above, $\ln(\eta_0) = -9.20$ and $\ln(\eta_1) = 4.50$, the probability of false detection is $P_F = 0.0031$.

Fig. 4.6 shows a scatter plot of $P_D$ versus $P_F$, where each filled point corresponds to a particular choice of $\eta_0$ and $\eta_1$. The color of each point corresponds to the value of $\eta_0$. As is evident from the plot, as $\eta_0$ tends to $-\infty$, the probability of detection increases. One can also see subsets of points falling in roughly vertical groups. These correspond to the performance of the test when $\eta_1$ is held fixed and $\eta_0$ is varied, giving similar values of $P_F$ while varying $P_D$.

### 4.5.7 Comparison to a Simple Thresholding Approach

We compare the proposed method with a simple thresholding approach. A threshold is directly applied to the total number $n(t)$ of observations of a pair of vehicles, based on the intuition that the more often a pair of vehicles are observed near each other, the more likely they are to be a convoy. For a fair comparison, we apply the same system parameters as described in Section 4.4: to first consider a pair of vehicles as a potential convoy they need to be observed within a distance of $L$ from each other within $T_s$ time units, and to continue being considered as a potential convoy the pair must be observed for a subsequent $T_d$ time units afterwards.

The empirical detection probability and false alarm probability of the thresholding approach are also shown in Fig. 4.6 as red hollow circles. The threshold on $n(t)$ is varied from 2 to 40. (Note that $n(t)$ only takes values in the positive integers, so we only apply integer thresholds.) When a small threshold is used, the simple thresholding approach achieves a $P_D$ comparable to what can be achieved using the proposed approach, but with a very high probability of false detection (nearly 0.2). Increasing the threshold reduces both the probability of false detection and the probability of detection. In general, for very low probability of false detection, which is clearly desirable in applications, the proposed approach has a significantly higher $P_D$. Moreover, it is evident from Fig. 4.6 that the performance of the proposed approach is much less sensitive to the choice of threshold parameters $\eta_0$ and $\eta_1$.

### 4.5.8 Expected Number of Observations to Make a Decision

In addition to making accurate decisions (low $P_F$ and high $P_D$), it is important to understand how varying the decision thresholds of the sequential hypothesis test affects the number of observations required to make a decision. Figs. 4.7 and 4.8 show the expected number of observations ($n(t)$, the total number of observations of either vehicle) to make a decision under $H_1$ and $H_0$, respectively, as a function of the decision thresholds. A smaller value in this metric is better since it corresponds to a faster time to detect convoys under $H_1$, and a faster time to stop tracking non-convoy pairs under $H_0$. In a practical implementation, discarding non-convoy pairs quickly (without sacrificing accuracy in terms of $P_D$ and $P_F$) is desirable since the computational resources used by the sequential hypothesis test (both memory and CPU cycles) are proportional to the number of pairs of vehicles being tracked.

As $\ln(\eta_0) \to -\infty$ and $\ln(\eta_1) \to \infty$, the number of observations required to make a decision for $H_1$ increases. Focusing on the specific decision threshold values $\ln(\eta_0) = -9.20$ and $\ln(\eta_1) = 4.50$ mentioned before, Figs. 4.9 and 4.10 show histograms of the number of observations required to make a decision under $H_1$ and $H_0$, respectively. In both cases, decisions are made, on average, when roughly 10–12 total observations of the pair of vehicles are available (i.e., 5–6 observations of each vehicle).

## 4.6 Discussion

### 4.6.1 Regarding the Explicit Use of Road Network Data

The sequential detection approach adopted in this paper does not explicitly make use of knowledge of the road network topology. Instead, it is implicitly encoded in the transition matrices of the Markov chain mixture model. Such information could be used in the models, e.g., when calculating the distance $\text{dist}(x, x')$, if it is available. Tracking vehicles explicitly over a state space consisting of the entire road network would be computationally cumbersome in a large system (which may observe on the order of tens of thousands of vehicles per hour), and a system making use of detailed road maps would also require updating of the maps when segments are closed (e.g., for construction) or changed (e.g., re-zoning). On the other hand, the proposed approach implicitly models traffic patterns using the Markov chain mixture model. The parameters of this model can be estimated directly from the data, and so no additional input or tuning is required.

### 4.6.2 Detecting Convoys of More Than Two Vehicles

In order to detect if a group of vehicles larger than two are traveling as a convoy a simple post-analysis can be performed. In order to understand this post-analysis for groups of convoys, consider a target vehicle $X$ and suppose that we detect $N$ other vehicles as traveling in a convoy with $X$ at a specific time. We then simply look at these $N$ vehicles which were detected as in a convoy with $X$ and look if they were also detected as being in a convoy with each other. This creates a set of vehicles where all the pairwise combinations are detected to be in a convoy in a set timeframe. This is a detected convoy "group". We note that the approach just described can be related to the notion of density-connected sets used in [12].

One may be tempted to view the problem of detecting convoys of more than two vehicles as a sort-of graph partitioning or community detection problem, with vertices in the graph corresponding to vehicles and edges placed between two nodes that belong to a convoy. The pairwise test presented in this paper identifies where there are likely edges, and one would hope that a convoy of two or more vehicles would give rise to dense connections between the vehicles in the convoy. However this is not necessarily the case since convoys may be formed by long lines of vehicles (e.g., along a single-lane road). For example, if three vehicles, $X - Y - Z$, form a convoy our test may not detect the correlation between $X$ and $Z$ directly if they are too far apart. This presents one of the main challenges we anticipate with detecting convoys of more than two vehicles. We leave a more detailed study and in-depth analysis of detecting larger convoy groups to future work.

### 4.6.3 Using Different Estimated Network Properties for Different Times of the Day

Some other issues which might arise in practice are such things as accidents or road closures as well as how traffic patterns behave differently throughout the day (e.g. rush hour). All of these real-world issues will cause traffic to behave differently from the network which was originally trained on. The problem of random events such as accidents and road closures is difficult to handle due to the unpredictable nature of it. This will likely cause more anomalies (such as convoys) to be flagged in the algorithm due to more vehicles taking a lower-likelihood route.

However the case of a varying traffic pattern throughout the day is one which is much more simple to mitigate. By simply swapping out the transition matrices as well as the properties for the inverse-Gaussian distributions and the initial distributions, one can in real-time update the detector for more realistic traffic patterns. This could be done, say, every hour to mimic changing traffic patterns throughout the day. This would not change the algorithm's design since it would say simply for a specific tracked pair of vehicles "the first $n$ samples came from the 1 a.m. to 2 a.m. mixture while the next $m$ samples came from the 2 a.m. to 3 a.m. mixture". This allows the algorithm to handle even a continuous-time distribution for the underlying mixture of Markov chains. An analytic solution will become much more difficult due to the addition of many additional chains to estimate (possibly infinite in the continuous-time mixture case), however it might be able to drastically improve the detection and false detection performance by more accurately measuring the nominal traffic distribution.

## 4.7 Conclusion

This paper proposes a novel approach to detecting convoys in urban environments. Typically long-range sensors are not applicable in urban environments. This means that only by using short-range sensors such as LPR can one do many types of road network analysis including convoy detection. The algorithm presented only uses a small amount of information about the detected vehicles to perform convoy detection which is an added benefit for minimizing the computational complexity. It is also capable of detecting convoys in real time as data arrives.

In the problem formulation of this work we assumed that measurements are exact; there are no mis-read license plates and no missed license plate reads. Our future work will address the case of missing and noisy data using a hierarchical Bayesian approach by adding one layer, so that the vehicle trajectory model becomes a hidden mixture of Markov chains.

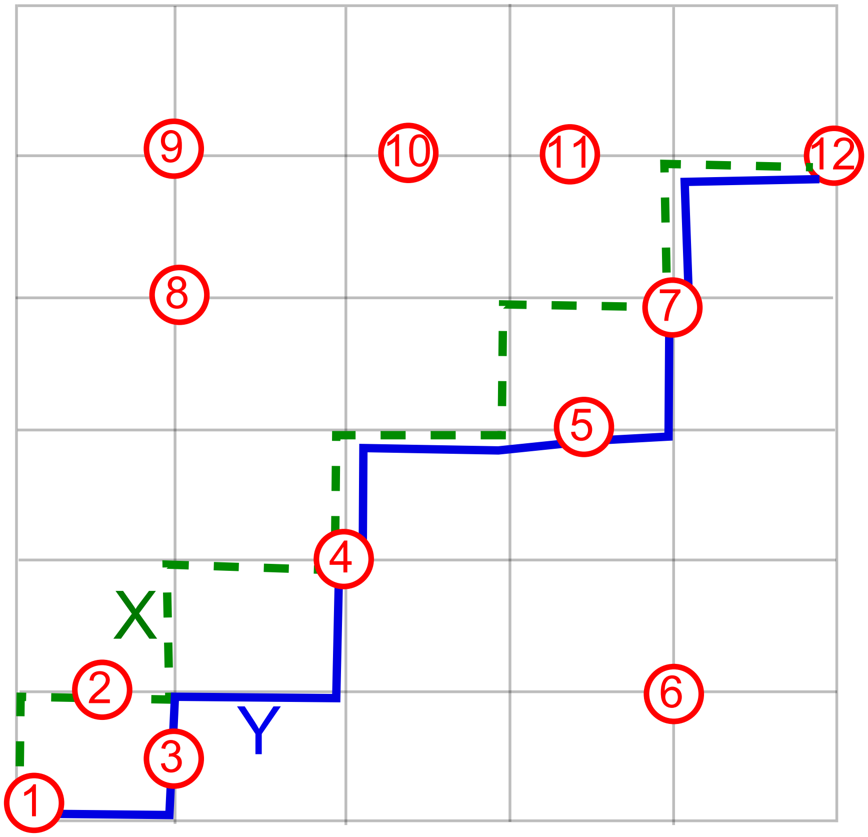
This paper focused on detecting convoys of vehicles in a road network. Individual vehicles were modeled as moving along paths in the network according to a first-order Markov model, and convoys are two or more vehicles whose paths are correlated in space and time. An interesting extension of this approach would be to detect when two or more epidemics spreading over a network are correlated. First-order Markov models are also commonly used to model epidemics spreading over networks, but the resulting patterns are trees rather than paths. In future work it would be interesting to explore extensions of the sequential hypothesis testing framework considered in this paper for detecting correlated epidemics.

## Appendix

Fig. 4.11(a) shows $P_D$ as a function of the decision thresholds when convoys are simulated using Scenario 1. This situation is where a vehicle $X$ moves independently through the network while vehicle $Y$ follows exactly the same path as $X$ with a 1-second lag. It can be seen here that the detection accuracy degrades quickly with the increase of the $\ln \eta_1$. This appears to no longer be the case in the next scenario, Scenario 2, as shown in Fig. 4.11(b) where there is still a constant 1-second time separation but the transitions of $Y$ are following the convoy model from Section 4.3.5. This is because in scenario following the model from Section 4.3.5 one vehicle can, in many circumstances, take an alternate, lower likelihood, path which is close to the leader so the likelihood of $H_0$ drops faster than the likelihood of $H_1$. For example, consider two vehicles traveling on parallel paths where one vehicle is on a high-likelihood path (such as a highway) and another is on a lower-likelihood path (such as a service road parallel to the highway). In this case the vehicle on the lower likelihood path will make the likelihood of $H_0$ lower faster than the likelihood of $H_1$ decreases.

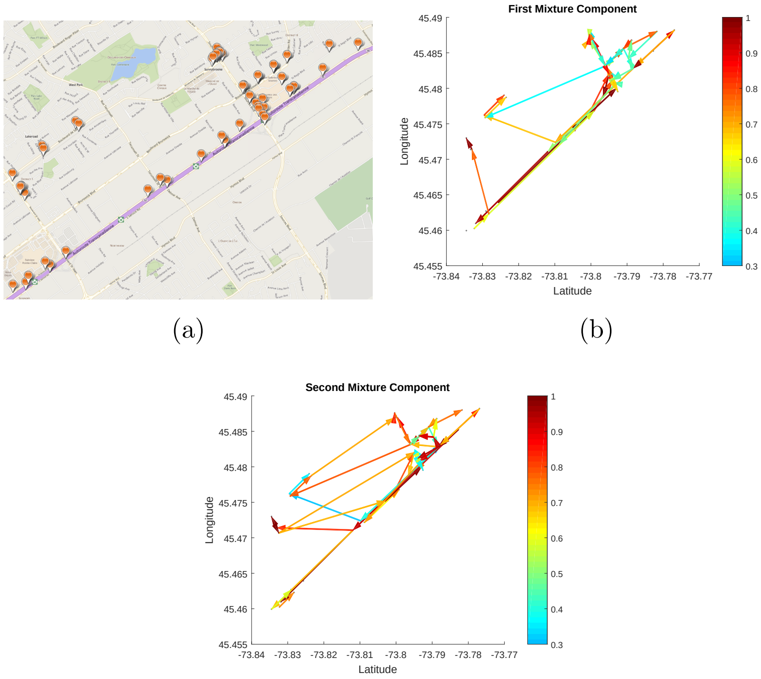
One can also see that the mitigation of the fast drop in the shape of the surface in Fig. 4.11(a) can be likely attributed to the constant time separation of 1 second as in Figs. 4.11(c) and 4.4. Here the exponential drop in the probability of detection with the increase of $\ln(\eta_1)$ appears to become at worst a linear relationship. This means that allowing a floating leader along with a variable distance between vehicles increases our detection ability. This is very good news since a constant separation between vehicles of 1 second throughout an entire observation sequence is very unlikely.

Figs. 4.12(a), 4.12(b), and 4.12(c) show the average number of observations required to make a decision under $H_1$ when convoys are simulated according to Scenario 1, 2, and 3, respectively. These figures exhibit a similar trend to that presented in Fig. 4.7. The main difference is in terms of the rate at which the average number of decisions plateaus with changes of $\ln \eta_1$.



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{x_i}$ | 1 | | 2 | | | 4 | | 7 | | 12 | |
| $\mathbf{r_i}$ | 0 | | 3 | | | 8 | | 14 | | 21 | |
| $\mathbf{y_i}$ | | 1 | | 3 | 4 | | 5 | | 7 | | 15 |
| $\mathbf{s_i}$ | | 1 | | 4 | 7 | | 10 | | 15 | | 22 |

**Fig. 4.2** Example of a convoy of two vehicles ($X$ and $Y$) on a simple grid network. The figure shows the trajectories of each vehicle along the locations of 12 sensors. The table shows the observations (sensor index and observation time) made of both vehicles, spaced so as to help illustrate the sequence of observations over time.



(a)

(b)

(c)

**Fig. 4.3** (a) Locations of the 75 simulated sensors along a stretch of Highway 40 in Montreal, Canada. Along this stretch there are two exits, one near the top-right corner (where sensors are shown perpendicular to the highway) and the other near the bottom-left corner of the figure. Sensors are also located on the feeder roads that run along side of the highway. The image appears to have less than 75 sensors since many of the points represent 2 sensors (one pointed in each direction). This is necessary since LPR sensors require that they be monitoring a specified direction so to cover a bi-directional road two sensors are necessary. A Markov chain mixture model is fit to simulated traffic from a 24-hour training period, and it is determined that a two-component mixture model provides the best fit as measured using the BIC. The transition matrices of these two mixture components are shown in panels (b) and (c). It can be seen that, while mixture components capture the flow of traffic along the highway, they capture distinctly different trends in terms of traffic entering/exiting the highway, and off of the highway.

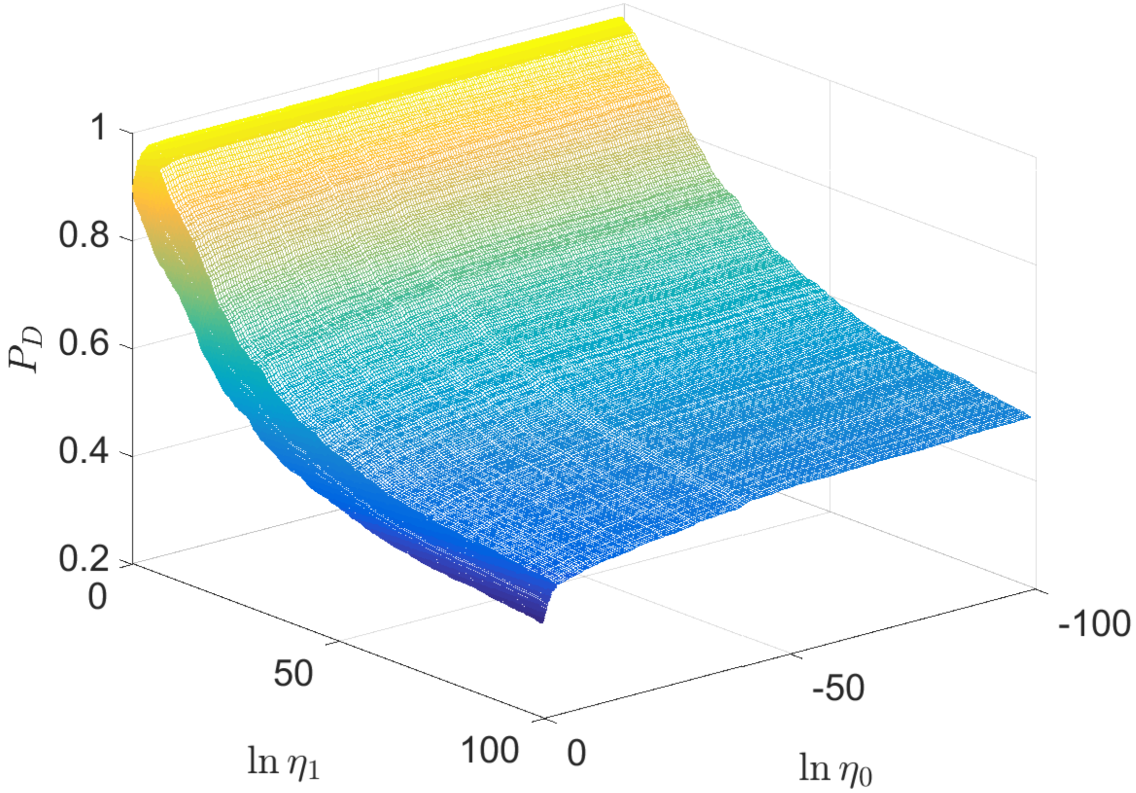


**Fig. 4.4** Probability of detection for varying decision boundaries $\eta_0$ and $\eta_1$ with convoys simulated by Scenario 4.

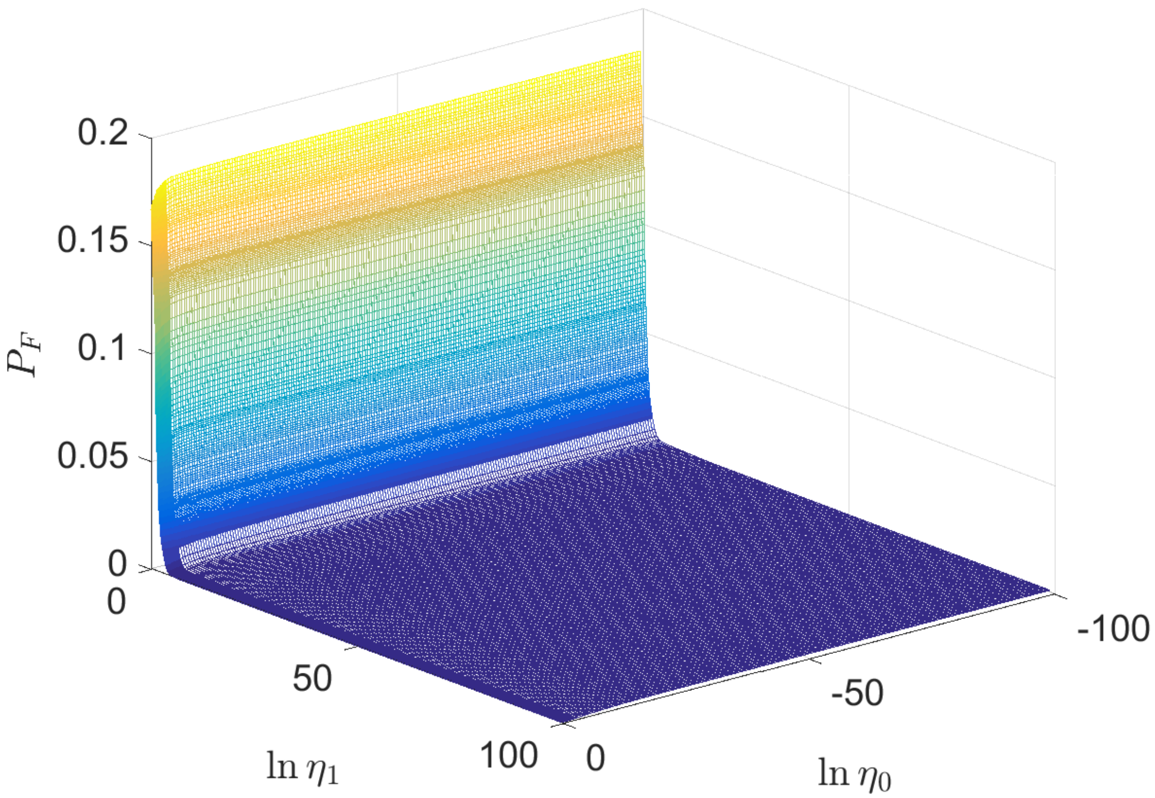


**Fig. 4.5** Probability of false detection for varying decision boundaries $\eta_0$ and $\eta_1$ for vehicles simulated following the independent model

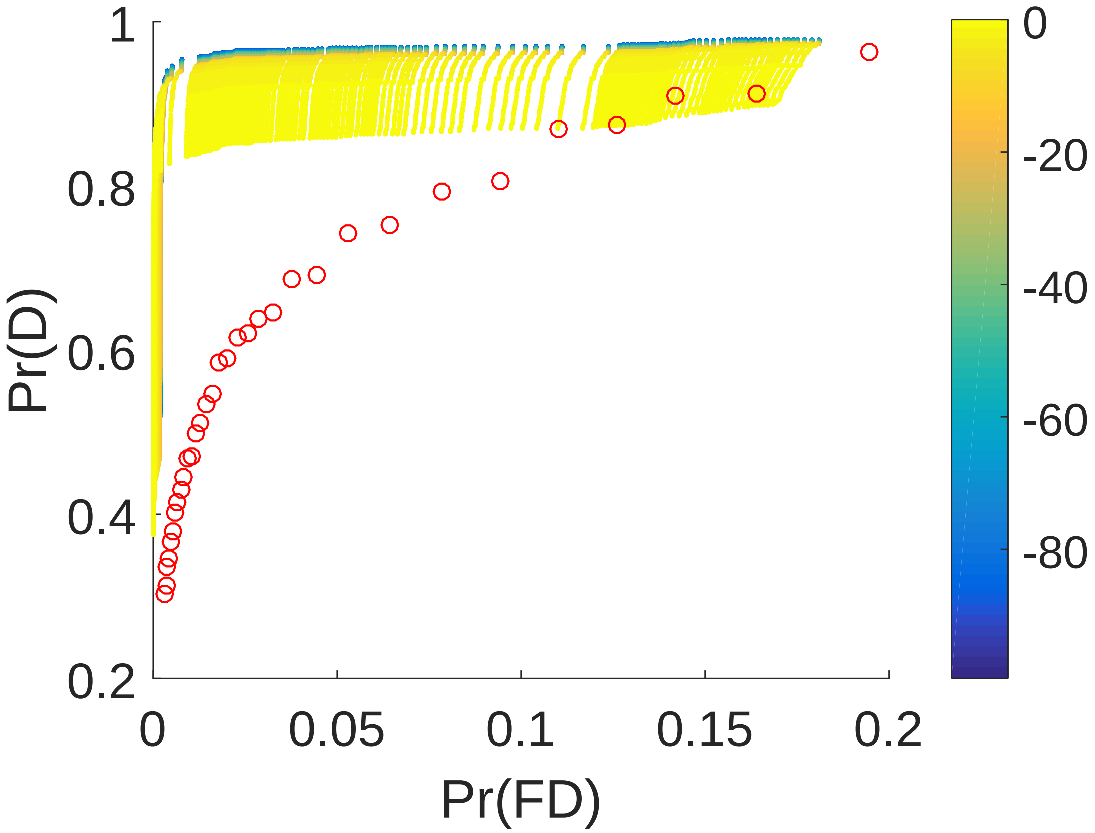


**Fig. 4.6** Scatter plot of the resulting probability of false detection values versus the probability of detection values for all combinations of $\ln(\eta_0)$ and $\ln(\eta_1)$ where convoys are simulated using the convoy model described in Section 4.3.5. The vertical coloring denotes changes in $\eta_0$. Note that the horizontal axis $(P_F)$ ranges from 0 to 0.2. Overlayed in red are the probability of detection and false detection rates from the thresholding approach.

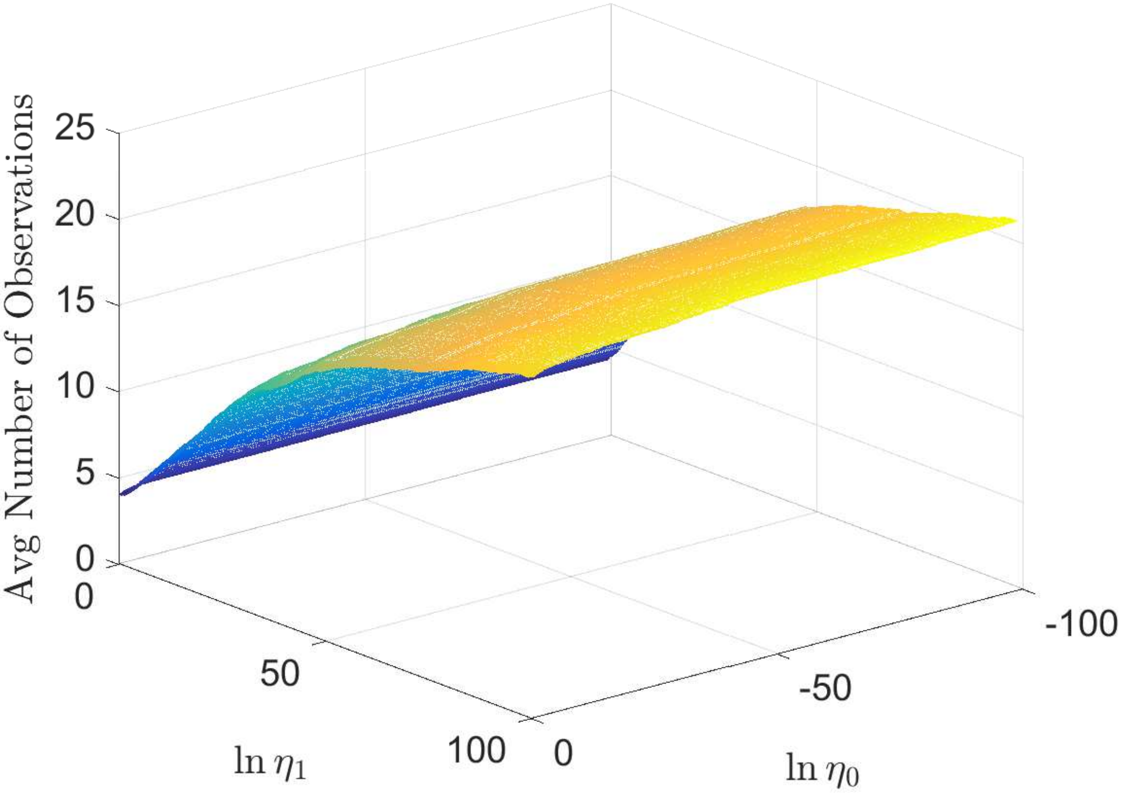


**Fig. 4.7** Expected number of observations to make a decision under $H_1$ for varying decision boundaries $\eta_0$ and $\eta_1$ with convoys simulated by Scenario 4.

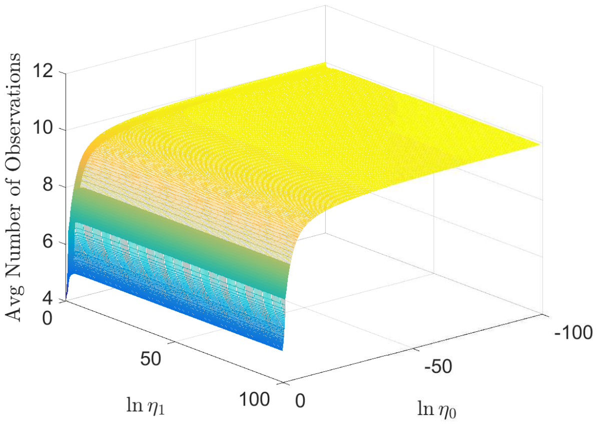


**Fig. 4.8** Expected number of observations to make a decision under $H_0$ for varying decision boundaries $\eta_0$ and $\eta_1$ where vehicles are simulated independent of each other.

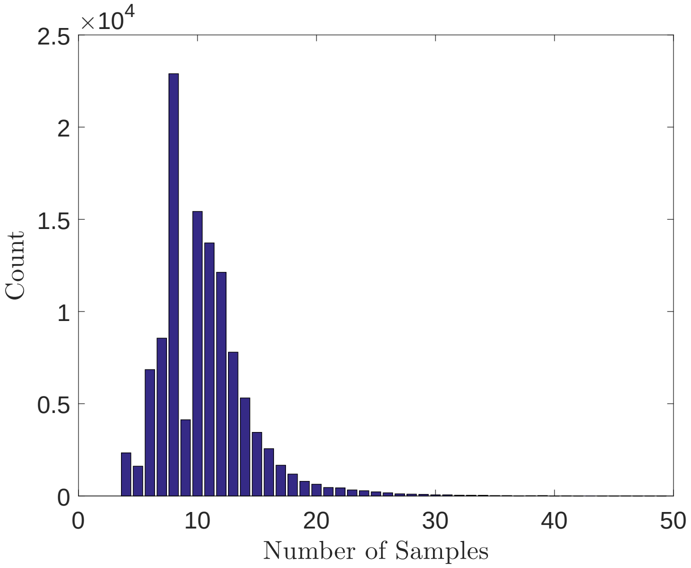


**Fig. 4.9** Histogram of the number of samples to make a decision under the alternate hypothesis ($H_1$) where convoys were simulated with half-normally distributed time separation and following the discrete convoy model in Section 4.3.5.

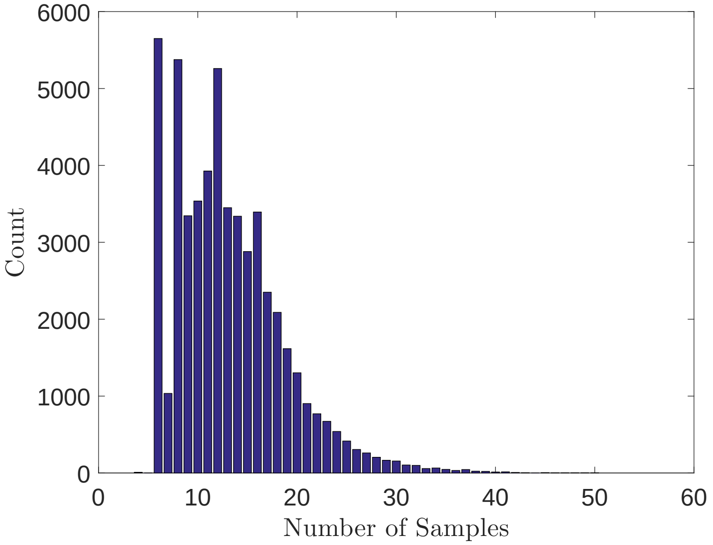
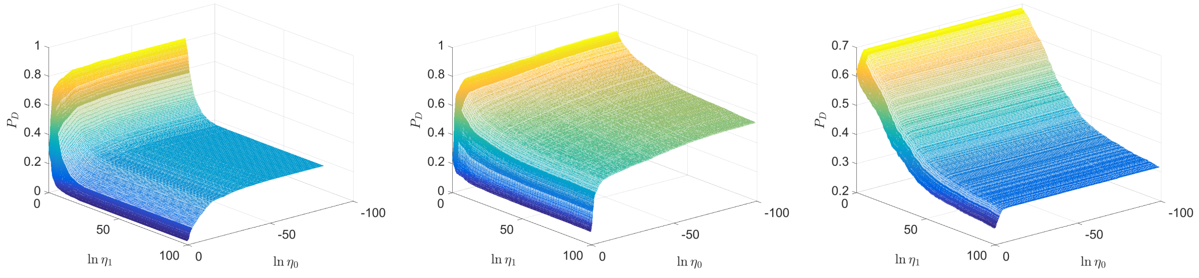


**Fig. 4.10** Histogram of the number of samples to make a decision under the independent hypothesis ($H_0$) where vehicles are traveling independently.

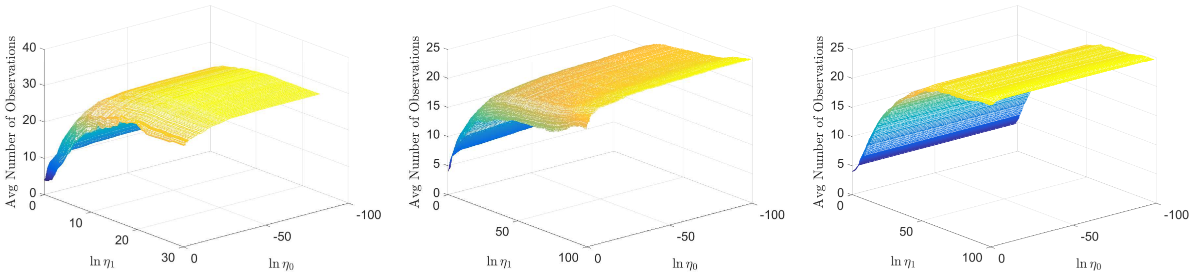


(a)

(b)

(c)

**Fig. 4.11** Probability of detection for varying decision boundaries $\eta_0$ and $\eta_1$ with convoys simulated by (a) Scenario 1, (b) Scenario 2, and (c) Scenario 3.



(a) (b) (c)

**Fig. 4.12** Expected number of observations to make a decision under $H_1$ for varying decision boundaries $\eta_0$ and $\eta_1$ with convoys simulated by (a) Scenario 1, (b) Scenario 2, and (c) Scenario 3.

# References for *Convoy Detection Using License Plate Recognition Sensors*

[1] A. Homayounfar, A. Ho, N. Zhu, G. Head, and P. Palmer, "Multi-vehicle convoy analysis based on ANPR data," in *Intl. Conf. on Imaging for Crime Detection and Prevention*, Nov. 2011, pp. 1–5.

[2] S. van de Hoef, K. Johansson, and D. Dimarogonas, "Fuel-optimal centralized coordination of truck-platooning based on shortest paths," in *American Control Conf.*, Chicago, IL, Jul. 2015.

[3] A. Alam, B. Besselink, V. Turri, J. Mårtensson, and K. Johansson, "Heavy-duty vehicle platooning for sustainable freight transportation: A cooperative method to enhance safety and efficiency," *IEEE Cont. Sys. Mag.*, vol. 35, no. 6, pp. 34–56, Dec. 2015.

[4] M. Thottan and C. Ji, "Anomaly detection in IP networks," *IEEE Trans. on Sig. Proc.*, vol. 51, no. 8, pp. 2191–2204, Aug. 2003.

[5] A. Hero, "Geometric entropy minimization (GEM) for anomaly detection and localization," in *Conf. on Neural Information Processing Systems*, Vancouver, Canada, Dec. 2010.

[6] C. Scott and E. Kolaczyk, "Nonparametric assessment of contamination in multivariate data using generalized quantile sets and FDR," *J. of Computational and Graphical Stat.*, vol. 19, no. 2, pp. 439–456, Jun. 2010.

[7] A. Wald, *Sequential Analysis*, ser. Wiley Publication In Statistics, R. A. Bradley, J. S. Hunter, D. G. Kendall, and G. S. Watson, Eds. John Wiley & Sons, Inc., 1966.

[8] W. Koch, “Information fusion aspects related to GTMI convoy tracking,” in *Fifth Int. Conf. on Information Fusion*, vol. 2, 2002, pp. 1038–1045.

[9] E. Pollard, B. Pannetier, and M. Rombaut, “Convoy detection processing by using the hybrid algorithm (GMCPHD/VS-IMMC-MHT) and dynamic Bayesian networks,” in *Int. Conf. on Information Fusion*, vol. 12, Seattle, WA, USA, July 2009.

[10] E. Pollard, M. Rombaut, and B. Pannetier, “Bayesian networks vs. evidential networks: An application to convoy detection,” in *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Dortmund, Germany, Jun. 2010.

[11] C. S. Jensen, D. Lin, and B. C. Ooi, “Continuous clustering of moving objects,” *IEEE Trans. on Knowledge and Data Eng.*, vol. 19, no. 9, pp. 1161–1174, Sept 2007.

[12] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen, “Discovery of convoys in trajectory databases,” in *Intl. Conf. on Very Large Data Bases*, Auckland, New Zeland, Aug. 2008, pp. 1068–1080.

[13] P. Kalnis, N. Mamoulis, and S. Bakiras, “On discovering moving clusters in spatio-temporal data,” in *Intl. Symp. on Spatial and Temporal Databases*, Angra dos Reis, Brazil, Aug. 2005, pp. 364–381.

[14] T. Weiherer, E. Bouzouraa, and U. Hofmann, “A generic map based environment representation for driver assistance systems applied to detect convoy tracks,” in *IEEE Intl. Conf. on Intelligent Transportation Systems*, Anchorage, AK, Sep. 2012.

[15] J. Yeoman and M. Duckham, “Decentralized network neighborhood information collation and distribution for convoy detection,” in *Seventh Int. Conf. on Geographic Information Science*, Columbus, OH, September 2012.

[16] R. A. Howard, *Dynamic Probabilistic Systems : Semi-Markov and Decision Processes*, 1st ed. Dover Publications, 2007.

[17] M.-C. Shih, T. L. Lai, J. F. Heyse, and J. Chen, “Sequential generalized likelihood ratio tests for vaccine safety evaluation,” *Stat. in Medicine*, vol. 29, no. 26, pp. 2698–2708, November 2010.

[18] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. of the Royal Statatistics Society Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[19] T. J. Perkins, “Maximum likelihood trajectories for continuous-time Markov chains,” in *Advances in Neural Information Processing Systems 22*, 2009, pp. 1437–1445.

[20] S. T. Buckland, K. P. Burnham, and N. H. Augustin, “Model selection: An integral part of inference,” *Biometrics*, vol. 53, no. 2, pp. 603–618, 1997.

[21] T. Sider, A. Alam, M. Zukari, H. Dugum, N. Goldstein, N. Eluru, and M. Hatzopoulou, “Land-use and socio-economics as determinants of traffic emissions and individual exposure to air pollution,” *J. of Transport Geography*, vol. 33, no. 0, pp. 230 – 239, 2013.

[22] PTV Vision, *PTV Vision, 2009*, VISUM 11.0 Basics ed., PTV AG, Karlsruhe, Germany, 2009.

[23] T. Sider, A. Alam, W. Farrell, M. Hatzopoulou, and N. Eluru, “Evaluating vehicular emissions with an integrated mesoscopic and microscopic traffic simulation,” *Canadian J. of Civil Eng.*, vol. 41, no. 10, pp. 856–868, Aug. 2014.

[24] A. Alam, G. Ghafghazi, and M. Hatzopoulou, “Traffic emissions and air quality near roads in dense urban neighborhoods: Using microscopic simulation for evaluating effects of vehicle fleet, travel demand, and road network changes,” *J. of the Transportation Research Board*, vol. 2427, pp. 83–92, 2014.

[25] A. Agresti, *Categorical Data Analysis*, ser. Wiley Series in Probability and Statistics. Wiley, 2013.

# Chapter 5

# Time-Varying Mixtures of Markov Chains

The paper presented in Chapter 4 proposes a novel way of modelling vehicles travelling as a convoy through a network of AVI (specifically LPR) sensors. This raises the question of how does one improve the accuracy of the detection of convoys in a mixture of Markov chains. With little investigation, traffic patterns can be noted to be very volatile in nature, meaning they vary very rapidly. Therefore the model utilized in Chapter 4, which assumes a static mixture of Markov chains governs traffic for all time, is likely not expressive enough.

The following paper proposed extending the model for nominal traffic to vary the parameters and model order of the mixture with time. We then organize observations into time-windows and assume the model is stationary within a time-window. This allows us to specify mixture models within a time-window and therefore they only need to be valid for the duration of the window. In addition, we impose a Markov dependence between time-windows which allows us to exploit this in the definition and estimation of our model and its parameters.

Chapter 6 proposes a generative model for the evolution of the mixture model between time-windows. This proves to be a flexible and expressive model for data which can be received in a streaming fashion and can be organized into time-windows. However the proposition of such a complex, high-dimensional model for streaming data requires a method to estimate the parameters of this model. Normally MCMC methods would be applied to a generative model, however we propose an alternative, deterministic, approximate estimation method which proves to yield reasonable estimates of the true underlying model in simulated data as well as being applicable to predicting future vehicle transitions in real-data.

In Chapter 4, the model for nominal traffic included estimates for the distribution of the transition times between states. The model in the following manuscript, only extends the mixture of discrete-time Markov chains which governs the transition path vehicles follow. In Chapter 4, we proposed the assumption that the transition times between states in the Markov chain are independent of the overall path being followed by a vehicle. This assumption is carried forward in the following work and therefore we do not address the estimation of transition times in the following manuscript.

# Chapter 6

# Paper: Time-Varying Mixtures of Markov Chains: An Application to Traffic Modeling

[1]Time-varying mixture models are useful for representing complex, dynamic distributions. Components in the mixture model can appear and disappear, and persisting components can evolve. This allows great flexibility in streaming data applications where the model can be adjusted as new data arrives. Fitting a mixture model with computational guarantees which can meet real-time requirements is challenging with existing algorithms, especially when the model order can vary with time. Existing approximate inference methods may require multiple restarts to search for a good local solution. Monte-Carlo methods can be used to jointly estimate the model order and model parameters, but when the distribution of each mixand has a high-dimensional parameter space, they suffer from the curse of dimensionality and from slow convergence. This paper proposes a generative model for time-varying mixture models, tailored for mixtures of discrete-time Markov chains. A novel, deterministic inference procedure is introduced and is shown to be suitable for applications requiring real-time estimation, and the method is guaranteed to converge at each time step. As a motivating application, we model and predict traffic patterns in a transportation network. Experiments illustrate the performance of the scheme and offer insights regarding

[1] The contents of this chapter are based on the article: S. Lawlor and M. Rabbat, “Time-varying mixtures of Markov chains: An application to traffic modeling,” submitted, August 2016

tuning of the algorithm parameters. The experiments also investigate the predictive power of the proposed model compared to less complex models and demonstrate the superiority of the mixture model approach for prediction of traffic routes in real data.

## 6.1 Introduction

Mixture models are a useful tool for modelling complex distributions. Allowing a mixture model to be dynamic, with a varying number of components as well as time-varying mixand parameters, allows for a very flexible model which can be applied to many forms of streaming data. In this paper we develop a procedure for fitting time-varying mixtures of discrete-time finite-state Markov chains.

We are motivated by the application of traffic modelling using streams of automatic vehicle identification data [1]. A model of traffic patterns can be used for a variety of traffic analysis applications [2]. Potential applications include vehicle path prediction, visualization of the difference in traffic flows between varying time-spans, and other forms of statistical detection and estimation such as convoy detection [3]. Models for traffic routes, and traffic flow, are commonly used in law enforcement, city planning, private for-profit parking industry, and other applications [4].

### 6.1.1 Automatic Vehicle Identification Data

Traditionally, road traffic is measured and monitored using sensors such as inductive loops embedded in the roadway that provide counts of the number of vehicles passing by a particular location. Increasingly, traffic can also be measured and estimated using data from *automatic vehicle identification* (AVI) sensors, such as bluetooth sensors or cameras performing license plate recognition. Bluetooth sensors record the MAC address of the in-vehicle entertainment/communications system. License plate recognition cameras recognize and record the license plate of vehicles which pass in their field of view. Sensors that read the highway toll-pass ID of an equipped vehicle when it passes nearby (such as E-ZPass in northeastern U.S.A. [5]) provide similar information. While each of these three types of sensors has different strengths and weaknesses (e.g., reliability, detection rate), in all cases the resulting data can be leveraged in the same fashion for traffic modelling. Each AVI observation includes the ID of the vehicle observed, the time, and the location of the observation.

In typical monitoring systems, AVI sensors report their data in a streaming manner to a fusion center. Since observations are made in an event-driven manner (with events corresponding to a vehicle passing near a sensor), the sequence of observations of a particular vehicle at different sensors can be viewed as an irregularly sampled time series. Since vehicles may come and go freely within the coverage area of the sensor network, the observation sequence of a vehicle can start or stop at any point and the number of observations (the length of the observation sequence) of a vehicle is variable.

This paper considers the case where a region has been instrumented with a collection of static AVI sensors. The sensors are treated as the states of a Markov process, and the trajectories of individual vehicles are modeled as realizations of this process. The observation times are only used to order the sequence of locations where a vehicle is observed, leading to a discrete-time model for the sequence of AVI sensors.

### 6.1.2 Previous Work

#### Time-Varying Mixture Models

In this paper we propose to model a network of AVI sensors as a time-varying mixture of discrete-time Markov chains. Previous work on parameter estimation in time-varying mixture models typically adopts a Bayesian non-parametric perspective and focuses on mixtures of Dirichlet processes [6]. However these works perform inference using Monte-Carlo based methods, where computational performance guarantees are difficult to achieve. Stephens [7] proposes a Bayesian representation of a mixture model using a marked point-process. However MCMC-based inference methods have challenges in high-dimensional parameter spaces (e.g, the entries of a transition matrix), especially in the streaming setting. Work on deterministic inference methods of these types of models, where computational timing guarantees are more feasible, is limited. Also much of the previous work on time-varying mixture models focuses on mixtures of Gaussian distributions [8] which do not directly apply to AVI data because of the discrete nature of the observations.

The algorithm proposed in this work includes a novel automatic selection of the mixture model order. The early literature on model order selection includes approaches using model selection criteria such as the *Akaike information criterion* (AIC) [9], *Bayes information criterion* (BIC) [10], and *model description length* (MDL) criterion [11][2]. These criteria

[2] The MDL is the same model order selection rule as the BIC derived using different approaches.

are used to score potential candidate models and require computing all of the candidate models before selecting one, which can be computationally intensive. Models which require only computing a single or small set of candidate models are better suited to this class of problems, in which we would like to track traffic patterns as they evolve.

More recent work on automatic model order selection is performed by Corduneanu and Bishop [12] where a Bayesian specification of the mixture model is made. Variational methods are used to anneal the number of components from some maximum number down to a number which specifies the model most accurately. Another important work in annealing a maximum number of components is the CEM$^2$ algorithm [13]. This class of work continues with Chen et al. [14] which removes the variational estimation schemes of [12] for a more traditional EM approach using unique penalty terms in the likelihood specification of the mixture model. These penalty terms allow for components to disappear if they were deemed unnecessary.

We opt for an alternate approach which does not require pre-specification of the maximum number of components. Verbeek et al. [15] explore this by taking a greedy approach of adding clusters as deemed necessary. At each iteration their algorithm fits a model with $k$ components using the EM algorithm and then searches over a set of possible candidate models for a suitable new component to insert (if necessary), then fitting the new model with $k+1$ components using the EM algorithm. This repeats until no new components are inserted.

### Traffic Modelling

Previous approaches to modelling road network traffic typically use other sensors such as road counters, induction loops, cell phone data, and manual counting to estimate traffic patterns [16, 17].

Typically in traffic estimation research, estimation of traffic flow is done through the estimation of an *origin-destination* (OD) matrix. Peterson [18] provides an overview of many methods and estimation techniques typically applied to the OD-matrix estimation problem. Most of the methods are initialized using manually-collected traffic survey data, which is time-consuming and labor-intensive to gather, and then they use data from count sensors such as inductive loops to update or refine the OD matrix. An OD matrix is a helpful tool in city planning, but it only quantifies the volume of vehicles traveling between each origin and destination. Estimation of the OD matrix using AVI data is also addressed in [19]. The *path* which traffic takes, however, is not estimated in traditional OD matrix estimation schemes. The model proposed in this paper takes traditional traffic estimation one step further and estimates the paths that vehicles take to move from one location to another.

A preliminary version of this work appears in [20]. There we introduced the idea of modelling traffic using a mixture of Markov chains. This paper expands on that work in a number of ways: 1) the inference algorithm described here differs from the one in [20] and admits a simpler analysis of convergence and computational complexity; 2) the performance evaluation is greatly expanded; 3) the issue of tuning algorithm parameters is investigated and a procedure for determining suitable choices is described.

### 6.1.3 Contribution

This work proposes a new method for the estimation of a time-varying mixture of Markov chains. The proposed algorithm extends the Classification EM algorithm [21] so components of the mixture can be greedily added as necessary. It also proposes novel penalization methods on model complexity to automatically choose a more appropriate model. A discussion of how the choice of the threshold on the proposed penalization method based on the Kullback-Leibler (KL) divergence influences is also provided. The accuracy of the parameter estimates and estimated model order generated by this algorithm are analysed and it is shown that the selection of the correct model order and estimation of the component parameters is well solved by this method. Experiments illustrate that the proposed time-varying mixture model representation of traffic is more accurate than an approach using a single Markov chain (i.e., a model of order one) for predicting vehicle routes when predictions must be made from a small number of initial observations.

### 6.1.4 Paper Organization

The rest of the paper is organized as follows. Section 6.2 gives a detailed description of the problem setup and assumptions. Section 6.3 proposes a generative time-varying mixture model. Section 6.4 discusses *maximum a posteriori* (MAP) inference for the proposed model and motivates the need to use approximate inference schemes for this class of models. Section 6.5 describes a novel approximate algorithm for parameter inference of time-varying mixture models. The proposed method involves two tuning parameters, and Section 6.6 investigates the proper choice of these parameters. Section 6.7 presents experimental results, including a comparison to an alternative MCMC-based approach, and we conclude in Section 6.8.

## 6.2 Problem Description

This section gives a more detailed description of the problem of modelling traffic using AVI data. We describe characteristics of the measurement system that make the problem challenging. Then we discuss the assumptions made and describe performance metrics which will be used to evaluate the proposed model and estimation algorithm.

### 6.2.1 Vehicle Identification Data

Consider a system of urban roads instrumented with AVI sensors. When a vehicle passes within range of a sensor, the sensor records the vehicle's unique identifier as well as the time and location. AVI sensors typically have a short range of detection (e.g., 10 meters). The measurements from many of these sensors, at different locations in the road network, are transmitted to a fusion center whose goal is to estimate the traffic patterns, or flows, through the network of sensors.

More formally we consider a collection of $C$ sensors. Let the set of sensor indices be $\Omega = \{1, ..., C\}$. The sequence of observations of vehicle $i$ produced by the network is $x^i = \{x^i_0, x^i_1, x^i_2, ..., x^i_{n_i}\}$ where $x^i_j \in \Omega$ is the sensor that captured the $j^{th}$ observation of vehicle $i$. In this paper we focus on modelling the routes present in the transition sequences between sensors and therefore ignore the transition times it takes to move from one sensor to another.

### 6.2.2 Measurement System Characteristics and Assumptions

We make the following assumptions about the measurements made by the system. First the sensors are synchronized so that the timestamps from different sensors are directly comparable. This is justified since existing AVI sensors are typically equipped with GPS receivers that provide reliable and accurate synchronization. This allows us to properly time-order observation sequences of vehicles so there are no errors in the order of vehicle observations.

Second, we assume that a vehicle cannot be observed by two sensors at *exactly* the same time. This ensures that the sequence of observations of a vehicle is well-defined. This is justified when the timestamps at each sensor are of a sufficiently high resolution and sensors have non-overlapping fields of view.

Third, we assume that the vehicle identifiers (e.g. license plate of the vehicle) recorded by the sensors do not contain errors and that there are no missed detections (i.e., a vehicle is always detected when it passes within range of a sensor). This is reasonable for license plate recognition cameras, which have a very low error rate and high detection probability, but it is less reasonable for other AVI sensors (e.g., Bluetooth-based). Extending the model and inference method to accommodate errors and missed detections is a subject of future work.

Fourth, we assume that the sensors are static and their locations are known to the fusion center. Thus, the structure of the network of AVI sensors does not change over time.

Finally, we assume that the sensors transmit their observations to the fusion center over a transmission channel with negligible delays and errors so that ordering errors due to delayed data is impossible. This assumption can be justified since the number of bits required to encode an individual measurement is very small, so transmission delay should be significantly smaller than the time between successive observations of a vehicle.

### 6.2.3 Sequential Modeling

In this work we consider a sequential setting where the observations arrive in time windows $t \in \{1, 2, ...\}$. To simplify the presentation, we take all time windows to be the same length. For example, in a typical urban deployment, a time window may be one hour long. A typical vehicle trip may last 20–30 minutes, and the number of observations of the vehicle (which depends on the number and density of cameras deployed) may be between five and ten. We assume that the distribution governing vehicle routes is stationary within a time window, and it may vary from one time window to the next. The number of vehicles observed in time window $t$ is $N(t)$. The collection of all AVI observations in time window $t$ are organized into the set of per-vehicle observation sequences, $X(t) = \{x^1, ..., x^{N(t)}\}$. In each time window a model is created to describe the data observed within that time window. This model uses the model estimated in the previous time window as a prior on the current model. This enforces the notion of the model evolving. The details of the model used to estimate traffic patterns are described in Section 6.3.

Our goal is to model vehicle routes, as given by the sequence of locations where the vehicle is observed. One possible application is to predict vehicle trajectories through the network. In Section 6.7 we report the results of experiments where, after fitting the model with training data, we test the accuracy of predicting the next location where a vehicle will be observed, $x^i_n$, given the history $x^i_0, \ldots, x^i_{n-1}$.

Note that we do not model the observation times (or inter-observation times) in this work, since the focus is on modelling vehicle routes. Such information could be incorporated, if it is of interest in a given application, following the approach described in [3]. There a probability density function $f(t|x, x')$ is associated with each pair of sensors $(x, x') \in \Omega \times \Omega$, and it is used to express the probability that a vehicle is observed by sensor $x'$ no more than $t$ seconds after having been observed at sensor $x$. These densities can be estimated while also fitting other route-related model parameters; see [3] for a detailed example.

## 6.3 The Model

Observations from automatic vehicle identification (AVI) sensors result in timestamped and location-tagged observations of vehicles identified by their unique identifier. If one groups the observations by vehicle, then each vehicle has a location (state) where it was initially observed at and then a sequence of following observations. Each following observation can be viewed as a transition from the previously observed state. We can then model these observations as transitions between the states of a Markov chain where the states of the chain are the AVI sensors. In order to capture more complex traffic patterns we propose to use a mixture of Markov chains, since a single Markov chain may be unable to properly summarize the multiple traffic patterns present.

We model the sequence of sensors observing a particular vehicle $i$ as following a first-order discrete-time Markov chain. Under the first-order Markov assumption, sufficient statistics for the sequence $x^i = (x^i_0, x^i_1, \ldots, x^i_{n_i})$ of observations of vehicle $i$ are the initial state $x^i_0$ and a matrix[3] $X^i \in \mathbb{Z}^{|\Omega|\times|\Omega|}_{\geq 0}$ of transition counts, where $X^i_{j,k}$ is the number of times

[3] We denote the set of non-negative integers by $\mathbb{Z}_{\geq 0}$.

the vehicle $i$ was observed at sensor $k$ immediately after having been observed at sensor $j$.

If one considers a single Markov chain at time window $t$ with initial state distribution $\pi(t)$ and transition matrix $P(t)$, which we group together and write as $\phi(t) = (\pi(t), P(t))$ for convenience, the likelihood of the observations of vehicle $i$ is

$$p(x^i|\phi(t)) = \pi_{x_0^i}(t) \prod_{j=1}^{|\Omega|} \prod_{k=1}^{|\Omega|} (P_{j,k}(t))^{X_{j,k}^i} . \tag{6.1}$$

Using only one Markov chain to describe all of the traffic observations in time-window $t$ provides limited predictive capacity. For example, envision a large volume of traffic moving from the suburbs into the downtown core of a city during the morning rush-hour. At the same time, traffic may be moving from the downtown core into the suburbs at a much smaller volume. If this behavior was modelled using a single Markov chain, the small traffic volume moving from downtown to the suburbs may get lost. Modelling this as a mixture model may allow us to identify these inter-twined traffic flows occurring within the same time-window (e.g., with one component capturing each "direction" or "flow"). Therefore, in order to capture more traffic patterns present in the data, we propose to use a mixture of Markov chains. This mixture model contains $M(t)$ mixture components at time-window $t$, and the $m$th mixture component has parameters $\phi^{(m)}(t) = \{\pi^{(m)}(t), P^{(m)}(t)\}$, $m = 1, \ldots, M(t)$. The likelihood for a vehicle $x^i$ under each individual component is then $p(x^i|\phi^{(m)}(t))$ as in (6.1).

Each vehicle's path is assumed to be generated by one of the components in the mixture model. In order to model this, we follow the standard approach of using the binary random variables $z_i^{(m)}$ for each vehicle $i$ and mixture component $m$, with $z_i^{(m)}$ equal to one if and only if the movement of vehicle $i$ is governed by mixture component $m$, and $\sum_{m=1}^{M(t)} z_i^{(m)} = 1$. Further let $\boldsymbol{\alpha}(t) = (\alpha^{(1)}(t), ..., \alpha^{(M(t))})$ denote the distribution of $z_i^{(m)}$ with $\mathrm{p}(z_i^{(m)} = 1) = \alpha^{(m)}(t)$ for a vehicle $i$ observed during time-window $t$.

Since the assignment variable $\boldsymbol{z}_i$ is not observed, one typically works with the marginalized complete-data likelihood,

$$p(x^i|\boldsymbol{\alpha}(t), \boldsymbol{\phi}(t)) = \sum_{m=1}^{M(t)} \alpha^{(m)}(t) p\left(x^i|\phi^{(m)}(t)\right) \tag{6.2}$$

where $\boldsymbol{\phi}(t)$ is the collection of all mixture component parameters $\phi^{(m)}(t)$ in time-window

$t$. Finally, assuming that the paths of different vehicles are i.i.d., the likelihood of $N(t)$ vehicles generating the observations $X(t)$ in time-window $t$ is

$$p(X(t)|\boldsymbol{\alpha}(t), \boldsymbol{\phi}(t)) = \prod_{i=1}^{N(t)} \sum_{m=1}^{M(t)} \alpha^{(m)}(t) p\left(x^i|\phi^{(m)}(t)\right). \tag{6.3}$$

In our previous work [3] we considered the problem of detecting when two or more vehicles were traveling along correlated routes (i.e., detecting convoys), and non-convoy traffic was modelled as independent samples from a (static) mixture of Markov chains.

The remainder of this section defines the model dynamics.

### 6.3.1 Mixture Component Death

Consider the mixture model in time-window $t-1$ with $M(t-1)$ components. This mixture model can be described by the vector of mixture weights $\boldsymbol{\alpha}(t-1)$ and parameters for each component $\phi^{(m)}(t-1), m = 1, ..., M(t-1)$. Let $\boldsymbol{\Theta}(t) = (\boldsymbol{\alpha}(t), \boldsymbol{\phi}(t))$ denote the complete set of model parameters at time $t$.

Existing components either persist or die between time-windows, and they die with probability

$$p_d(\phi^{(m)}(t), \alpha^{(m)}(t)). \tag{6.4}$$

This probability can be constant or could be related to the rate at which $\alpha^{(m)}(t)$ is decreasing signifying that the component is "dying". It could also be related to a rate of decrease in the *Kullback-Leibler* (KL) divergence between a pair of components signifying they may be "merging". The notions of a component dying on its own or merging with another are handled by the algorithm in a following section.

For each component death, the mixture weight assigned to that component is distributed proportionally among the remaining components which is similar to the generative model of Stephens [7].

### 6.3.2 Persisting Component Evolution

If a component persists from time-window $t-1$ to $t$ its parameters evolve according to the dynamics

$$\phi^{(m)}(t) \sim H(\phi^{(m)}(t-1)), \tag{6.5}$$

where $H(\cdot)$ is a distribution parametrized by the previous iteration's component's parameters with pdf $h(\phi^{(m)}(t)|\phi^{(m)}(t-1))$. In order to easily compute the MAP estimate, this distribution would ideally be conjugate to the distribution parametrized by $\phi^{(m)}(t)$. In the traffic model considered here, with a *time-varying mixture model* (TVMM) of *discrete-time Markov chains* (DTMC), we adopt the model that

$$P^{(m)}(t) \sim \begin{bmatrix} \mathrm{Dir}(P^{(m)}_{(1,:)}(t-1)) \\ \mathrm{Dir}(P^{(m)}_{(2,:)}(t-1)) \\ \vdots \\ \mathrm{Dir}(P^{(m)}_{(|\Omega|,:)}(t-1)) \end{bmatrix} := \mathbf{Dir}\left(\boldsymbol{P}^{(m)}(t-1)\right) \tag{6.6}$$

and

$$\pi^{(m)}(t) \sim \mathrm{Dir}(\pi^{(m)}(t-1)) \tag{6.7}$$

where $\mathrm{Dir}(\cdot)$ is the Dirichlet distribution.

### 6.3.3 Mixture Component Birth

The number of mixture components born at each time-window follows a Poisson distribution with parameter $\lambda(t)$ as $n_b \sim \mathrm{Pois}(\lambda(t))$. This allows for a countably infinite number of components to be born in each time-window. Each new component which is born in time-window $t$ has parameters distributed according to some distribution, $\phi^{(m)}(t) \sim H(\mathbb{G})$, which has a density $h(\phi^{(m)}(t)|\mathbb{G})$. Here $\mathbb{G}$ are the parameters, or set of parameters, of the base distribution of the model. In our TVMM DTMC these distributions are

$$P^{(m)}(t) \sim \mathbf{Dir}(G_p) \text{ and } \pi^{(m)}(t) \sim \mathrm{Dir}(G_\pi) \tag{6.8}$$

where $\mathbb{G} = \{G_p, G_\pi\}$ are the set of base distribution parameters.

Each new component which is born has a mixture weight distributed according to a Beta distribution so that

$$\alpha^{(M(t)+1)}(t) \sim \mathrm{Beta}(1, M(t)). \tag{6.9}$$

Following [7], for each new component added, the weights of previously existing components are scaled down by $\left(1-\alpha^{(M(t)+1)}\right)$. This model for component birth can be related to the notion that components are "born from the prior" so they have some shared knowledge between all components but are independent from all other existing components. In this model new components also receive progressively smaller weights which helps in that no new component can appear and take the majority of the weight in one iteration. It will be born small and, if necessary, will grow through its dynamics. This notion of a curbed mixture weight with additional births is also adopted from [7] and differs from the typical non-parametric Dirichlet process mixture models in that the parameter of the Beta distribution varies, depending on the number of existing components, over time.

### 6.3.4 Mixture Weight Dynamics

The fully dynamic model is now defined, save for the evolution of the vector of mixture weights, $\boldsymbol{\alpha}$, between time windows $t-1$ and $t$. The dimension of this vector can change between time-windows and it is modelled as following a multinomial distribution. Consequently, it is natural to model these weights as evolving according to a Dirichlet distribution because the Dirichlet distribution is conjugate to the multinomial distribution [22]. However the parameter to the Dirichlet must have the same dimension as the variable output from the distribution. Therefore we propose that one considers that the model first determines the component births and deaths, where the mixture weights are adjusted for each birth and death, and then the dimension of the mixture weight vector is known. After the births and deaths are established, we have a resized and rescaled weight vector for time $t-1$ called $\boldsymbol{\alpha}'(t-1)$. With this rescaled mixture weight vector the next mixture weight vector is distributed according to

$$\boldsymbol{\alpha}(t) \sim \mathrm{Dir}(\boldsymbol{\alpha}'(t-1)). \tag{6.10}$$

## 6.4 MAP Inference for a Mixture of Markov chains

In the ideal case we would estimate the parameters of a time-varying mixture model at time $t$ via Bayes' Rule and get a maximum a posteriori (MAP) estimate of the parameters. Given the data from time-windows $1, ..., t$, denoted $\mathbf{X}(1:t) = \{X(1), ..., X(t)\}$, we want to estimate the parameters that maximize the posterior $p(\boldsymbol{\Theta}(t)|\mathbf{X}(1:t))$ which can be rewritten as

$$\begin{aligned}
p(\mathbf{\Theta}(t)|\mathbf{X}(1:t)) =& \frac{p\left(X(t), \mathbf{\Theta}(t)|\mathbf{X}(1:t-1)\right)}{p(X(t)|\mathbf{X}(1:t-1)} \\
&\propto p\left(X(t)|\,\mathbf{\Theta}(t)\right) p\left(\mathbf{\Theta}(t)|\mathbf{X}(1:t-1)\right) \qquad (6.11)\\
&= p\left(X(t)|\,\mathbf{\Theta}(t)\right) \int p\left(\mathbf{\Theta}(t)|\,\mathbf{\Theta}(t-1)\right) \\
&\qquad \times p\left(\mathbf{\Theta}(t-1)|\mathbf{X}(1:t-1)\right) d\,\mathbf{\Theta}(t-1)
\end{aligned}$$

by application of Bayes' theorem where the denominator in the first equation is simply a scaling factor so it does not affect the maximization. The MAP estimate is then given by

$$\begin{aligned}
\hat{\mathbf{\Theta}}_{MAP}(t) =& \arg\max_{\mathbf{\Theta}(t)} p\left(X(t)|\,\mathbf{\Theta}(t)\right) \qquad (6.12)\\
&\times \int p\left(\mathbf{\Theta}(t)|\,\mathbf{\Theta}(t-1)\right) p\left(\mathbf{\Theta}(t-1)|\mathbf{X}(1:t-1)\right) d\,\mathbf{\Theta}(t-1).
\end{aligned}$$

As the dimensionality of the model grows, computing the integral over the entire parameter space $\mathbf{\Theta}(t-1)$ quickly becomes intractable. We can however further develop the terms appearing in the MAP objective.

In order to solve this maximization, we need expressions for $p(X(t)|\,\mathbf{\Theta}(t))$ and $p(\mathbf{\Theta}(t)|\,\mathbf{\Theta}(t-1))$ in (6.12). The likelihood $p(X(t)|\,\mathbf{\Theta}(t))$ is given by (6.3). The transition density $p(\mathbf{\Theta}(t)|\,\mathbf{\Theta}(t-1))$ is governed by the model dynamics described in Sec. 6.3. The term $p(\mathbf{\Theta}(t-1)|\mathbf{X}(1:t-1))$ is the recursive posterior from time window $t-1$, acting as the current time-window's prior.

In order to express the parameter transition density we need to match the persisting mixture components in time window $t-1$ with those in time window $t$. If $n_d$ components die from $t-1$ to $t$ then $M(t-1)-n_d$ components persist.

Consider the set of choices of size $M(t-1)-n_d$ from the set $\{1, ..., M(t-1)\}$. Furthermore consider all permutations of each resulting choice in this set of choices. The concatenation of all permutations of all possible choices from this set is denoted by $\mathcal{A}(t)$. This is the set of all possible mappings of the components at time-window $t-1$ to those which persisted to time-window $t$. In the definition of $p(\mathbf{\Theta}(t)|\,\mathbf{\Theta}(t-1))$ we need to marginalize over all the possible mappings in $\mathcal{A}(t)$. We can note that incorrect mappings will likely have very small probability, and there will likely only be one mapping which results in a reasonably large probability. Now if one further assumes that the probability of death for any component is $p_d$ for all $t$, then we can define $p(\boldsymbol{\Theta}(t)|\,\boldsymbol{\Theta}(t-1))$ to be [4]

$$\begin{aligned}
&p(\,\boldsymbol{\Theta}(t)|\,\boldsymbol{\Theta}(t-1)) \qquad (6.13)\\
&= \sum_{n_d(t)=\max(0,M(t-1)-M(t))}^{M(t-1)} \Bigg\{ f_d(n_d(t)|p_d, M(t-1)) \times \\
&\quad f_b(M(t)-M(t-1)+n_d(t)|\lambda(t)) \times \\
&\quad \left( \sum_{A\in\mathcal{A}(t)} \prod_{m=1}^{M(t-1)-n_d(t)} g(\phi^{(m)}(t)|\phi^{(A_m)}(t-1)) \right) \times \\
&\quad \left( \prod_{n_b(t)=M(t-1)-n_d(t)+1}^{M(t)} g(\phi^{(m)}(t)|\mathbb{G}) f_\alpha(\alpha^{(m)}(t)|M(t-1)) \right) \\
&\quad \times f_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}(t)|\boldsymbol{\alpha}'(t-1)) \Bigg\}
\end{aligned}$$

where

$$f_d(x|p_d, M(t-1)) = p_d^x\,(1-p_d)^{M(t-1)-x} \qquad (6.14)$$

$$f_b(k|\lambda(t)) = \frac{\lambda(t)^k e^{-\lambda(t)}}{k!} \qquad (6.15)$$

$$f_\alpha(x|M(t)) = \frac{(1-x)^{M(t)-1}}{\mathrm{B}(1, M(t))} \qquad (6.16)$$

$$f_{\boldsymbol{\alpha}}(\mathbf{x}|\boldsymbol{\alpha}'(t-1)) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha}'(t-1))} \prod_{m=1}^{|\mathbf{x}|} x_m^{\alpha'_m(t-1)} \qquad (6.17)$$

and where $\mathrm{B}(\cdot)$ is the Beta function and $f_{\boldsymbol{\alpha}}()$ controls the evolution of the *rescaled* mixture component weights ($\boldsymbol{\alpha}'$) defined in (6.10).

In order to do inference on this model, one could resort to a Monte-Carlo based method, such as Gibbs Sampling. However, due to the high-dimension of the model parameter space the number of burn-in and sampling iterations of the MCMC method quickly goes to infinity in order to sample reasonable candidates models from such a large search space. As

[4] We adopt the convention that $\prod_{i=1}^{0} x = 1$.

a simple example, consider the TVMM DTMC model again with a state space of $|\Omega| = 25$ representing a network of 25 nodes. If the current model in time-window $t$ is comprised of 3 DTMCs, one needs to estimate the mixture weights ($\boldsymbol{\alpha}(t)$), the initial state distributions ($\pi(t)$), and transition matrices ($P(t)$) for each component resulting in

$$\begin{aligned}(|\alpha| - 1) + |\alpha| \times [(|\Omega| - 1) + |\Omega|(|\Omega| - 1)]\\ = 2 + 3(24 + 25 \times 24) = 1874\end{aligned}$$

free parameters which need to be estimated. In addition to the number of free parameters which need to be estimated, there is the issue that random samples from the posterior distribution in a MCMC method will be candidate mixture models which have a varying number of components. Therefore determining a method of averaging the posterior samples or choosing the best candidate model in order to obtain the best parameter estimate is not a straightforward task as well.

Now since MCMC-based methods result in unreasonable sampling times, the question becomes how does one estimate the parameters of this type of model? An alternate might be to try and derive an exact Maximum a Posteriori (MAP) estimate based on the probability definitions in 6.12, 6.13, 6.14, 6.15, 6.16, and 6.17. If one excludes the definition of $p(\boldsymbol{\Theta}(t)|\,\boldsymbol{\Theta}(t-1))$ for a moment and only focuses on $p(X(t)|\,\boldsymbol{\Theta}(t))$ this would result in a maximum likelihood (ML) estimate of the mixture model which we can show is still intractable to compute.

To demonstrate the infeasibility of a ML solution in the mixture model, consider the hidden assignment variable $z_i^{(m)}$ for vehicle $i$ outlined in Section 6.3 again. This variable $z_i^{(m)} = 1$ when vehicle $i$'s path is distributed according to mixture component $m$ and 0 otherwise. Now the updated likelihood function using this hidden variable definition is

$$f(X(t)|\,\boldsymbol{\Theta}(t)) = \prod_{i=1}^{N(t)} \sum_{z_i \in \mathcal{Z}} \prod_{m=1}^{M(t)} f(x^i, z_i^{(m)}|\phi^{(m)}(t)) \tag{6.18}$$

$$= \prod_{i=1}^{N(t)} \sum_{z_i \in \mathcal{Z}} \prod_{m=1}^{M(t)} \left( \pi_{x_0^i}^{(m)}(t) \prod_{j=1}^{|\Omega|} \prod_{k=1}^{|\Omega|} \left( P_{j,k}^{(m)}(t) \right)^{X_{j,k}^i} \right)^{z_i^{(m)}} \tag{6.19}$$

where the marginalization over all possible $z_i$'s make computing the ML estimate quickly go to infinity. The actual number of iterations to marginalize over $\mathcal{Z}$ is $M(t)^{|X(t)|}$ which one can see is dependent on the mixture-model order in time-window $t$ and the number of vehicles observed in time-window $t$. This marginalization requirement is true of all types of mixture models, not only a TVMM of DTMCs. These types of models are traditionally estimated using the Expectation-Maximization algorithm [23] which replaces the hidden assignments $z_i$ from binary random variables with weighted probabilities by stating that vehicle $i$ has probability $\alpha_i^{(m)}$ of being distributed according to component $m$. This greatly simplifies the computation of the marginal likelihood and it then becomes possible to maximize the *expected complete log-likelihood* via a simple recursive algorithm. It proceeds by choosing the component mixture probabilities based on the ratio of the likelihood of the data being distributed according to each individual component. This is performing a form of *clustering*, which can motivate the need to first *cluster* the data into groups and then updating the individual component parameters based on the data membership to each component. We propose the automatic hard EM algorithm defined in the following section to address the clustering and estimation problem in a time-varying mixture model.

## 6.5 Automatic Hard EM Estimation for Time-Varying Mixture Models

When the true number of underlying components in a mixture model is unknown a priori, then a mixture model estimation scheme needs to estimate the number of mixture components as well as the parameters of the individual mixture model components and the mixing weights for each component. Traditional *Expectation-Maximization* (EM), *Classification EM* (CEM), and $k$-Means algorithms require that the number of mixture components be known beforehand [21, 23, 24]. This means that a modification to these types of estimation schemes is necessary.

We propose an approach to model order estimation similar to that described in [15], which progressively adds components. However, the proposed approach eliminates the search over candidate models by simply adding the same base candidate model $\mathbb{G}$ at each iteration. This reflects the generative model where new components born at each iteration are generated according to this global prior.

### 6.5.1 Modified Hard EM Clustering

The EM algorithm [23] can be viewed as performing a form of clustering where the cluster assignments are soft: for each datapoint we get a likelihood that the datapoint came from each of the possible clusters. Then the cluster parameter estimates are updated to the weighted average of the data assigned to the cluster. Once the weighted cluster assignments and component parameters stop changing, to within some tolerance, the algorithm is considered to have converged. By modifying the EM algorithm to make hard assignments instead of weighted assignments, one obtains an alternate method commonly referred to as Hard EM. This is a reasonable approximation when the mixture components are well-separated, and it is generally much faster to compute. Hard EM algorithms can be considered to have converged once the component assignments stop changing since, once this occurs, the component parameter estimates will be exactly the same in subsequent iterations.

Since we assume that vehicles follow a single DTMC in the mixture model, we use Hard EM to estimate the DTMC that generated the observed vehicle path. We then use the paths of the group of vehicles assigned to a component to compute a simple MAP estimate of the component's parameters.

### 6.5.2 The Proposed Algorithm

For each time-window $t$, the proposed inference algorithm is divided into three phases. The first phase is a modification of the Hard EM algorithm to jointly estimate model parameters and the model order. The phase is initialized with the mixture model parameters estimated in the previous time-window, and it allows for an arbitrary number of new components to be created. In the second phase, components that have fewer than $L_\alpha$ data points assigned to them are trimmed from the model. In the last phase, pairs of the remaining components are merged if the KL divergence between them falls below a threshold $L_{KL}$. These last two phases simplify the resulting model, reducing the storage requirements and simplifying future evaluations of the model, while also acting as a sort-of regularizer to avoid overfitting. Pseudocode is shown in Alg. 1 and each phase is described in more detail next.

**Algorithm 1** Automatic Hard EM algorithm for $T$ time-windows

```
$\mathbf{\Theta}(0) = \{\}$
Require: Data $\mathbf{X}(1:T) = \{\mathbf{X}(1), ..., \mathbf{X}(T)\}$
for $t \in 1..T$ do
    $\hat{\phi}_0 = \{\phi^{(m)}(t-1)\}^{\forall m} \bigcup \{\mathbb{G}\}$
    while $|S_{M(t)}| > 0$ do
        $i = 1$
        repeat
            for $m \in \{1, ..., M(t)\}$ do
                $S_m = \left\{ x^j : p(x^j|\hat{\phi}_{i-1}^{(m)}) > p(x^j|\hat{\phi}_{i-1}^{(m')}) \forall m' \neq m \right\}$
                $\hat{\phi}_i^{(m)} = \arg\max_{\hat{\phi}_i^{(m)}} f(S_m|\hat{\phi}_i^{(m)}) h(\hat{\phi}_i^{(m)}|\hat{\phi}_0^{(m)})$
            end for
            $i = i + 1$
        until $|\hat{\phi}_i^{(m)} - \hat{\phi}_{i-1}^{(m)}| = 0, \forall m$
        $\alpha^{(m)}(t) \leftarrow \frac{|S_m|}{|\mathbf{X}(t)|}, \forall m$
        if $|S_{M(t)}| > 0$ then
            $\hat{\phi}_i(t) = \{\hat{\phi}_i^{(m)}(t)\}^{\forall m} \bigcup \{\mathbb{G}\}$
            $\hat{\phi}_0(t) = \{\hat{\phi}_0^{(m)}(t)\}^{\forall m} \bigcup \{\mathbb{G}\}$
        end if
    end while
    $\mathbf{\Theta}(t) \leftarrow \left\{ (\alpha^{(1)}(t), \hat{\phi}_i^{(1)}(t)), ..., (\alpha^{(M(t))}, \hat{\phi}_i^{(M(t))}(t)) \right\}$
    Trim components in $\mathbf{\Theta}(t)$ with $\alpha^{(m)} < L_\alpha$
    Merge components in $\mathbf{\Theta}(t)$ with
        $D_{KL}(\phi^{(m)}(t)||\phi^{(m')}(t)) < L_{KL}$
end for
```

### Modified Hard EM

The modification that we propose to the standard hard EM algorithm allows the algorithm to greedily add new components as necessary, so the number of components does not need to be specified in advance. The first phase begins by adding a new component with parameters $\mathbb{G}$ (line 3). It then finds the component which achieves the maximum likelihood for each observed path (line 8). From the vehicle paths assigned to each component, the MAP estimate of the parameters is computed (line 9). In the MAP estimate computation, $f(S_m|\hat{\phi}_i^{(m)})$, is the likelihood of the data assigned to component $m$, and $h(\hat{\phi}_i^{(m)}|\hat{\phi}_0^{(m)})$ is the same as in (6.5). By adding components with initial parameters of $\mathbb{G}$, this is equivalent to setting the parameters of the prior over the component parameter distribution to the same values. This means that the distribution over the components, $h(\cdot|\cdot)$, is the same for all newly added components. These steps are repeated until the assignments no longer change from one iteration to then next (line 12). Then (line 13), the algorithm updates the mixture weights $\alpha^{(m)}(t)$ based on the number of observations assigned to each component.

Next (line 14), the algorithm checks if any data has been assigned to the most-recently appended component. Since this component was initialized as the base distribution $\mathbb{G}$, it is more vague (i.e. it has higher variance or spread) than the components propagated from the previous time-window. If paths are assigned to this component, then at least one new cluster is deemed to have formed, and another component is added (lines 14–17). Then the Hard EM algorithm is executed again with this new, larger, model order. This process repeats until another instance of the base component $\mathbb{G}$ is added and no data points are assigned to it. In the worst case scenario where all observed paths in a time-window are well-separated, one could end up with a number of components equal to the number of observed paths. However in the empirical studies discussed in Section 6.7, we did not observe this extreme, and typically $M(t) \ll |\mathbf{X}(t)|$. This estimation technique allows new component creation and existing component movement.

If this were the only phase of the algorithm, the number of components would continuously grow without limit since we only add new components in this phase of each time-window. Therefore we next need a method of annealing the number of components to those which are necessary, both for computational considerations and to avoid over-fitting.

### Trimming Weak Components

The most basic way of trimming components that do not describe the data is to remove components that are associated with a small proportion of data; these are precisely the components with small weights $\alpha^{(m)}(t)$. This is implemented in line 20. The threshold, $L_\alpha$, could be constant, a time-window dependent value, or a data size-dependent value. One might wish to keep all components which are sufficiently well-separated from the rest, in which case $L_\alpha$ would be set to zero.

### Merging Similar Components

A more complex notion is examining when two components become very close to each other (meaning they are essentially the same distribution). After the clustering has converged and weak components are deleted, we compute the KL divergence between all pairs of components and merge pairs if the KL divergence from one to the other is below a threshold, $L_{KL}$, starting from the smallest difference and proceeding sequentially. When computing the KL divergence between all pairs of components, we need to compute both combinations $D_{KL}(m||m')$ and $D_{KL}(m'||m)$ and take the minimum component and test if it is below $L_{KL}$. To merge components $m$ and $m'$ we take the weighted average of their parameters,

$$\begin{aligned}\alpha^{\text{merged}}(t) &= \alpha^{(m)}(t) + \alpha^{(m')}(t)\\ \phi^{\text{merged}}(t) &= \frac{\alpha^{(m)}(t)\phi^{(m)}(t)}{\alpha^{(m)}(t) + \alpha^{(m')}(t)} + \frac{\alpha^{(m')}(t)\phi^{(m')}(t)}{\alpha^{(m)}(t) + \alpha^{(m')}(t)}.\end{aligned}$$

We then compute all KL divergences with the new component structure and repeat the test and merge if necessary until no KL divergence is below the threshold. This is implemented in line 21 of Alg. 1. The choice of an appropriate threshold $L_{KL}$ is discussed in Section 6.6.

### 6.5.3 Intuitive Explanation of the Automatic Hard EM Algorithm

Next we provide an intuitive explanation to help understand the function of the Automatic Hard EM algorithm. Suppose that the mixture model estimated at time-window $t-1$ has two components, and 500 data points are observed in time-window $t$. In the first through the CEM algorithm (lines 6–12), each data point will be assigned to one of three components: the two from the previous time-window, and the additional one added in line 3.

After this first pass of the CEM has converged, if no data points were assigned to the third mixture component then the algorithm determines that the original two components were sufficient to model the data, and the while loop exists. Otherwise, if at least one data point was assigned to the third mixture component, then the algorithm allows for the possibility of increasing the model order further. Another generic component is added in lines 15–16, and another pass of the CEM algorithm is performed. This repeats until no data points are assigned to the most recently added component, at which point it is determined there is no need to further increase the model order. Thus this first phase involves greedily increasing the model order as long as the CEM continues to associate data points to new mixture components.

To be concrete, suppose that this phase finishes with a model of order $M(t) = 4$, and the number of data points associated to each component are as follows:

- Component 1 has 200 data points;
- Component 2 has 0 data points;
- Component 3 has 300 data points;
- Component 4 has 0 data points.

Recall that the first two components were propagated forward from the previous time step, and components 3 and 4 were added in this time-window.

At this stage the last two algorithm steps (trimming and merging) are executed. The trimming phase deletes components to which no (or only a few) data points have been assigned; in our example, component 2 from the previous time-window has died, since it no longer explains any of the observations, and component 4 is eliminated because it was added unnecessarily.

The last step is the merging step. Following our example, if the KL divergence between the remaining two components, 1 and 3, is smaller than the threshold $L_{KL}$, then the initial state distributions and transition matrices of these two components are deemed too similar, and so they will be merged, resulting in a final model with a single component. If the KL divergence between the two components is larger than $L_{KL}$ then the two components are deemed to be sufficiently different that they both remain and propagate forward to the next time-window.

### 6.5.4 On the Convergence of the Algorithm per Time-Window

For a fixed model order, Wu [25] shows that convergence of the EM algorithm to a local maximum is guaranteed by demonstrating that the EM iterations monotonically increase the likelihood. Convergence of the CEM algorithm to a local maximum, in the case of a fixed model order, is shown by Celeux and Goavert [21]. They accomplish this by introducing the *classification maximum likelihood* (CML) criterion,

$$C_2(Z(t), \mathbf{\Theta}(t)) = \sum_{m=1}^{M(t)} \sum_{x_i \in Z^{(m)}(t)} \log\left(\alpha^{(m)}(t) p(x_i|\phi^{(m)}(t))\right) \tag{6.20}$$

where $Z^{(m)}(t) \subset \mathbf{X}(t)$ is the subset of data in $\mathbf{X}(t)$ which is associated with component $m$. Then they show that each CEM iteration increases this criterion. Since it is bounded above, it follows that the algorithm converges to a local maximum.

Now we discuss why the Automatic Hard EM algorithm is guaranteed to converge within each time window. Specifically, we explain why the while loop spanning lines 4–18 in Algorithm 1 is guaranteed to converge. First, observe that the inner loop spanning lines 6–12 correspond precisely to the CEM algorithm with fixed model order $M(t)$. Convergence of this inner loop is guaranteed by the results of [21], and the value to which it converges is a local maximum of the CML criterion.

It remains to be shown that the outer loop terminates—specifically, that the conditional in line 4 eventually evaluates to false. Recall that $M(t)$ is the number of components in the mixture model. Within the loop spanning lines 4–18, $M(t)$ is strictly increasing. Specifically, if the condition in line 14 is true then $M(t)$ is incremented in lines 15 and 16, and the condition in line 14 is true if the most recently added component has at least one observation sequence assigned to it. Under the assumption that there are a finite number of observations in each time window, the maximum number of components is also finite; it is at most equal to the number of observations $N(t)$, in which case every component has exactly one datapoint assigned to it. Since $M(t)$ is monotonic increasing and bounded above during the execution of lines 4–18, this portion of the algorithm is guaranteed to converge. The last two steps evaluated in each time window (lines 20 and 21) both strictly decrease the number of components in the mixture model, and so they also are guaranteed to converge. Hence, the Automatic Hard EM algorithm is guaranteed to converge at each time step.

Although the algorithm is deterministic and is guaranteed to converge, we note that there are no guarantees about the quality of the estimate to which the algorithm converges. Recall that the CEM iterations monotonically increase the CML criterion given in (6.20). The loop spanning lines 4–18 involves repeatedly executing CEM iterations while progressively increasing the model order each time CEM converges. Consequently, lines 4–18 indeed monotonically increase the CML criterion, and so when this loop converges the algorithm is at a local maximum of the CML criterion. However, the trimming and merging steps in lines 20 and 21 will generally result in a decrease of the CML.

## 6.6 Threshold for Merging Similar Components

Before proceeding to numerical experiments, we discuss how to choose $L_{KL}$, the limit at which two estimated components are merged. Since the parameters of the DTMCs in the mixture model are random variables, the KL divergence between pairs of estimated DTMCs can also be considered a random variable parametrized by the hyper-parameters $\phi^{(m)}(t)$ and $\phi^{(m')}(t)$. The KL divergence in time-window $t$ between two DTMCs is non-symmetric and is defined as

$$D_{KL}(m||m'; \mathbf{\Theta}(t)) = \sum_{i=1}^{|\Omega|} \pi_i^{(m)}(t) \sum_{j=1}^{|\Omega|} P_{i,j}^{(m)}(t) \log \frac{P_{i,j}^{(m)}(t)}{P_{i,j}^{(m')}(t)}.$$

The KL divergence test in Alg. 1 aims to identify when two estimated component distributions are sufficiently close to be merged into one. This means that the hyper-parameters which govern $\phi^{(m)}(t)$ and $\phi^{(m')}(t)$ should be the same hyper-parameters. In this TVMM setting, this set of *shared* hyper-parameters is either $\{G_\pi, G_P\}$ or $\{\pi^{(m)}(t-1), P^{(m)}(t-1)\}$ for some $m$.

Due to the complexity of this expression, a closed-form expression for the distribution of the KL divergence function between two random DTMCs is not available. Therefore we propose an experimental evaluation which investigates the distribution of sampled KL divergences from mixture models with a mis-specified model order [26].

### 6.6.1 Simulated dataset 1

In order to perform an experimental evaluation of the distribution of the KL divergence, we first introduce a simulated dataset which will be used in the evaluation. The simulated dataset is created using a random network of 25 nodes generated as a random Delaunay graph: nodes are placed uniformly and independently in the unit square, and edges are obtained from the Delaunay triangulation of these points. The resulting graph is planar, similar to many road networks. An example random Delaunay graph is shown in Fig. 6.1. This dataset contains 100 time-windows where in each time-window a mixture model (as defined in the generative model of Sec. 6.3) governs the sampled data. We refer to this as Simulated Dataset 1 (SDS1), and it is generated from a mixture-model with a model order that varies between one and three components during the 100 time-windows. The probability of component death in the model is $p_d = 0.1$ and the parameter of the Poisson distribution controlling component birth is $\lambda(t) = 1$ for all $t$. The mixture model order over the 100 time-windows is shown in Fig. 6.3(a). Vehicle observations in SDS1 have a path length that is sampled uniformly between 13 and 23.

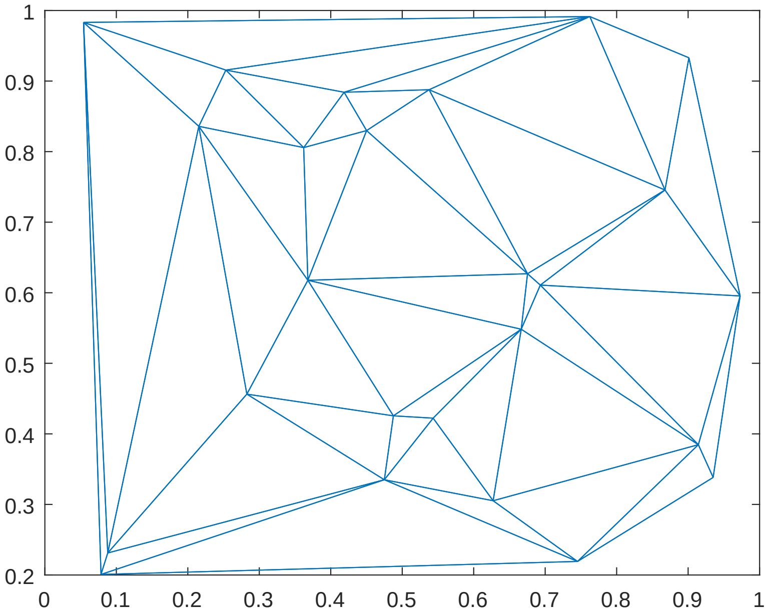
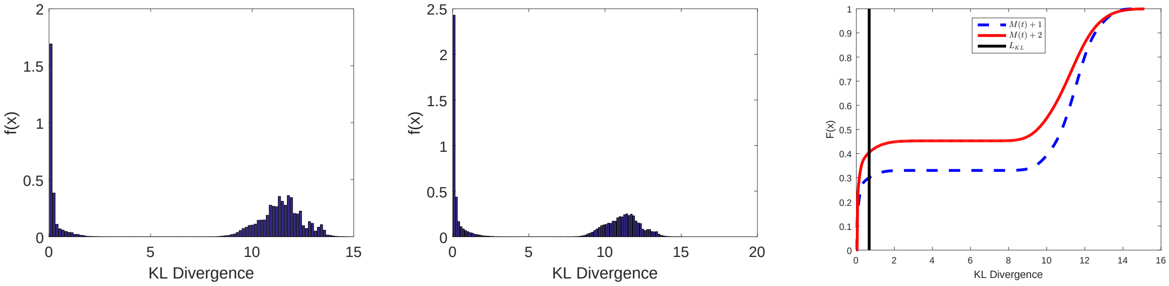


**Fig. 6.1** An example random Delaunay graph with 25 nodes

### 6.6.2 Fitting a Mis-Specified Model

Using the TVMM in SDS1, we sample 50 datasets, each containing observations of 5000 vehicles per time-window. For each of these datasets, we fit a model with exactly $M(t)+1$ components at every time-window by running the EM algorithm [23] with 10 random restarts. We keep the model with the largest likelihood which, for the same model-complexity in each fit, is equivalent to using BIC or AIC as a model selection criterion.



(a) (b) (c)

**Fig. 6.2** (a) A histogram density estimate over the sampled KL divergence values for the mis-specified $M(t)+1$ model. (b) A histogram density estimate over the sampled KL divergence values for the mis-specified $M(t)+2$ model.(c) The CDF of the fitted functions in (a) and (b) with the proposed KL divergence threshold $L_{KL} = 0.678$ superimposed at the vertical black line.

Then, for each best fit model, we compute the KL divergence between all distinct pairs of components in each time-window. The distribution of the KL divergence values is shown in Fig. 6.2(a), where it can be clearly seen that there are two peaks present. We postulate that the first peak, near zero, corresponds to the over-fit models which should be merged. In other words, one would like to see the peak near zero disappear and only the remaining peak be present, corresponding to estimated components that are well-separated.

We continue to investigate this hypothesis by examining when there is a mis-specified model of order $M(t)+2$ in each time-window. Using the same method as for when $M(t)+1$, the distribution of the KL divergence values is shown in Fig. 6.2(b). Here we can see the mass of the distribution near zero has indeed grown. In order to further see this consider Fig. 6.2(c) which shows the CDF of the distributions in Fig. 6.2(a) and Fig. 6.2(b), obtained using kernel density estimates. Here we can clearly see that the mass in the near zero models has indeed grown under the *more mis-specified* model $(M(t)+2)$, which confirms that we would like to set $L_{KL}$ to trim this lower peak.

We propose to set the threshold $L_{KL}$ by computing the numerical derivative of the CDF and searching for the first value after the left-most peak where the slope approaches zero (e.g., a slope less than 0.05). This assures that one only removes the peak close to zero, while leaving the majority of the KL space available for models to exist. If one were to choose the threshold from the right, computing the slope less than 0.05, then we would only allow components to exist which are very well separated when in actuality components may naturally have exist close to each other. For the simulated network this results in a value of $L_{KL} = 0.678$, shown as the vertical, black line in Fig. 6.2(c). The reason for choosing the threshold as close to 0 as possible is we want to be as sensitive as possible to components which will naturally appear close together and will randomly have a small divergence while removing components which are over-fit and have an unnaturally small KL divergence.

## 6.7 Numerical Experiments

We now continue with a performance evaluation of the automatic hard EM algorithm using both simulated and real data. The first simulated dataset is SDS1 which was described in Sec. 6.6.1. Two other simulated datasets were also created with a random network of 25 nodes simulated as another realization of a random Delaunay graph. These simulated datasets each also contain 100 time windows, and in each time window observations are sampled from a mixture model. The plot of this second dataset's mixture model order over time is shown in Fig. 6.3(b). For this network and mixture model, two datasets (SDS2-a and SDS2-b) were generated. The first dataset, SDS2-a, contains sets of vehicle observations with path length distributed uniformly between 5 and 15, and SDS2-b contains sets of vehicle observations with path length distributed uniformly between 15 and 25. The intention is to investigate how the number of observations per-vehicle affects the accuracy. The probability of death and birth are set to the same as SDS1, described in Sec. 6.6.1. A summary of the simulated dataset parameters is given in Table 6.1.

Comparing the time-varying model orders in Fig. 6.3, note that SDS1 corresponds to a simpler test case and SDS2, with overall larger model orders, is more challenging. For SDS2-a and SDS2-b we generate 10 independent realizations for each of the eight possible values of $N(t)$ outlined in Table 6.1 in order to report average results.

We set $L_\alpha = \frac{2}{N(t)}$ so that we only delete components where two or less vehicles gets

**Table 6.1** Summary of the parameters of the three simulated datasets SDS1, SDS2-a, and SDS2-b. The set $\mathcal{N} = \{100, 200, 500, 1000, 2000, 5000, 10000, 20000\}$.

| **Dataset** | SDS1 | SDS2-a | SDS2-b |
|---|---|---|---|
| **Model Complexity** | Fig. 6.3(a) | Fig. 6.3(b) | Fig. 6.3(b) |
| **Network Size** $(\lvert\Omega\rvert)$ | 25 | 25 | 25 |
| $(N(t))$ | $N(t) \in \mathcal{N}$ | $N(t) \in \mathcal{N}$ | $N(t) \in \mathcal{N}$ |
| **Number of Realizations** | 1 | 10 | 10 |
| **Observation Path Length** | $\sim$ Uniform$(13, 23)$ | $\sim$ Uniform$(5, 15)$ | $\sim$ Uniform$(15, 25)$ |
| $L_{KL}$ | 0.678 | 0.4 | 0.4 |
| $L_{\alpha}$ | $2/N(t)$ | $2/N(t)$ | $2/N(t)$ |

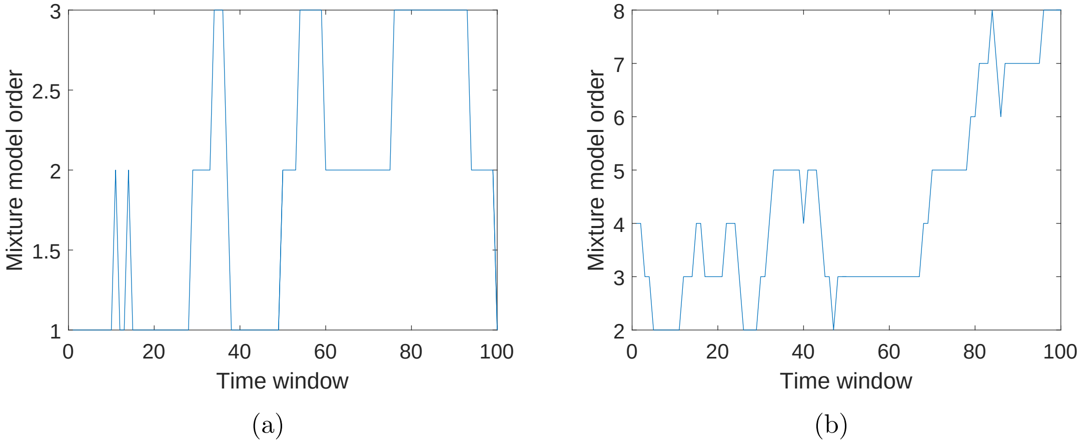


**Fig. 6.3** (a) The mixture model order (number of mixture components) for the first dataset (SDS1) over the 100 time-windows. (b) The mixture model order for the second and third datasets (SDS2-a and SDS2-b) over the 100 time-windows.

assigned after the hard EM phase. For these simulations we want to see all possibly relevant components which have been estimated. For SDS2-a and SDS2-b, the same analysis as SDS1 was performed in order to choose the $L_{KL}$ threshold (see Sec. 6.6), and we conclude that a threshold of $L_{KL} = 0.4$ is appropriate for this network.

### 6.7.1 Evaluation of the Algorithm Performance

Evaluation of Alg. 1 was performed by examining both the estimated number of components and the $\ell_1$ error of the model component parameters.

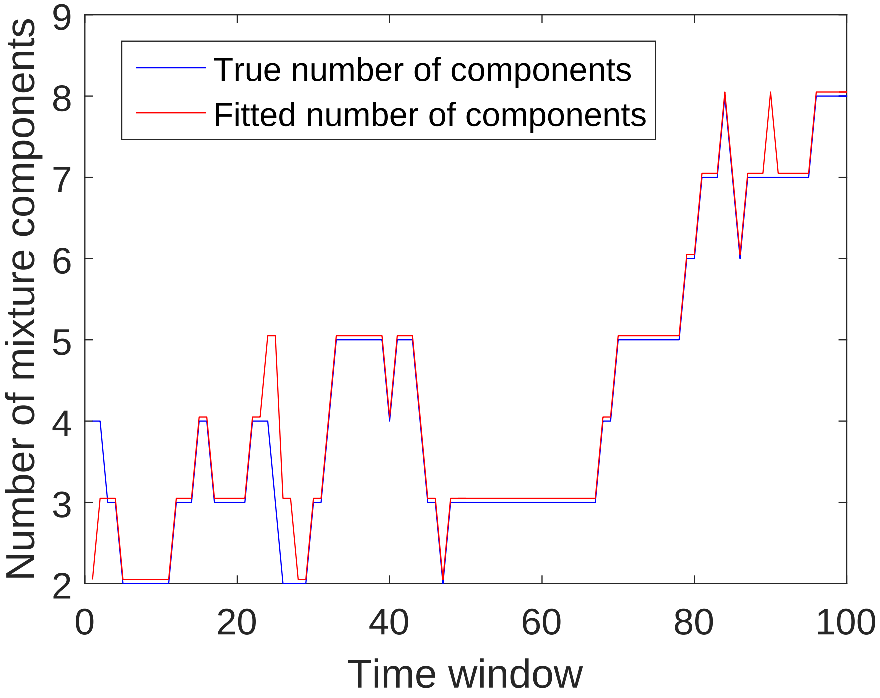


**Fig. 6.4** An example estimated number of components versus the true number of components over 100 time windows for the dataset in SDS2-a with 1000 vehicles per time-window.

Dataset SDS1 had a perfect model-order fit for all trials, so we focus on SDS2 where we have non-perfect fits. The estimate for the model order of SDS2-a is shown in Fig. 6.4 for 1000 vehicles per time-window in one of the 10 random trials. The other fitted models of SDS2-a and SDS2-b also showed that the number of estimated components is not a perfect fit and varies from what is shown in Fig. 6.4 (i.e. the error in the estimated number of components is not the same through all the different dataset sizes). We attribute this increase in the error of the number of mixture components to two sources. First, the model order complexity can be quite large (up to 8 mixture components). Also, since the average vehicle observation length is smaller than in the first dataset, mis-classification of vehicles to the wrong component (or creation of false components) is more likely.

We also investigate how the algorithm performs when selecting the number of mixture model components for varying lengths of the vehicle observation sequences. In Fig. 6.5 we see that, after an initial transient period, the algorithm maintains the estimate of the mixture model order more easily when there are longer observation sequences (as in SDS2-b). Selecting the correct number of components is an important step in the estimation of the overall mixture model however we want to further investigate the error in the model's parameter estimates as well.

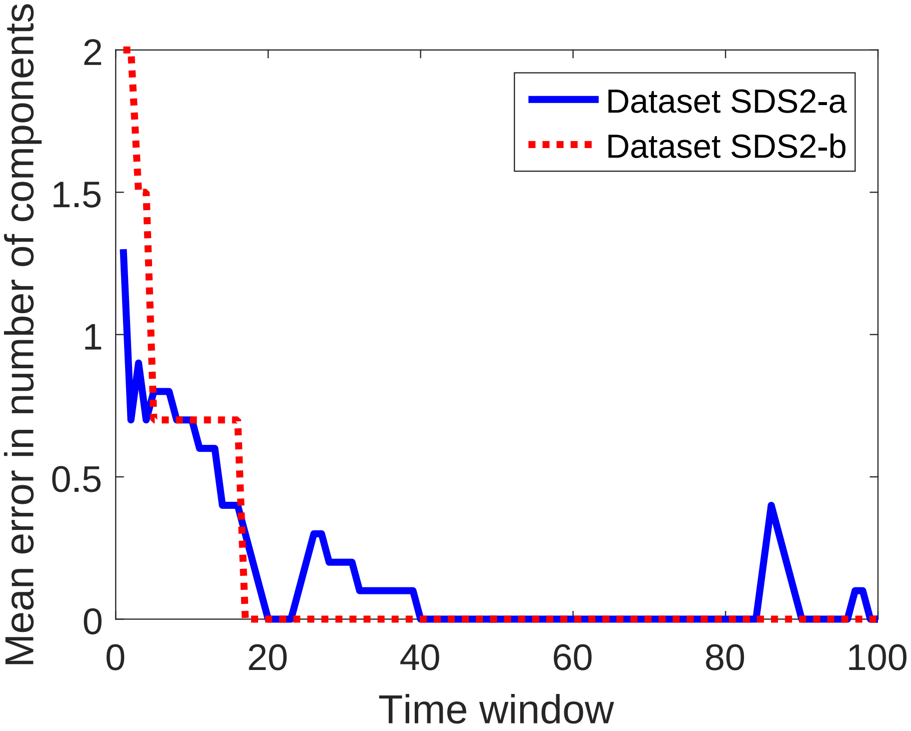


**Fig. 6.5** The average absolute error in the number of estimated components for SDS2-a and SDS2-b using 5000 vehicles per time-window and averaged over 10 trials

Next we examine the fidelity of the estimated model parameters. We quantify the accuracy using two approaches discussed below: the marginal $\ell_1$ error, and the component-wise $\ell_1$ error.

### Marginal $\ell_1$ error

The first approach to error quantification is based on the true and estimated marginal models, computed from the mixture model as

$$P^{(\text{marg})}(t) = \sum_{m=1}^{M(t)} \alpha^{(m)}(t) P^{(m)}(t) \tag{6.21}$$

$$\text{and } \pi^{(\text{marg})}(t) = \sum_{m=1}^{M(t)} \alpha^{(m)}(t) \pi^{(m)}(t). \tag{6.22}$$

As a benchmark, we estimate a one-component model using the maximum a posteriori (1-MAP) estimate of the marginal distribution which is

$$\hat{p}_{j,k}^{(\text{1-MAP})}(t) = \frac{n_{j,k} + \hat{p}_{j,k}^{(\text{1-MAP})}(t-1)}{1 + \sum_{k=1}^{|\Omega|} n_{j,k}} \tag{6.23}$$

$$\text{and } \hat{\pi}_j^{(\text{1-MAP})}(t) = \frac{n_j + \hat{\pi}_j^{(\text{1-MAP})}(t-1)}{1 + \sum_{j=1}^{|\Omega|} n_j}\ , \tag{6.24}$$

where $n_{j,k}$ is the total number of observed transitions from $j$ to $k$ and $n_j$ is the number of observations of starting state $j$. For the first time-window ($t = 1$), we use the same global prior as with our proposed model.

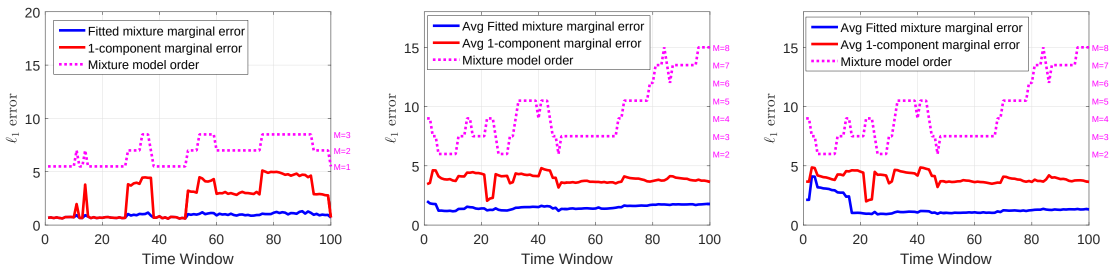
We measure accuracy in terms of the $\ell_1$ error,

$$\ell_1(\hat{p}, \hat{\pi}, p, \pi) = \sum_{j=1}^{|\Omega|} \sum_{j=1}^{|\Omega|} |\hat{p}_{j,k} - p_{j,k}| + \sum_{j=1}^{|\Omega|} |\hat{\pi}_j - \pi_j|, \tag{6.25}$$

where $\hat{p}$ and $\hat{\pi}$ are the estimated marginal transition matrix and initial state distribution, and $p$ and $\pi$ are the true marginal transition matrix and initial state distribution. This reduces the error to a single scalar value and has the benefit that it is straightforward to calculate even when the model order is not estimated correctly.

Recall that the total variation distance between two probability densities is at most two. The metric defined in equation (6.25) is the sum of $|\Omega| + 1$ total variation distances (one for the initial state distribution, and one for each row of the transition matrix). In all of the simulated datasets there are $|\Omega| = 25$ states (i.e., sensors), so $\ell_1(\hat{p}, \hat{\pi}, p, \pi) \leq 52$.

Fig. 6.6 shows the $\ell_1$ errors among the marginal distributions on all three simulated data sets, for estimates generated using the proposed mixture model as well as using a single-component Markov chain estimated using the MAP approach described above. In general, the mixture model gives a significantly lower error than using a simple 1-component Markov model. Note that the time windows where the errors of the two models coincide in SDS1 occur when the true model order is $N(t) = 1$. The $\ell_1$ marginal error of the mixture estimate also appears to be correlated with the model order. In these simulations, the total number of observations (vehicles) per time-window is constant, so when the model order is higher there are fewer observations per component, resulting in a higher error. Also observe that the steady-state marginal error of the mixture model is slightly higher for SDS2-a than it is for SDS2-b. This is expected, since there are fewer observations per vehicle, on average, in SDS2-a.



(a) SDS1 (b) SDS2-a (c) SDS2-b

**Fig. 6.6** (a) The $\ell_1$ error of the estimated marginal Markov chain distribution versus the true underlying distribution for SDS1 with 5000 vehicles per time-window. Also shown is a MAP estimated marginal distribution and the mixture model superimposed. (b) The $\ell_1$ error of the estimated marginal Markov chain versus the truth with SDS2-a and 5000 vehicles per time-window. (c) The same plot as (b) for dataset SDS2-b.

We next examine how the error in the marginal distribution changes as a function of the number of observed vehicles, $N(t)$. Fig. 6.7 shows that the error of the mixture model goes to zero significantly faster than that of the 1-MAP as $N(t)$ increases on the SDS2-b dataset. Datasets SDS1 and SDS2-a exhibit the same behaviour, only differing in the scale of the plots, and they are therefore not shown here.

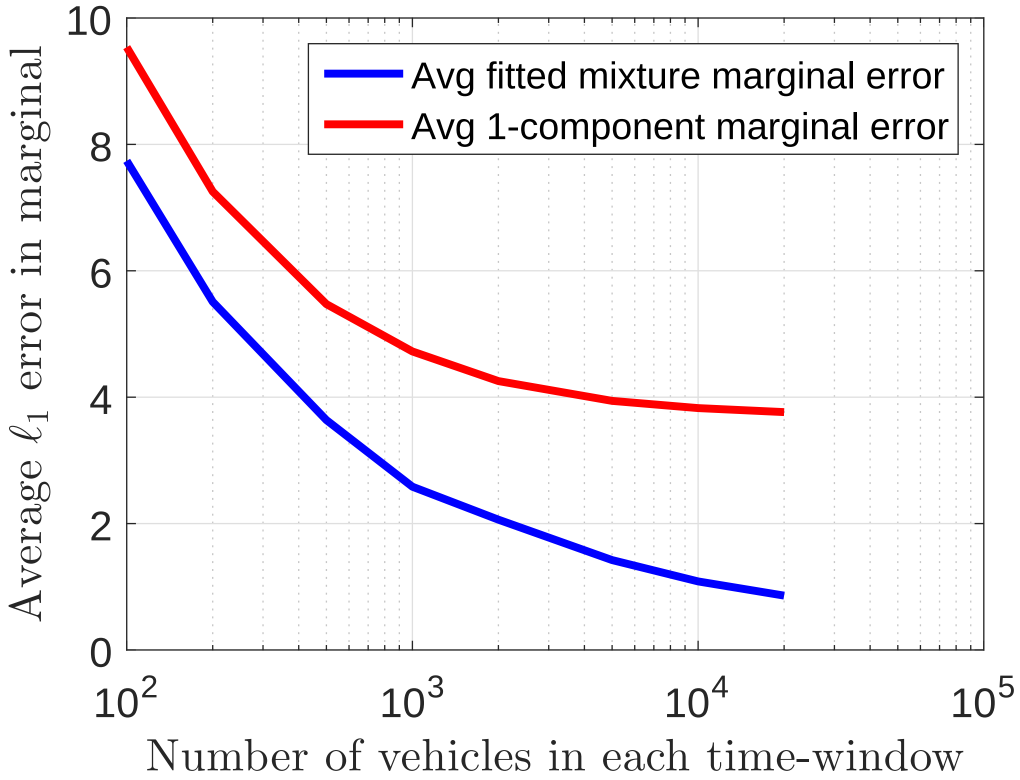


**Fig. 6.7** The average $\ell_1$ error in the estimate of the marginal distribution estimated using the proposed method and a MAP estimate for SDS2-b with a varying number of vehicles in each time-window. The number of vehicles is on a log-scale.

### Component-wise $\ell_1$ errors

When the model order is estimated correctly, we also examine the average component-wise errors. In this case, the labels (indices) of components in the true and estimated models may not coincide, so to assess the component-wise error we first perform a matching as follows. Let $(\pi^{(m)}, P^{(m)})$ denote the model parameters associated with component $m$ of the true mixture model, and let $(\hat{\pi}^{(m')}, \hat{P}^{(m')})$ denote the parameters of component $m'$ of the estimated mixture model. Note that we have dropped the time index $t$ to simplify the notation. Suppose that there are $M$ mixture components overall, and let $\mathbb{S}_M$ denote the set of permutations of $M$ elements. We first find the permutation $\sigma^\star$ that minimizes the total $\ell_1$ error,

$$\sigma^\star = \arg\min_{\sigma \in \mathbb{S}_M} \sum_{m=1}^{M} \ell_1 \left( \hat{P}^{(\sigma(m))}, \hat{\pi}^{(\sigma(m))}, P^{(m)}, \pi^{(m)} \right).$$

The solution $\sigma^\star$ is computed using the Hungarian algorithm. Then we compute the average component-wise $\ell_1$ errors for the initial state distributions, $\pi^{(m)}$, the transition matrices, $P^{(m)}$, and mixture weights, $\alpha^{(m)}$, separately. For example, for the initial state distribution, the average component-wise $\ell_1$ error is

$$\frac{1}{M} \sum_{m=1}^{M} \sum_{i=1}^{|\Omega|} \left| \pi_i^{(m)} - \hat{\pi}_i^{(\sigma^\star(m))} \right|.$$

Note that it is not straight-forward to assess component-wise errors when the model order has not been accurately estimated. For example, if the estimated model has more components than the true model, it may be that two or more components in the estimate should be associated with a single component in the true model, or alternatively, it may be that one component in the estimated model should be partially associated with two or more components in the true model. Hence, we only report component-wise errors for time-windows where the model order has been correctly estimated.

Fig. 6.8 shows the average component-wise errors for the SDS2-a dataset with $N(t) = 5000$ observations. The same trends were observed for the other two datasets so they are not shown here. In general, we observe that the average component-wise errors also correlate well with the model order. As the model becomes large (e.g., time-windows 80–100), the average component-wise $\ell_1$ also becomes large, suggesting that the estimated components do not precisely match those of the true model. This may again be attributed to the fact that, in the simulation, when the model order is higher there are fewer observations per component. However, as we will see in the experiments on real data reported below, this error does not necessarily imply that predictions made using the estimated mixture are inaccurate.

In addition to computing the average component-wise errors, we also computed the standard deviation of the component-wise errors. These were all many orders of magnitude smaller than the mean errors, so they are not shown in the figure.

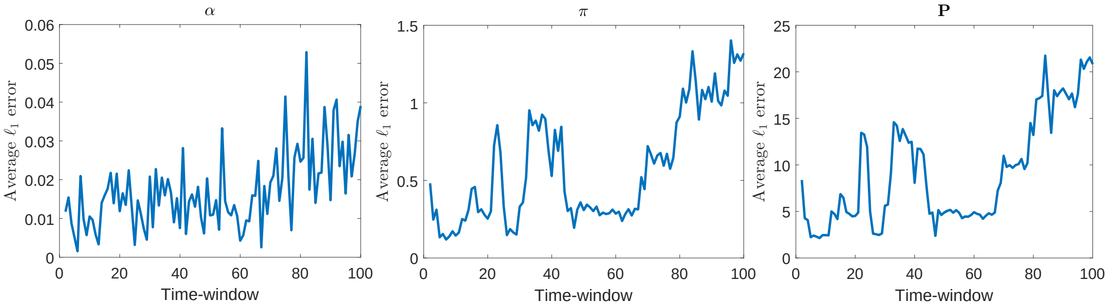


**Fig. 6.8** Average component-wise $\ell_1$ errors of the mixture weights $\alpha^{(m)}$, initial state distributions $\pi^{(m)}$, and transition matrices $P^{(m)}$, on SDS2-a for $N(t) = 5000$ observations per time-window.

### 6.7.2 Computation Time

The real benefit of Algorithm 1 compared to some of the non-deterministic approaches, such as Gibbs sampling in [7], is the computation time and complexity. Even with 20,000 observations in each time-window, the computation time never exceeds 30 seconds per iteration on a laptop with a quad-core 2 GHz Intel Core i7 processor and 8 GB of RAM. This makes this algorithm practical for real-time applications where estimation of a complex mixture model is necessary without having unlimited time or computational resources available. In addition to computation time, the algorithm is deterministic and requires no random restarts or bootstrapping.

### 6.7.3 Comparison to Gibbs Sampling

The MCMC method described in Stephens [7] was also applied to the generative model in Section 6.3. This algorithm was run on a single time-window ($t = 35$) of SDS1 where the true mixture has 3 components and 10,000 vehicles are observed. The method was run for

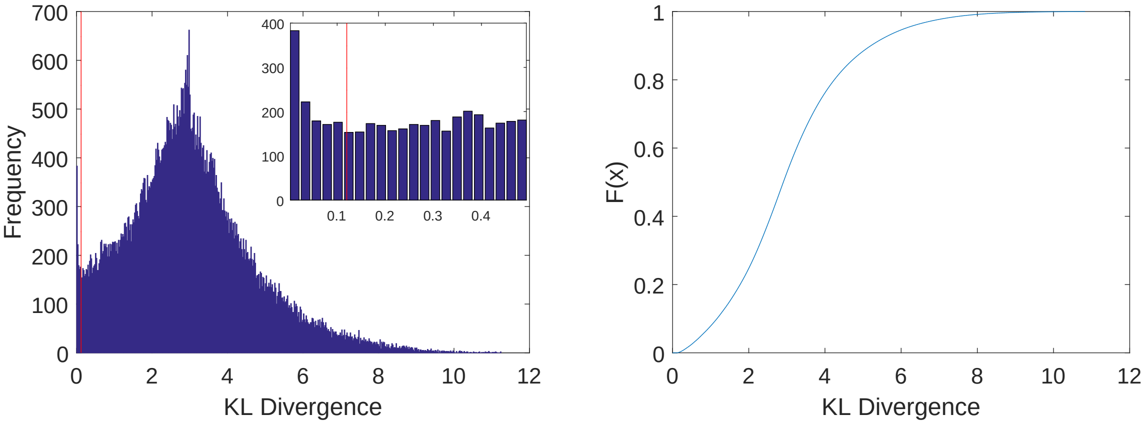
5,000 burnin iterations and 50,000 sampling iterations. The best candidate model, with the correct number of components, chosen in the sampling had a $\ell_1$ marginal error of 20.68, which is significantly larger than the $\ell_1$ error of 0.8444 obtained using Alg. 1. Upon further inspection, we observe that the Gibbs sampler appears to sample poor candidate models due to the high-dimension of the parameter space (we conclude this because most sampled models are immediately rejected using the approach described in [7]). Consequently the estimated model order alternates between two and three specific mixture components and does not explore the space of possible candidate components well. Based on these observations we conclude that the class of MCMC models will fail to estimate traffic models with such a high-dimensional and time-varying parameter space. Therefore we do not consider this approach further.

### 6.7.4 Real Data Analysis

Next we run Alg. 1 on a real LPR dataset provided by a corporate partner. The dataset is a collection of LPR reads from a network of 20 LPR cameras over a period of 31 days and 2 hours. We take each time-window to be an hour long so there are a total of 746 time-windows; the effect of time-window length is discussed further in Section 6.7.6. In every time-window we create observation sequences for the vehicles which appear in that window. From the data in each time-window, we estimate the mixture model using Alg. 1. The threshold parameters $L_\alpha$ and $L_{KL}$ are set to

$$L_\alpha = \frac{2}{|X(t)|} \text{ and } L_{KL} = 0.12 \tag{6.26}$$

so that the trimming of weak components deletes any component with less than two vehicles assigned to it. We estimate the KL divergence threshold, $L_{KL} = 0.12$, by running Alg. 1 once with $L_{KL} = 0$ so that we essentially over-fit as much as possible. We then compute the KL divergence between all pairs of components and show the density of the KL divergences in Fig. 6.9(a). Here we can clearly see another peak near 0, which we hypothesize is appearing due to the same reason as in the simulated case. Those components with very small divergence are being over-fit and should be merged into other existing components. In Fig. 6.9(a) we draw the red line to demonstrate where we place the $L_{KL}$ threshold to trim off that lower peak, which is at 0.12. We then plot the CDF of the resulting KL divergences when Alg. 1 is run with $L_{KL} = 0.12$ which demonstrates that the lower peak is no longer present. There is no ground-truth distribution for this dataset, so validating the KL divergence threshold beyond this is not feasible.



(a)

(b)

**Fig. 6.9** (a) A histogram of the distribution of the KL divergence values when setting the KL divergence threshold to 0 ($L_{KL} = 0$). The red line ($y = 0.12$) denotes the proposed threshold value, $L_{KL}$, to remove the lower-component, over-fit estimated DTMCs. (b) The CDF of the distribution of the distribution of the KL divergences in Fig. 6.9(a) once we impose our threshold of $L_{KL} = 0.12$. Note that there is no "hump" near zero meaning we have removed unnecessary, over-fit components.

The estimated number of components is shown in Fig. 6.10(a) where we see that the estimated mixture model is quite complex, having up to 18 components. However upon closer inspection, we observe a cyclic behavior, as shown in Fig. 6.10(b), where the model is most complex at rush-hour in the morning and afternoon and least-complex during the late evening and early morning. This agrees with one's intuition about traffic simply due to the fact that at rush-hours, the volume of traffic traversing the roads will be greatest therefore the largest number of paths are likely to be observed.

### 6.7.5 Mixture Model Predictive Ability

There is no ground-truth model available for the real dataset. To assess the quality of the estimated mixture model, we consider the following prediction task. In each time-window, we split the observations (i.e., per-vehicle sequences) into a training set and a test set. Specifically, 20% of the observations are randomly (uniformly) sampled to form a test set

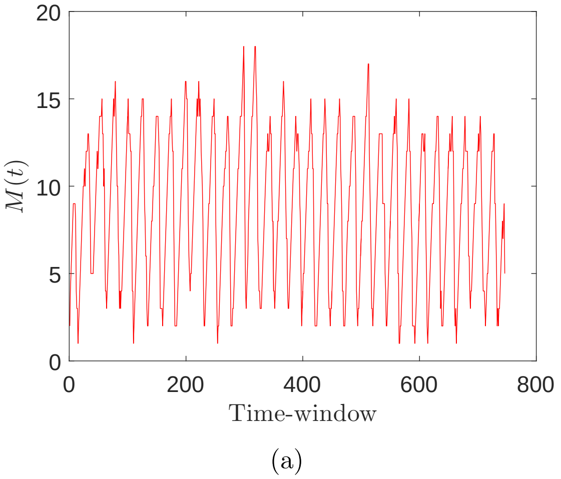


**Fig. 6.10** (a) The estimated number of components for the real LPR dataset for the 746 time-windows. (b) A portion of the estimated number of components on the real dataset in Fig. 6.10(a) zoomed in to view the cyclic behavior.

for the time-window, and the remaining 80% of the samples form the training set. For each time-window, we use the training data to update the estimated model parameters. Then, for each test point (the sequence of observations of a single vehicle), the task is to predict the last state where the vehicle will be observed given all observations but the last one; i.e., for a test sequence $(x_1, x_2, \ldots, x_n)$, we must predict the value of $x_n$ given $x_1, \ldots, x_{n-1}$. We allow the prediction to be in the form of a probability density over the set $\Omega$, which can also be viewed as the parameters of a multinomial distribution.

We again compare the performance of the proposed mixture model, fit using the automatic hard EM algorithm, with that of a single-component (1-MAP) model. For the 1-MAP model, the prediction is simply given by row $x_{n-1}$ of the estimated transition matrix. For the mixture model, we use the first $n-1$ observations of each test vehicle to determine which of the estimated mixture components best explains the observations, and then the prediction is given by row $x_{n-1}$ of the transition matrix of that specific component. As a baseline, we also compare with a naive, uniform prediction that places mass $1/|\Omega|$ on every sensor (i.e., every state in $\Omega$).

The ideal predicted distribution would have unit mass on position $x_n$, and zero everywhere else. To measure the quality of the predictions given by the mixture model and

1-MAP, we compute the multinomial log-loss

$$LL_{MN}(p^{(pred)}) = -\frac{1}{N_{\text{test}}(t)} \sum_{i=1}^{N_{\text{test}}(t)} \sum_{j=1}^{|\Omega|} y_{i,j} \log\left(p_{i,j}^{(pred)}\right),$$

where $N_{\text{test}}(t)$ is the number of testing samples in time-window $t$, $y_{i,j}$ is the indicator that the final observation of the $i$th test vehicle was made at sensor $j$, and $p^{(pred)}$ is the predicted probability that the $i$th test vehicle was last observed at sensor $j$, given the initial observations of that vehicle. The multinomial log-loss of a particular training instance is equal to zero if the prediction puts all of its mass on the correct sensor, and it increases as the predicted probability of the correct state decreases.

Fig. 6.11 shows the multinomial log-losses for each time window. The boxplots show that the proposed model ("Mixture fit") has a lower median and much lower spread of $LL_{MN}$, and thus its predictions are closer to the truth for a larger volume of the datapoints than predictions made by the single-component MAP Markov model ("1-component fit"). This is likely due to the proposed model providing more accurate estimates for vehicles on less-frequently observed routes; these can be represented as low-weight mixture components in the mixture model, while the corresponding data gets washed out in the 1-component model. The vast majority of predictions made by the mixture model have lower loss than the naive prediction, whereas the upper whisker of the 1-component fit lies above the naive loss line.

Fig. 6.12 shows performance as a function of the number of vehicle observations ($n-1$ in the discussion above) available before making a prediction. The mixture model clearly shows a benefit when making predictions from a few initial observations. When the number of observations of a vehicle gets large, both the mixture and 1-component fits offer comparable performance.

### 6.7.6 Time-Window Length Selection

The selection of an appropriate time-window length when estimating the time-varying mixture model is critical to the algorithm's performance. If the time-window length is set too large, then the true underlying distribution of traffic may change significantly within one time-window. On the other hand, if the time-window is too short than there may not be a sufficient number of observations available to estimate model parameters within each

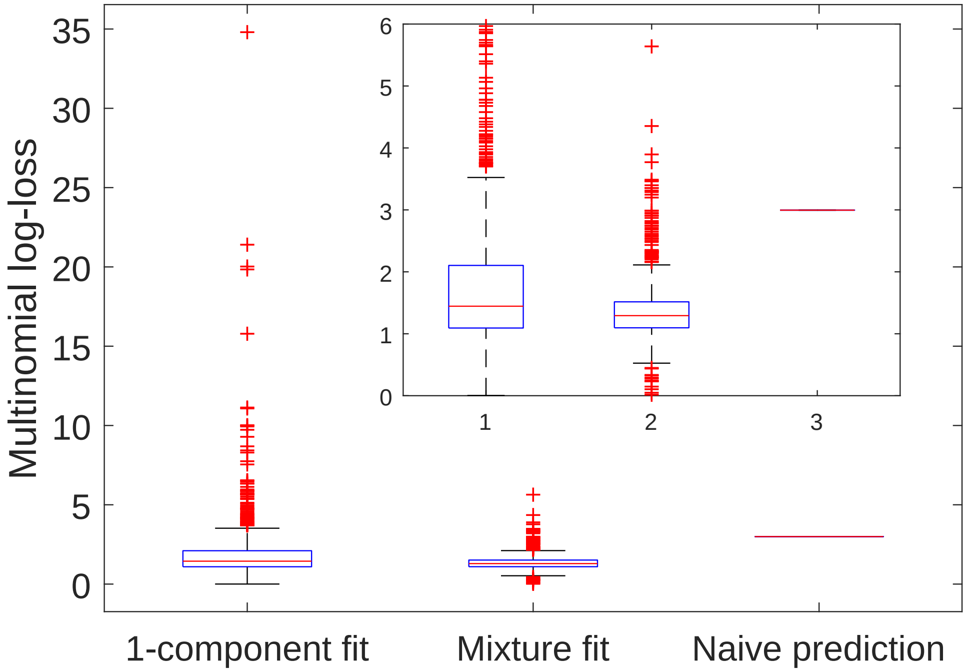


**Fig. 6.11** The Multinomial log-loss using three different estimates: (1) A standard 1-component MAP estimate, (2) Our proposed mixture model and Automatic Hard EM algorithm, and (3) Simply assigning $\frac{1}{n}$ to all transition matrix weights (i.e. naive prediction). This plot includes a zoomed-in region in the top right to better see the boxes comparison.

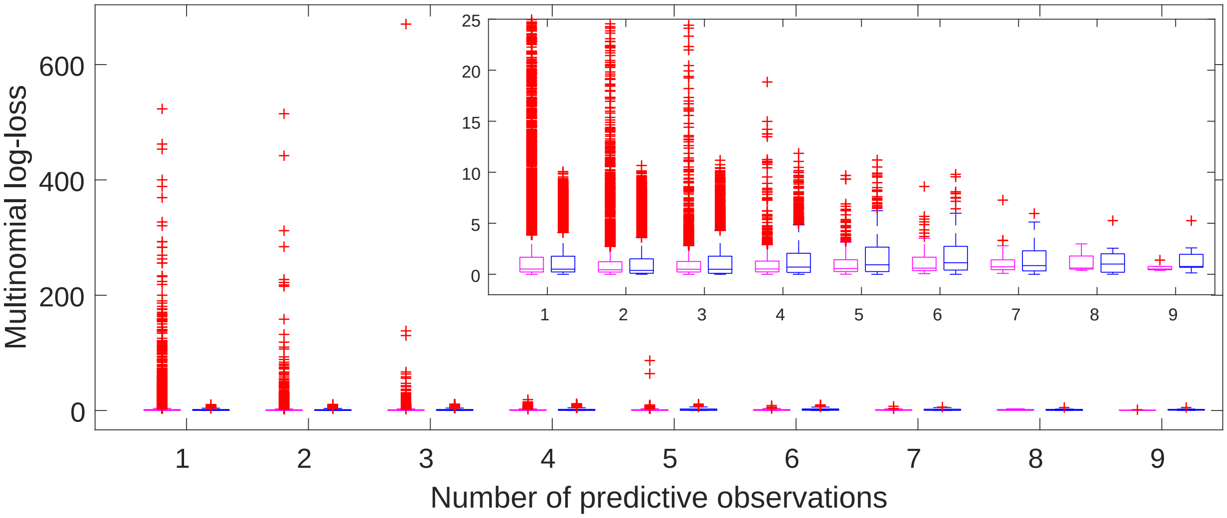
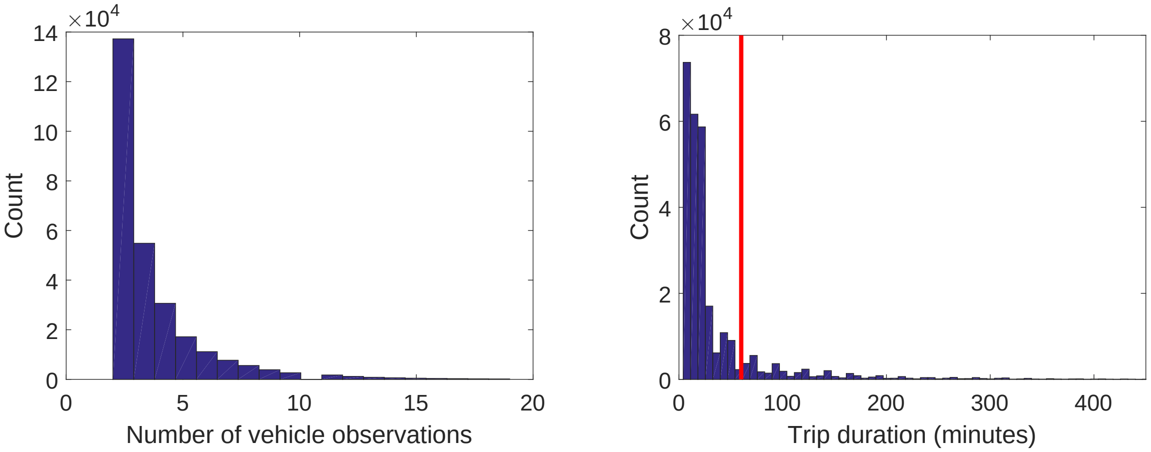


**Fig. 6.12** The individual Multinomial log-losses per-vehicle versus the number of observations available to compute the predictive multinomial density. In each group, the left box plot (cyan) is the prediction of the 1-component model and the right (blue) is our proposed mixture model. The upper corner shows a zoomed-in portion of the boxplots focusing on the range of multinomial log-loss between 0 and 25.

window.

Towards selecting a reasonable time-window length for the real data experiments, we explore some related characteristics of this dataset. We begin by organizing the dataset by vehicle, and segment the per-vehicle observations into individual trips. Recall that the entire dataset spans a period of one month, and many of the vehicles appearing in the dataset are observed multiple times during that period. The maximum distance between two sensors is roughly 40 km. Assuming a minimum speed of 10 km/hr, the maximum time it would take observe a vehicle at one of these sensors after it was observed at the other is four hours. Hence, whenever the consecutive observations of a vehicle are more than four hours apart we consider it the beginning of a new trip.

The number of observations per trip and the duration of each trip are shown in Fig. 6.13. In total there are 27,7798 observed vehicle trips. The median number of observations per trip is 3.5183, and the median trip duration is 36.0585 minutes. The red line in Fig. 6.13(b) indicates a duration of 60 minutes, which is the value we use for the trip-window duration. In this case, 86.08% of the observed trips have a duration of 60 minutes or less.



(a)

(b)

**Fig. 6.13** (a) Histogram of the distribution of number of vehicle observations per trip in the real dataset. Note that vehicles that are only observed once are excluded from the histogram for clarity. (b) Histogram of trip durations (the time from the initial observation of a vehicle to the final observation in a single trip). The superimposed red line denotes the 60-minute (1 hour) mark which we have used as the window-size for the results reported in the paper.

We also repeat the prediction experiment described in Section 6.7.5 with models trained using both 30 minute and 2-hour time windows. For each scenario, we estimate a time-varying mixture of Markov chains using the proposed approach, as well as a time-varying single-component Markov chain estimated as the MAP with the model from the previous time window as the prior. The results of these trials are shown in Fig. 6.14. Observe that there is a slight increase in the variability of the performance (multinomial log-loss) of the proposed mixture model as the time-window length decreases. However there is a much greater increase in the variability (as evidenced by the increase in the whisker locations) in the single-component estimated model. This suggests that the mixture model is less sensitive to the choice of time-window length than a single-component model.

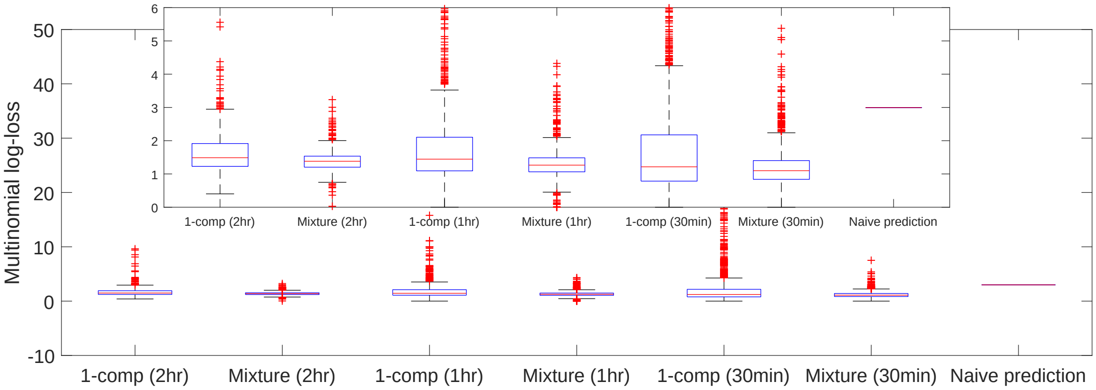


**Fig. 6.14** Boxplots of the multinomial log-loss for three different time-windows (30 minutes, 1 hour, 2 hours). The first, third, and fifth plots are using the single-component MAP estimated DTMC models while the second, fourth, and sixth plots are using the time-varying mixture of DTMCs proposed in this paper. The last plot is a "naive" fit using a single DTMC with $1/|\Omega|$ in all cells of the transition matrix. For clarity, a zoomed-in portion of the boxplots is provided.

## 6.8 Conclusion

This article proposes a novel approach to fitting time-varying mixture models, with an specific application to discrete-time Markov chains for the problem of traffic estimation. The Markov model specified in this paper enables complex, time-varying, network models to be modelled using AVI data. We then go on to define an algorithm to estimate the parameters of this time-varying mixture model and specifically apply it to AVI data. The performance of the algorithm is assessed using simulated data and data from a real deployment. The results indicate that the proposed model provides more accurate predictions of vehicle trajectories than the single component Markov model or naive estimates, especially when the prediction is made using fewer initial observations.

### 6.8.1 Future Work

#### Errors in Data Source

A future avenue of research stemming from this work is the modelling of errors in the incoming data stream. We currently assume that the observed vehicle IDs are error free. For license plates, the errors which can occur are insertions, deletions, and replacements of characters in the license plate. A probability distribution over the possible errors could be postulated and then transition count matrices for each vehicle could be updated based on the likelihood of the recorded license plate. This would allow for a natural extension to model the input errors in the recording of license plates.

#### Adapting the Time-Window Length

Another extension would be to dynamically adapt the time window length. To motivate why, consider normal traffic in an urban environment. Traffic patterns typically have two very high volume periods, during the morning and afternoon rush-hours. At these times, the largest variation in traffic patterns are observed (see, e.g., the real-data example in Section 6.7.4). On the other hand, traffic patterns observed in the middle of the night are likely remain unchanged for many hours since few vehicles are travelling at that time.

Therefore the mixture models which are estimated to be governing the current traffic patterns may be valid for longer or shorter times based on the observed traffic patterns and volumes at different times of the day. One approach could be to run a sequential hypothesis test in parallel to the estimation scheme, to detect when the current model no longer adequately fits or explains incoming observations. When the fit to current data becomes sufficiently poor then an adaptive model update would be triggered.

# References for *Time-Varying Mixtures of Markov Chains: An Application to Traffic Estimation*

[1] R. Smeed, “Road pricing: the economic and technical possibilities,” British Transport and Road Research Laboratory, Tech. Rep., 1964.

[2] J. Ren, X. Ou, Y. Zhang, and D. Hu, “Research on network-level traffic pattern recognition,” in *Proc. on Intelligent Trans. Sys.* IEEE, 2002, pp. 500–504.

[3] S. Lawlor, T. Sider, N. Eluru, M. Hatzopoulou, and M. Rabbat, “Detecting convoys using license plate recognition sensors,” *IEEE Trans. Signal and Information Processing over Networks*, vol. 2, no. 3, pp. 391–405, Sep. 2016.

[4] M. Nanni, R. Trasarti, B. Furletti, L. Gabrielli, P. V. D. Mede, J. D. Bruijn, E. D. Romph, and G. Bruil, *Transportation Planning Based on GSM Traces: A Case Study on Ivory Coast.* Springer International Publishing, 2014, pp. 15–25.

[5] Maine Turnpike Authority. (2016, July) Welcome to the Maine turnpike’s E-ZPass program. [Online]. Available: https://ezpassmaineturnpike.com/EZPass/

[6] F. Caron, M. Davy, and A. Doucet, “Generalized Polya urn for time-varying Dirichlet process mixtures,” in *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence, UAI*, July 2007.

[7] M. Stephens, “Bayesian analysis of mixture models with an unknown number of components- an alternative to reversible jump methods,” *The Annals of Statistics*, vol. 28, pp. 40–74, 2000.

[8] M. Abhijith, P. Ghosh, and K. Rajgopal, "Multi-pitch tracking using Gaussian mixture model with time varying parameters and grating compression transform," in *2014 IEEE Inter. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, May 2014, pp. 1473–1477.

[9] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. on Automatic Control*, vol. 19, no. 6, pp. 716 – 723, 1974.

[10] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 1978.

[11] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[12] A. Corduneanu and C. M. Bishop, "Variational Bayesian model selection for mixture distributions," in *Intl. Conf. on Artificial Intelligence and Statistics.* Morgan Kaufmann, Jan. 2001, pp. 27–34.

[13] M. A. Figueiredo and A. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, March 2002.

[14] J. Chen and A. Khalili, "Order selection in finite mixture models with a nonsmooth penalty," *J of the American Statistical Association*, vol. 103, no. 484, pp. 1674–1683, 2008.

[15] J. Verbeek, N. Vlassis, and B. Krose, "Efficient greedy learning of Gaussian mixture models," *Neural Computation*, vol. 15, no. 2, pp. 469–485, 2003.

[16] M. G. Singh and H. Tamura, "Modelling and hierarchical optimization for oversaturated urban road traffic networks," *International Journal of Control*, vol. 20, pp. 913–934, 1974.

[17] B. S. Kerner, S. L. Klenov, and D. E. Wolf, "Cellular automata approach to three-phase traffic theory," *Journal of Physics A: Mathematical and General*, vol. 35, no. 47, pp. 9971–10 013, November 2002.

[18] A. Peterson, “The origin-destinaton matrix estimation problem - analysis and computations,” Ph.D. dissertation, Linköping University, Linköpings universitet, SE-601 74 Norrköping, Sweden, 2007.

[19] N. V. D. Zijpp, “Dynamic OD-matrix estimation from traffic counts and automated vehicle identification data,” *Tranprn Res. Record: J. of the Transprn Res. Board*, vol. 1607, pp. 87–94, 1997.

[20] S. Lawlor and M. G. Rabbat, “Estimation of time-varying mixture models: An application to traffic estimation,” in *2016 IEEE Wrksp. on Stat. Signal Process. (SSP)*, June 2016.

[21] G. Celeux and G. Govaert, “A classification EM algorithm for clustering and two stochastic versions,” *Computational Statistics & Data Analysis*, vol. 14, no. 3, pp. 315–332, 1992.

[22] S. Kotz, N. Balakrishnan, and N. Johnson, *Continuous Multivariate Distributions. Volume 1: Models and Applications.* Wiley, 2000, vol. 1.

[23] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. of the Royal Statatistics Society Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[24] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics.* Berkeley, Calif.: University of California Press, 1967, pp. 281–297.

[25] C. J. Wu, “On the convergence properties of the EM algorithm,” *The Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 1983.

[26] H. White, “Maximum likelihood estimation of misspecified models,” *Econometrica*, vol. 50, no. 1, pp. 1–25, 1982.

# Chapter 7

# Origin Destination Matrix Estimation

Chapter 6 proposed a generative, time-varying model for mixtures of Markov chains which can be used to estimate traffic routes in an urban environment using AVI sensors. This model proposed a unique way of estimating multiple different routes between AVI sensors in an urban environment. After discussing the model and proposed outputs with collaborators from the transportation engineering field (Drs. N. Eluru and M. Hatatzopoulou) the question arose of how to relate this model to traditional models from the transportation research field.

A good candidate model which arose are the estimation origin-destination (OD) matrices, also referred to trip-tables, which denote the number of vehicles traversing from defined origins and terminating at defined destinations. However in a traditional Markov model, as is utilized in Chapter 6, the vehicle's terminating state is not considered, so determining the destination of a vehicle is impossible, since it appears as though once entering the Markov chain they continue to transition through the chain forever.

In order to estimate an OD matrix (or matrices), we needed to model the terminating state in the model so we propose adding a Bernoulli distribution depending on the origin state. Therefore at every transition in the vehicle's path, the vehicle flips a biased coin which states if the vehicle terminates at the specified state given its starting state. This modifies the traditional discrete-time Markov chain to account for the terminating state of a vehicle.

Now with this adjusted statistical model, once the estimation algorithm is run, the resulting estimated Markov chains with terminating probabilities can be transformed into

OD estimates. However by computing the OD estimates for each component, the model will naturally separate OD estimates based on the routes being followed. Normally an OD matrix estimation technique only estimates a single, summary OD matrix and contains no information about the routes being followed through the road network. The new model presented in Chapter 8 can get accurate OD estimates and also contains information about the specific routes being followed for specific OD pairs. This results in a natural decomposition into a mixture of OD matrices which are separated based on which OD pairs follow similar routes.

In addition to the natural separation of OD matrices into decomposed matrices, the method has the added benefit of not requiring a prior OD matrix which is needed in classical methods [Pet07b]. This prior matrix is typically estimated using expensive, time-consuming methods and may only remain valid for a short time. The method presented in the following manuscript addresses this issues with a new model independent of a prior OD matrix.

# Chapter 8

# Paper: A Mixture Model Approach to Origin-Destination Matrix Estimation with Routing Information

[1]We propose a new approach to estimating the *origin-destination* (OD) matrix using a time-varying mixture of discrete-time Markov chains with an additional term to describe the termination state of a vehicle. We then propose a variation of the expectation-maximization algorithm, with model-order selection, to fit such a mixture model to observations including data from automatic vehicle identification sensors, transforming it into a mixture of OD matrices. We demonstrate that the mixture of OD matrices can recover the full OD matrix by simply computing the marginal OD matrix of the mixture model, and we also show that the mixture of OD matrices contains additional information about the routes being travelled for specific OD pairs. We demonstrate the accuracy of the algorithm on a simulated small-scale network as well as on a larger-scale network representing downtown Montréal, Canada.

[1] The contents of this chapter are drawn from the manuscript: S. Lawlor, N. Eluru, M. Hatzopoulou, and M. Rabbat, “A mixture model approach to origin-destination matrix estimation with routing information,” submitted.

## 8.1 Introduction

### 8.1.1 The Origin-Destination Matrix

The *origin-destination* (OD) trip table is a matrix of counts of the number of vehicle trips made between different pairs of origins and destinations. OD matrices summarizing travel demand for a particular time window are commonly used in urban planning and traffic management. For example, transportation planners and traffic engineers consider the OD matrix when making decisions related to issues such as infrastructure upgrades and parking management [1].

Estimation of the OD matrix has been widely studied, with various models applied. Work in this field traditionally starts with an out-dated OD matrix of historical value, typically generated through a user survey. Then sensor inputs (e.g., traffic counts from various locations) are used to update the out-of-date OD matrix to an applicable value for the current time-frame.

This paper proposes a method for OD matrix estimation that does not require that an out-dated OD matrix be provided to compute the estimated OD matrices. In addition to the estimated OD matrix, the method proposed in this paper automatically clusters OD pairs into distinct components which behave similarly based on the observed traffic. We validate the proposed model and inference algorithm on small-scale simulated networks as well as on a regional road network with traffic volumes derived from a stochastic user equilibrium traffic assignment model.

### 8.1.2 Previous Work

#### OD Matrix Estimation Using Traffic Counts

OD matrix estimation has been widely studied. The most prevalent formulation in the literature uses traffic count data. In this context, one of two types of estimation schemes are typically applied to the problem: parameter calibration techniques or matrix estimation methods [2]. Parameter calibration techniques use linear or non-linear models, often assuming a gravity-type flow pattern, to estimate the OD matrix entries. The gravity model is a derivative of Newton's law of gravity which is used to predict how two regions influence each other via migration [3]. However, these methods require prior information about the OD pairs being estimated which is updated given the observed trip counts [4]. This prior information is typically encoded as another OD matrix based on survey data, which has the drawback that it is typically only valid for a short period of time. Moreover, survey data is expensive and time-consuming to gather [2]. The survey data may contain information about the socio-demographic information at the zonal level, their work location, and mode of transportation to their job.

The alternate class of methods use matrix estimation techniques, also starting with a prior OD matrix, and compute an updated estimate based on the prior OD matrix and traffic counts. These methods attempt to minimize the distance between the prior OD matrix and a target OD matrix in order to reproduce observed traffic counts [5]. A large body of work has been applied to this problem and multiple solutions for the matrix estimation problem have been proposed. A review of the main contributions to this field and an examination of the matrix estimation problem given only a *partial* starting OD matrix are provided in [6]. For a recent overview of the literature on OD matrix estimation methods that use traffic count data see [7].

### OD Matrix Estimation Using Automatic Vehicle Identification Data

The class of problems and solutions discussed thus far only consider the OD matrix estimation problem using traffic count data. Traffic counts can be gathered from a variety of sensors, ranging from inductive loops laid in the road network to traffic counting techniques using video recordings [8]. Recent advances in traffic sensing technology include sensors that provide more detailed information than simple vehicular counts. *Automatic vehicle identification* (AVI) sensors record a unique identifier of a vehicle when it passes by the sensor. Examples of AVI sensors include Bluetooth sensors and cameras with license plate recognition ability. Bluetooth sensors record the Bluetooth address of the in-vehicle entertainment system and/or the operator's telephone. *License plate recognition* (LPR) cameras recognize and record the license plate of vehicles that pass in their field of view. Both of these types of AVI sensors record a unique identifier for the vehicle, the location of the vehicle when the capture occurred, and the time of the capture.

Using *only* AVI data, and assuming a perfect detection rate at all nodes in the network, hypothetically one would be able to exactly reconstruct the OD matrix since observations of the origin and destination of every vehicle passing through the sensor network would be available. A deployment level of 100% is impractical due to the cost of deploying and maintaining that level of infrastructure. Previous work has addressed the imperfect collection rates of AVI sensors or incomplete network coverage.

Using AVI data for OD matrix estimation has been considered by [9] and [10]. Both of these approaches treat AVI data in a similar way, using it in conjunction with traffic counts to improve the performance and reliability of OD matrix estimates. [9] combines a traditional model for traffic counts with a separate model for AVI based OD pair estimates and tries to compute the solution of both as "trajectory count contributions" where the AVI based input is of a different form than the traffic counts based method. The model they propose involves a sort of correlation expression between data from AVI sensors and the traffic count sensors deployed on a network. It has the benefit of not modelling the OD pairs as the locations of the AVI sensors and allowing arbitrary origin and destination points/zones to be estimated. [10] use the AVI data to compute the OD estimate on an incorrect scale and the estimated OD pairs only occur at the locations of the AVI sensors. Then [10] use traffic count data to estimate a sampling rate variable that can be used to boost the AVI-only estimated OD matrix to the correct scale.

[11] use AVI data in conjunction with link counts to estimate the trip matrix and path flow reconstruction. They suggest that the use of AVI data, with sufficient coverage, can lead to perfect or near-perfect estimates of the OD matrix. They propose a method that requires the enumeration of all feasible paths through the network, and they use plate scanning in conjunction with a prior OD matrix and link counts to estimate the paths that are being followed by vehicles through the network. From the estimated paths, they are able to reconstruct a trip-table (OD matrix) estimate. They also propose a method for determining where to place AVI sensors. However, the method requires enumerating all possible paths in the network and scales super-linearly with the size of the network. This is computationally intractable for most urban settings of interest. We propose an alternate approach where a subset of the most relevant feasible paths are learned from historical data, without prior path information except for the number of sensors and their identifiers.

Similar to [11], [12] present a method for using a small sample of traffic trajectories from Bluetooth sensors in addition to complete coverage with traffic count sensors on every road segment to estimate a *link-dependent OD matrix* (LODM). The LODM model extends the traditional OD matrix estimation problem in that the flow distribution on links is estimated in addition to the OD counts. Consequently, the problem of link assignment in OD matrix estimation is integrated into the model to reduce computational complexity.

Using properties of a transportation system, they formulate LODM estimation as a convex optimization problem and then propose a primal-dual algorithm to solve it. This method fits most aptly to a system where traffic count sensors are readily available and newer AVI technology is being deployed to complement the existing sensor network. However, by their own argument, their method is only applicable to large artery road segments which are typically the only road links equipped with traffic count sensors. Rural or neighbourhood roads are not able to be modelled unless they are also equipped with traffic count sensors.

[13] present a particle filtering method using only AVI data which reconstructs complete vehicle trajectories using only partial trajectory observations. Once the complete trajectory distribution is estimated, calculating the OD matrix is trivial by simply looking at the starting and terminating locations of the trajectories. They also demonstrate how the coverage level of AVI sensors in the network affects OD matrix reconstruction performance. However knowledge of all possible routes between OD pairs is necessary in order to estimate the contribution of partially observed trajectories.

### 8.1.3 Contribution

We propose a Bayesian approach to estimate OD matrices from AVI data based on a Markov model. The proposed approach does not require an initial OD matrix. We propose an inference method for this model which is deterministic (i.e., it does not use random sampling techniques such as Markov chain Monte Carlo or Gibbs sampling), and consequently it is practical for applications requiring real-time OD matrix estimation. The proposed method also processes data in an online fashion, as it becomes available. The method does not require the specification or computation of possible routes; rather, this information is learned from the data. Finally, the proposed method has the ability of separating route flows and OD pairs via a unique mixture model representation. This allows for some routes and OD pairs to be separated from others, helping the user to identify which routes contribute to which OD pairs.

### 8.1.4 Paper Organization

The rest of the paper is organized as follows. Section 8.2 gives a detailed description of the problem setup and assumptions. Section 8.3 outlines the generative time-varying mixture model applied to AVI data. Section 8.4 describes how the generative model can be used to estimate a mixture model of OD matrices as well as compute a traditional OD matrix estimate for the observed traffic. Section 8.5 investigates the proposed model and estimation method performance on multiple datasets and compares it to traditional results, and we conclude in Section 8.6.

## 8.2 Problem Description

This section describes the problem of estimating an OD matrix using AVI data. We describe characteristics of the measurements that make the problem challenging. We further discuss assumptions made under the proposed statistical model. Finally, we describe performance metrics which will be used to evaluate the proposed model.

### 8.2.1 AVI Data

Consider a network of urban roads instrumented with AVI sensors. When a vehicle passes within the sensor's detection range, the sensor records the unique identifier of the vehicle, the time of the vehicle observation, the location (latitude and longitude) of the observation, and the ID of the sensor that collected the observation. AVI sensors typically have a short range of detection (e.g., 10 meters). Each collected vehicle observation is then reported to a fusion center whose goal is to estimate the OD matrix as it changes over time.

Formally, we consider a collection of $C$ sensors which form the state space $\Omega = \{1, ..., C\}$. The sequence of observations, $x^i$, of a vehicle $i$ through this network are $x^i = (x^i_0, ..., x^i_{n_i})$ where $x^i_j \in \Omega$ is the $j^{th}$ state where vehicle $i$ is observed, and $n_i$ is the number of observations of vehicle $i$. In this paper we focus on the observation sequence of states; the observations times are only used to determine the order of states that observe a vehicle, and otherwise they are ignored.

### 8.2.2 Measurement System Characteristics and Assumptions

We make the following assumptions about the AVI sensors and the deployed network structure. First the sensors are time-synchronized so that the timestamps between sensors are comparable. This is necessary to time-order observations of a vehicle between different sensors without errors. It is justifiable since the AVI sensors are typically equipped with a GPS receiver that provide reliable and accurate synchronization.

Second, we assume that no vehicle can be observed by different sensors at exactly the same time. This is necessary so that all the observations can be uniquely ordered. This is justified since sensors have a sufficiently high time resolution and non-overlapping fields of view. A realistic deployment may involve sensors with overlapping regions of detection, so that two sensors may detect a vehicle at the same time. However, one could take the sensor locations into account to determine which of the detections was earlier. We leave this extension for future work.

Third, we assume that no matter what unique identifier is used for a vehicle (license plate, Bluetooth MAC address, EZpass ID) there is no error when the identifier is recorded. This is reasonable for LPR sensors, which have a very low error rate, but less reasonable for other AVI sensors (e.g., Bluetooth-based). Extending the model to account for errors in the measurements is left for future work.

Fourth, we assume that the sensors are static and that their positions are known to the fusion center. This means that the structure of the sensor network does not change over time.

Finally, we assume that the sensors send their observations to the fusion center over a sufficiently delay-minimal channel that errors in the orderings of the observation sequences is impossible due to transmission delay. This is reasonable because the amount of data being transmitted by each AVI sensor (e.g., 64-bit identifier, 64-bit latitude, 64-bit longitude, 64-bit camera identifier, 64-bit timestamp) are sufficiently small that subsequent observations of a vehicle are further apart than the time to transmit the observation to the fusion center.

### 8.2.3 Sequential Modelling of OD Matrices

In this work we consider data to be organized into sequential time-windows $t \in \{1, 2, ...\}$. This means that the vehicle observation sequences are collected into a time-window and a model is computed for that specific window. The model for the observations in time-window $t$ is stationary within the time-window, but may vary from one time-window to the next. The number of vehicles observed in a time-window $t$ is denoted $N(t)$. Observations in time-window $t$ are further organized into sequences of time-ordered observations for each vehicle $X(t) = \{x^1, ..., x^{N(t)}\}$. In each time-window, the model described in Section 8.3 is used to describe the data that arrived in that time-window. The model from the previous time-window is used as a prior for the current time-window's model. This enforces dependence between time-windows and the notion of the model "evolving" from one time-window to the next.

Using the model in Section 8.3 we can compute component-wise and marginal OD matrices. The marginal OD matrix is typically what is estimated utilizing traditional methods with traffic count data. We can compare the root mean-squared error (RMSE) of our estimated OD matrix against the RMSE which results from other traditional methods [14]. The RMSE is defined as

$$\mathrm{RMSE}(\widehat{OD}, OD) = \sqrt{\frac{\sum_{i=1}^{Z}\sum_{j=1}^{Z}\left(\widehat{OD}_{ij} - OD_{ij}\right)^2}{Z^2}} \tag{8.1}$$

where $\widehat{OD}$ is the estimated OD matrix, $OD$ is the true OD matrix, and $Z$ is the number of zones in the network. This is demonstrated on a simulated dataset based on a random network of sensors and another dataset based on a real road-network in Section 8.5.

## 8.3 The Model

Given a collection of AVI observations, if one groups the observations by the vehicle observed and then orders the observations by the time received, then the observations for a single vehicle constitute an irregularly-sampled time-series through the network of AVI sensors. This is the same style of approach as taken in [15]. This type of data can be modelled as a random walk through the network. However a traditional Markov random walk would only model the initial state a vehicle was observed at and the route they follow, and it would not explicitly model the last state where the vehicle is observed (the destination). Therefore, we propose a random walk where, at each step in the walk, the walker flips a biased coin to determine whether they stop at the current state or continue. We believe this type of model can accurately model the origin and destination states, as well as the corresponding routes vehicles take through the network. In order to more accurately estimate non-dominant routes and OD pairs, we further propose using a mixture of these types of random walk models.

The model presented in the following section is an extension of an earlier model proposed for estimating road traffic [15]. This previous work used a similar sequential approach to incorporating AVI observations, but it only modelled the states where a vehicle is first observed, and subsequent transitions. Since the previous approach does not model the destination, it is not directly applicable for OD matrix estimation. The model technique described below expands on this previous work to include three different terms: 1) the vehicle origin; 2) the path of the vehicle through the network of sensors; and 3) the destination of the vehicle.

After describing the model, we describe how the *Automatic Hard Expectation-Maximimization (EM) algorithm*, introduced in [15], can be modified to perform inference of this new model. In particular, with appropriate modification of the algorithm, we infer a mixture model, where each mixture component describes a subset of the observations, yielding finer resolution OD/route estimation than is captured using traditional OD matrix estimation approaches.

### 8.3.1 First-Order Markov Model for Vehicle Trajectories

Formally, we begin with an observation sequence of vehicle $i$, $x^i = (x^i_1, \dots, x^i_{n_i})$, as previously defined. Under a first-order Markov model, it is equivalent to have three sufficient statistics, $x^i = \{x^i_0, x^i_{n_i}, X^i\}$, where $x^i_0, x^i_{n_i} \in \Omega$ are the first and last sensors where the vehicle was observed, and $X^i \in \mathbb{Z}^{|\Omega|\times|\Omega|}_{\geq 0}$ is the state transition count matrix, with the entry $X^i_{j,k}$ denoting the number of times vehicle $i$ was observed at sensor $k$ immediately after it was observed at sensor $j$.[2]

We begin by examining the model for a single-vehicle random walk with a termination term. Consider a discrete-time Markov chain with initial-state distribution $\pi \in \mathbb{R}^{|\Omega|}$ and transition matrix $P \in \mathbb{R}^{|\Omega|\times|\Omega|}_{\geq 0}$. These parameters are constrained to satisfy

$$\sum_{j=1}^{|\Omega|} \pi_j = 1 \tag{8.2}$$

$$\sum_{k=1}^{|\Omega|} P_{j,k} = 1, \quad \forall j \in 1, .., |\Omega|. \tag{8.3}$$

We further propose a matrix $B \in (0,1)^{|\Omega|\times|\Omega|}$ where each element $B_{j,k}$ represents the probability that vehicle $i$ terminates its path at $k$ given it started at initial state $j$. The likelihood

[2] Here $\mathbb{Z}_{\geq 0}$ denotes the non-negative integers $\{0, 1, ...\}$.

of this random walk with termination of a observation sequence $x^i$ is

$$p(x^i|\pi, P, B) = \pi_{x_0^i} \left[ \prod_{j=1}^{|\Omega|} \prod_{k=1}^{|\Omega|} (P_{j,k})^{X_{j,k}^i} \right] \left[ B_{x_0^i, x_{n_i}^i} \prod_{j=1}^{|\Omega|} \Big( 1 - B_{x_0^i, j} \Big)^{\sum_{k=1}^{|\Omega|} X_{j,k}^i} \right]. \tag{8.4}$$

The variable $B$ can be thought of as the parameters of $|\Omega|^2$ Bernoulli random variables, one for each pair of starting and terminating states. At each step of the observation sequence, we consider vehicle $i$ to be at state $k$ and an independent Bernoulli is drawn with parameter $B_{x_0^i, k}$ to determine whether or not vehicle $i$ terminates at state $k$ (i.e. state $k$ is vehicle $i$'s destination). The last term in this expression is similar to a Geometric distribution stating that of $n_i$ trials, there was only one success; however, here each trial has a different probability. This proposed likelihood is governed by three parameters $(\pi, P, B)$ which for brevity we denote by $\phi = \{\pi, P, B\}$. The traffic model proposed in [15] does not model a terminating state and only estimates the initial state distribution $\pi$ and transition matrix $P$, modelling only how vehicles move through the network.

Now we extend this to the time-varying mixture model of $M(t)$ mixands, similar to [15], which becomes

$$p(x^i|\, \mathbf{\Theta}(t)) = \sum_{m=1}^{M(t)} \alpha^{(m)}(t) p(x^i|\phi^{(m)}(t)), \tag{8.5}$$

where $\mathbf{\Theta}(t) = \{ \big( \alpha^{(1)}(t), \phi^{(1)}(t) \big), ..., \big( \alpha^{(M(t))}(t), \phi^{(M(t))}(t) \big) \}$, and $\alpha(t) = (\alpha^{(1)}(t), \ldots, \alpha^{(M(t))})$ is the vector of $M(t)$ mixture weights, one for each of the components in the mixture model in time-window $t$, satisfying

$$\sum_{i=1}^{M(t)} \alpha^{(i)} = 1.$$

In the following sections we describe the model for the component and parameter dynamics. It is a modified version of the model described in [15] which governs the birth of new components, death of existing components, and evolution of persisting components as time continues. The dynamics also relate a model via a Markovian dependence from the previous time-window.

### 8.3.2 Mixture Component Death

Consider the mixture model in time-window $t-1$ which contains $M(t-1)$ components. This mixture model is parametrized by the collection of parameters $\boldsymbol{\Theta}(t-1) = \{\boldsymbol{\alpha}(t-1), \boldsymbol{\phi}(t-1)\}$ where $\boldsymbol{\alpha}(t-1)$ is the vector of mixture component weights of length $M(t-1)$ and $\boldsymbol{\phi}(t-1)$ is the vector of parameters of each component of length $M(t-1)$.

Components can either persist or die from time-window $t-1$ to $t$. They die with probability

$$p_d(\phi^{(m)}(t), \alpha^{(m)}(t)).$$

When a component dies, the mixture weight of that component is distributed proportionally among the remaining components as in [15].

### 8.3.3 Persisting Component Evolution

Components which survive from time-window $t-1$ to $t$ have their parameters evolve according to [15]

$$\phi^{(m)}(t) \sim H(\phi^{(m)}(t-1)) \tag{8.6}$$

where $H$ is a distribution that is parametrized by the previous time-window's component parameters. It has a probability density function $h(\phi^{(m)}(t)|\phi^{(m)}(t-1))$. In this adjusted model for traffic, this evolution is specifically the following. The starting state and transition matrices evolve according to [15]

$$P^{(m)}(t) \sim \begin{bmatrix} \mathrm{Dir}(P^{(m)}_{(1,:)}(t-1)) \\ \mathrm{Dir}(P^{(m)}_{(2,:)}(t-1)) \\ \vdots \\ \mathrm{Dir}(P^{(m)}_{(|\Omega|,:)}(t-1)) \end{bmatrix} := \mathbf{Dir}\left(P^{(m)}(t-1)\right) \tag{8.7}$$

$$\pi^{(m)}(t) \sim \mathrm{Dir}(\pi^{(m)}(t-1)) \tag{8.8}$$

where $\mathrm{Dir}(\cdot)$ denotes the Dirichlet distribution. The terminating distribution similarly evolves according to

$$B^{(m)}_{j,k}(t) \sim \mathrm{Beta}(B^{(m)}_{j,k}(t-1), 1 - B^{(m)}_{j,k}(t-1)) \tag{8.9}$$

which we write as

$$B^{(m)}(t) \sim \mathbf{Beta}(B^{(m)}(t-1)) \tag{8.10}$$

to represent the evolution of all of the values in the matrix $B$. This term expresses the probability that vehicle $i$ terminates at state $j$ given is started at state $x_0^i$. Given the component which a vehicle is distributed according to $(m)$.

### 8.3.4 Mixture Component Birth

The number of mixture components which are born in the $t$th time-window, $n_b(t)$, is distributed according to a Poisson random variable with parameter $\lambda(t)$, $n_b(t) \sim \text{Pois}(\lambda(t))$. Each new component born in time-window $t$ is distributed according to a set of base-distributions with parameter $\mathbb{G} = \{G_p, G_\pi, G_B\}$ so $\phi^{(m)}(t) \sim H(\mathbb{G})$. In this model for traffic routing with termination, the disribution $H()$ is defined via

$$P^{(m)}(t) \sim \mathbf{Dir}(G_p) \tag{8.11}$$

$$\pi^{(m)}(t) \sim \text{Dir}(G_\pi) \tag{8.12}$$

$$b^{(m)}(t) \sim \mathbf{Beta}(G_B). \tag{8.13}$$

Each new component also has an initial mixture weight distributed according to the model in [15] which is

$$\alpha^{(M(t)+1)} \sim \text{Beta}(1, M(t)). \tag{8.14}$$

The other remaining weights are then scaled down by $(1 - \alpha^{(M(t)+1)})$ as in [16].

### 8.3.5 Mixture Weight Dynamics

All the dynamics except for the mixture weight evolution from $t-1$ to $t$ are now defined. In order to handle this evolution, we note that the dimension of this vector will change between time-windows as components are born or die. Therefore we assume that after the births and deaths have occurred we have a *rescaled* weight vector for time-window $t-1$ to $t$, denoted $\boldsymbol{\alpha}'(t)$, which governs the dynamics of the weights vector for time $t$ via

$$\alpha(t) \sim \text{Dir}(\alpha'(t)). \tag{8.15}$$

## 8.4 OD Matrix Estimation Method

Given a collection of AVI data from a network with state space $\Omega$, a fitting procedure based on the Automatic Hard EM algorithm [15] can be applied to the model described in Section 8.3. As mentioned above, the algorithm of [15] does not account for the destination, and hence it must be modified to perform inference on the parameters $B^{(m)}$, in addition to the initial state distributions $\pi^{(m)}$ and transition matrices $P^{(m)}$.

### 8.4.1 The Automatic Hard EM Algorithm

The Automatic Hard EM Algorithm performs a similar clustering function to that of the traditional Hard EM algorithm [17] where it clusters datapoints based on the maximum likelihood estimate of which cluster the datapoint was generated by. However the Automatic Hard EM Algorithm provides a series of modifications to traditional Hard EM in order to automatically select the correct number of mixture components as well as perform estimates on time-varying data.

It begins, in each time-window $t$, by first initializing the number and parameters of the mixands being estimated to the previous iteration's estimated mixands plus one new component with parameters equal to the global prior distribution $\mathbb{G}$. It then performs a traditional Hard EM estimation of this new mixture model until the EM assignments stop changing (i.e. the Hard EM algorithm has converged). The *expectation* (E) and *maximization* (M) steps for this phase are

1. E-Step: $S_m = \left\{ x^j : p(x^j|\hat{\phi}^{(m)}) > p(x^j|\hat{\phi}^{(m')}) \forall m' \neq m \right\}$
2. M-Step: $\hat{\phi}^{(m)} = \underset{\hat{\phi}^{(m)}}{\arg\max} \, f(S_m|\hat{\phi}^{(m)}) h(\hat{\phi}^{(m)}|\hat{\phi}_0^{(m)})$

where $S_m$ is the assignment matrix, assigning datapoints to mixands, and $\hat{\phi}_0^{(m)}$ are the *initial* values of the parameters for mixand $m$ for time-window $t$. If any data was assigned to the *last* mixand, this indicates that a new component has been estimated (since this mixand has the parameters of the global prior distribution). In this case, the Automatic Hard EM Algorithm will add an additional mixand with parameters equal to $\mathbb{G}$, increasing the mixture model order by 1. The Hard EM and cluster creation portions of the algorithm are repeated until no datapoints are assigned to the last mixand meaning that a *maximum* mixture model order has been estimated [15].

The main difference between the Automatic Hard EM algorithm of [15] and the version required for the model proposed here lies in the M-step. For the proposed model, updated estimates for the model parameters $\phi^{(m)} = (\pi^{(m)}, P^{(m)}, B^{(m)})$ are calculated via

- $\hat{\pi}^{(m)}(j) = \hat{\pi}_0^{(m)}(j) + \sum_{k=1}^{N(t)} \mathbb{1}\{[x_0^k = j]\}\mathbb{1}\{[S_m^k = m]\}, \forall j \in \Omega \wedge \forall m \in 1..M(t)$
- $\hat{P}^{(m)} = \hat{P}_0^{(m)} + \sum_{k=1}^{N(t)} X^k \mathbb{1}\{[S_m^k = m]\}, \forall m \in 1..M(t)$
- $\hat{B}^{(m)}(i,j) = \hat{B}_0^{(m)}(i,j) + \sum_{k=1}^{N(t)} \mathbb{1}\{S_m^k = m\}\mathbb{1}\{x_0^k = i\}\mathbb{1}\{x_{n_i}^k = j\}, \forall i,j \in \Omega \wedge \forall m \in 1..M(t)$

and then normalizing the expressions to satisfy the constraints

- $\sum_j \hat{\pi}_j^{(m)} = 1$
- $\sum_j \hat{P}_{i,j}^{(m)} = 1, \forall i \in \Omega$
- $\sum_j \hat{B}_{i,j}^{(m)} = 1, \forall i \in \Omega.$

The Automatic Hard EM Algorithm has a final two steps in order to anneal the model order, which is likely to have been over-estimated by the previous phase since it only involves the addition of new components. It first removes any components which have a mixture weight $\alpha^{(m)}$ below a threshold $L_\alpha$ since they describe no, or possibly very few datapoints, and therefore do not warrant the added model complexity. Additionally the algorithm merges mixands by a weighted summation of their parameters when the KL divergence between mixands falls below a threshold $L_{KL}$. This means that if two mixands were estimated for very similar data (the traditional overfitting problem), then only one mixand is needed to describe the portion of data [15].

### 8.4.2 Estimating an OD Matrix from the Output of the Automatic Hard EM Algorithm

The algorithm will output a single mixture model with $M(t)$ components, and we can compute an estimated OD matrix for each component by

$$\widehat{OD}_{j,k}^{(m)}(t) = N(t)\pi_j^{(m)}(t)\alpha^{(m)}(t)B_{j,k}^{(m)}(t), \forall j,k \in 1,...,|\Omega|, m \in 1,...,M(t). \tag{8.16}$$

This expression gives the OD matrix entries associated with individual components. Each component also includes the routing information, encoded in the transition matrix $P^{(m)}$, and so each component naturally only describes a subset of all traffic observed over the network. To compute the marginal OD matrix, for all traffic over the network, we take the sum of the component-wise OD matrices (since they are already scaled by $\alpha^{(m)}$),

$$\widehat{OD} = \sum_{m=1}^{M} \widehat{OD}^{(m)}. \tag{8.17}$$

The estimated marginal OD matrix will be used when making comparisons with traditional OD matrix estimation methods.

### 8.4.3 Bootstrapping the Automatic Hard EM Algorithm

The Automatic Hard EM Algorithm [15] is designed to work with time-varying data that is organized into time-windows. However the algorithm can also be applied to a single time-window of data with an unknown number of components in the mixture model via bootstrapping the algorithm. The algorithm is Bayesian and outputs the *maximum a posteriori* estimate at each time-window, using the previous time-window as a prior. The first time the algorithm is executed, there is no previous time-window to use. As an alternative, one could use a very uninformative prior distribution (e.g., setting each entry of the initial state distribution and and transition matrix to $1/|\Omega|$), however such a prior is likely to lead to poor-quality estimates since it uses no knowledge of the road network structure.

An alternative approach is to form a prior by fitting a 1-component MAP estimate to a preliminary set of data, using the uninformative prior for this estimate. Then the 1-component MAP can be used as a prior when fitting the mixture model. In the generative model this restricts the space at which new components can be born to a smaller, lower-variance region. It is a way of providing domain knowledge of the problem in advance to make the convergence speed of the algorithm greater. This will make the prior distribution have just a high enough variance to cover all of the observed data. The 1-component MAP estimate in a discrete-time Markov chain model is simply the proportion of times a transition was observed and therefore computed with minimal computational overhead.

It is appropriate to utilize the 1-component MAP estimate when only dealing with a single time-window of data since the need to be flexible for new, unobserved components to occur is not present. In the time-varying case, as time progresses, components that have not ever been observed before can appear. Therefore the prior distribution needs to have as high a variance as possible to have non-zero support on the regions where these unobserved components may appear. Utilizing a single component MAP estimate is therefore not appropriate in the time-varying case but is acceptable in the single time-window case.

## 8.5 Numerical Experiments

In the following section we asses the proposed model and estimation algorithm's performance on a series of simulated datasets. We begin with a simple road network, commonly used in work on OD estimation, to demonstrate that the proposed inference method and model perform as expected, as well as to investigate the model's ability to separate OD pairs into a mixture. We then extend this to simulated data based on the road network of Montréal, Canada.

### 8.5.1 The Yang Network

In order to evaluate the proposed model and algorithm, we use a sample network from [14]. The network in shown in Fig. 8.1 and consists of nine nodes. There are two possible origins, nodes 1 and 2, and two destinations, nodes 3 and 4. Each of the links shown in Fig. 8.1 has a specified capacity $C_a$ and free-flow travel-time $c_a(0)$, where $a$ is the link index. The values for each link are reproduced in Table 8.1. The link capacity determines the number of vehicles the link can hold before having congestion and the free-flow travel-time denotes the time it takes for a vehicle to traverse the link (in time-units).

In order to gather samples from this network to evaluate our proposed method, we need to assign traffic to the routes present on the network. In order to do this, we use the method described in [18]. This method uses the Bureau of Public Roads (BPR) function,

$$S_a(x) = c_a(0) \times \left(1 + 0.15 \left(\frac{x}{C_a}\right)^4\right) \tag{8.18}$$

where $S_a(x)$ is the average travel time of a vehicle on link $a$. The BPR function relates the link capacity to the free-flow travel time. This method solves the Wardrop equilibrium conditions [19] using the Frank-Wolfe algorithm [18]. The problem can be formulated as a non-linear programming problem,

$$f(\mathbf{v}) = \min \sum_a \int_0^{v_a} S_a(x) dx \tag{8.19}$$

subject to

$$v_a = \sum_i \sum_j \sum_r \alpha_{ij}^{ar} x_{ij}^r \tag{8.20}$$

$$T_{ij} = \sum_r x_{ij}^r,\ v_a \geq 0,\ x_{ij}^r \geq 0 \tag{8.21}$$

where $v_a$ is the volume of traffic on link $a$, $\alpha_{ij}^{ar} = 1$ if link $a$ is on path $r$ from origin $i$ to destination $j$ and zero otherwise, and $x_{ij}^r$ is the number of vehicles on path $r$ from origin $i$ to destination $j$.

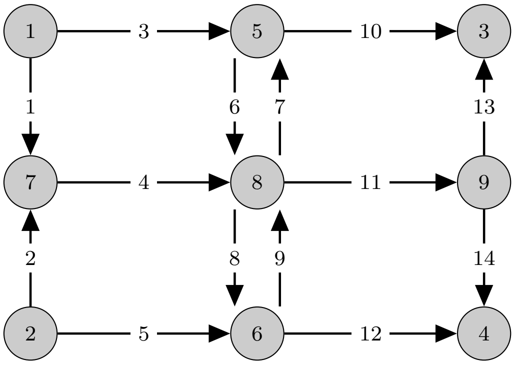


**Fig. 8.1** Example network used in the transportation research field, proposed in [15].

### Extending the Yang network

Due to the small size of this network, any reasonable deployment of AVI sensors will observe all of the vehicles leaving a source node and ending at a terminating node. Therefore we treat the nine nodes of the Yang network as *zones* in the extended network shown in Fig. 8.2. Here each zone contains four nodes and the four nodes are inter-connected as a ring. All intra-zone links have capacity $C_a = 300$ and free-flow travel-time $c_a(0) = 4$.

| Link Number | Link Start Node | Link End Node | $C_a$ | $c_a(0)$ |
|---|---|---|---|---|
| 1 | 1 | 7 | 150 | 4 |
| 2 | 2 | 7 | 150 | 4 |
| 3 | 1 | 5 | 250 | 12 |
| 4 | 7 | 8 | 250 | 10 |
| 5 | 2 | 6 | 250 | 11 |
| 6 | 5 | 8 | 150 | 3 |
| 7 | 8 | 5 | 150 | 4 |
| 8 | 8 | 6 | 150 | 4 |
| 9 | 6 | 8 | 150 | 5 |
| 10 | 5 | 3 | 250 | 12 |
| 11 | 8 | 9 | 250 | 12 |
| 12 | 6 | 4 | 250 | 11 |
| 13 | 9 | 3 | 150 | 4 |
| 14 | 9 | 4 | 150 | 3 |

**Table 8.1** The capacities and free-flow travel times for each link in the network shown in Fig. 8.1 as introduced in [15].

Consequently, traffic flows smoothly within each zone and then follows the model of Yang for inter-zonal travel. In the extended model we also put two links between each zone, so that we can investigate the effect of observing a subset of the traffic flowing between zones. The capacity of these intra-zone links is set to one-half the capacity defined in the [14] model, so that the total capacity between zones remains the same as in the original model. For example, in Fig. 8.1, the link between zones 1 and 5 has capacity $C_a = 250$, and in Fig. 8.2 the links $1b \to 5a$ and $1d \to 5c$ both have a capacity of $C_a = 125$.

### Traffic Assignment

[14] proposes to use the true OD matrix, shown in Table 8.2, as a prior. In the extended network of Fig. 8.2, the origins and destinations are zones, not specific nodes. Therefore we try two alternate mappings between the origin/destination zones and nodes. The first mapping sets the origin and destination to be one specific node in the zone. Specifically, all traffic going to/from zone 1 starts or ends at node $1a$, and the respective nodes in zones 2–4 are $2c$, $3b$, and $4d$. We call this *Scenario 1.* Alternately, the second mapping distributes the load associated with a given OD pair uniformly over the four nodes of that zone, and we refer to this as *Scenario 2.* For either scenario, after mapping the zone-level OD values to

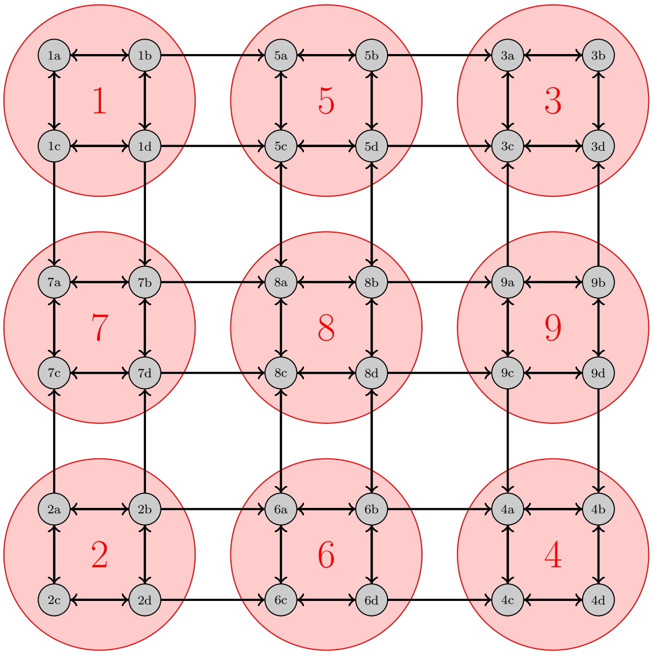


**Fig. 8.2** Example network with zones used in this work. Extended from Fig. 8.1

node-level values, we solve the equilibrium problem and assign traffic to follow the resulting routes.

| Origin | Destination | Vehicle Count |
|---|---|---|
| 1 | 3 | 200 |
| 1 | 4 | 150 |
| 2 | 3 | 140 |
| 2 | 4 | 185 |

**Table 8.2** True OD pairs for example network in Fig 8.1. All other OD pairs are 0.

### Fitting the Model

Once traffic is assigned to routes on the network according to the user equilibrium, we simulate the observations of AVI sensors deployed in the network shown in Fig. 8.2. We do this by selecting a subset of the nodes and record when vehicles pass by one of the monitored nodes. For example, in Fig. 8.2, if a vehicle $i$ moves from $1a$ to $1b$ to $5a$ and there are sensors at nodes $1a$ and $5a$, the observation sequence for the vehicle is $x^i = (1a, 5a)$ where there is a missing observation of when vehicle $i$ passed through $1b$.

We note that the state space used in the model (8.4) is not the total number of nodes in the network. Rather, there is one state (an element of $\Omega$) for each deployed sensor. Interpolating traffic patterns from this subset of observations points to all edges of the road-network is an interesting problem we leave for future work.

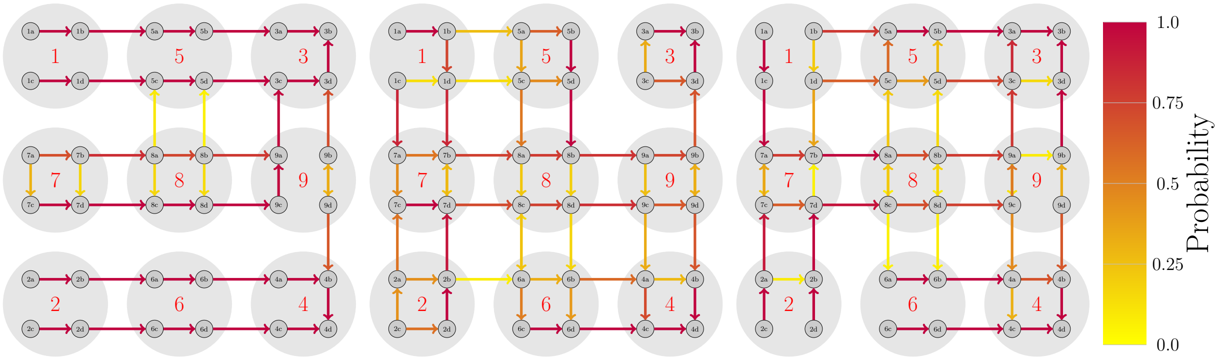
We can now fit the Markov chain-based time-varying mixture model (8.4) for the single time-window of data. For the fitting procedure we set $L_\alpha = \frac{2}{N(t)}$, where $N(t)$ is the number of vehicles observed in time-window $t$, and $L_{KL} = 0.05$. This means that $L_\alpha$'s value will change with each time-window but components created with only 1 or 2 observation sequences assigned to them will be deleted regardless of the total number of observations. Also if the KL divergence between two components gets very small, below 0.05, we merge them.

### 8.5.2 Resluts

#### Monitoring All Nodes

We begin by monitoring all of the nodes in the network. This is equivalent to having a sensor at every intersection in the road network. We estimate the model described in Section 8.3 and the result is a mixture model with three components. We visualize the resulting transition matrices by coloring the edges of the network as shown in Fig. 8.3. There are visibly distinct differences in the routing between nodes for different OD pairs. The first component, Fig. 8.3(a), represents the estimated OD pairs 1-3 with a load of 200 vehicles and 2-4 with a load of 185 vehicles. Further Figs. 8.3(b) and 8.3(c) represent the OD pairs 1-4 and 2-3 at estimated vehicle counts for each pair of (116 for pair 1-4, 51 for pair 2-3) and (33 for pair 1-4, 88 for pair 2-3) respectively. These latter two figures represent the separation of the many alternate routes for an OD pair. The unique aspect to note here is that the model and estimation technique naturally separates OD pairs based on the routes that are involved in those pairs. For the marginal estimated OD matrix computed using (8.17), the resulting RMSE is 0.3338 vehicles.

We also examine the relative error in the distribution of the path-lengths of vehicles' observed paths. If $N$ vehicles are travelling from zone 1 to node 2, we compute the distance between the centers of nodes 1 and 2 and state $N$ vehicles are travelling that distance.



(a) 1-3 (200 v) and 2-4 (185 v) (b) 1-4 (116 v) and 2-3 (51 v) (c) 1-4 (33 v) and 2-3 (88 v)

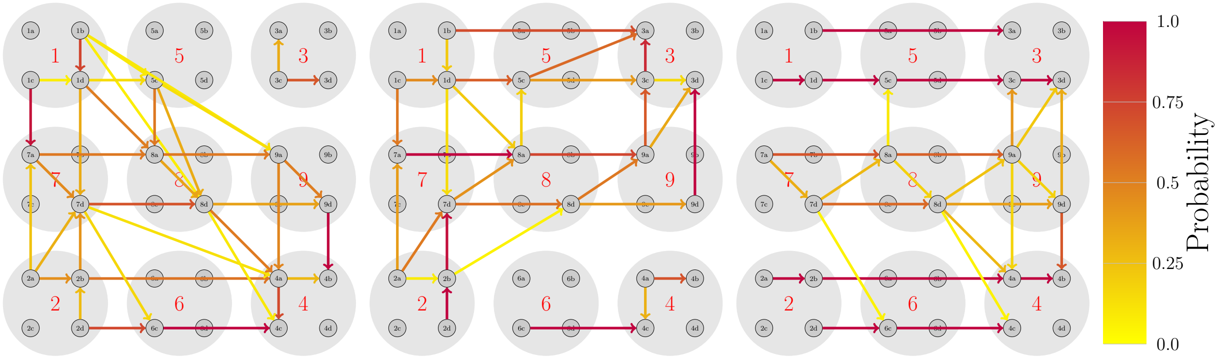
**Fig. 8.3** "Hot" route plots when monitoring *all* of the nodes in the network. These plots show the routing matrices for each of the three estimated components for the following scenarios. The color of each link corresponds to a probability, via the colorbar on the right. Links with estimated probabilities below 0.05 are not shown removed for clarity. (a) Estimated routing matrix for estimated OD pairs 1-3 (200 vehicles) and 2-4 (185 vehicles). (b) Estimated routing matrix for estimated OD pairs 1-4 (116 vehicles) and 2-3 (51 vehicles). (c) Alternate estimated routing matrix for estimated OD pairs 1-4 (33 vehicles) and 2-3 (88 vehicles).

We perform this mapping for all the estimated OD pairs and true OD pairs, giving us collections of distances covered. Then we categorize these vehicle path lengths into bins of $0-1$, $1-2$, ..., etc km traveled. The vector of bin counts in time-window we call $h(t)$. Finally, we calculate a histogram density estimate for the true and estimated vehicle path length distributions, and we compute the following relative $\ell_2$ norm error

$$\frac{\|\hat{h}(t)-h(t)\|_2^2}{\|h(t)\|_2^2} \tag{8.22}$$

where $\hat{h}(t)$ is the estimated vector of bin-counts in time-window $t$. This gives a relative $\ell_2$ norm error for a density estimate on the estimated trip-length distribution of the estimated OD matrix ($\hat{OD}$) versus the true OD matrix ($OD$).

In order to asses relative error on this small, simulated network, we need to assign locations to each node. We do so on a grid structure so node 1a is at location (1,1), 1b is at (1,2), and so on, up to 9d at (9,9). For this first scenario, where all nodes are observed, we get a relative $\ell_2$ error in the path-length distribution of only $2.39 \times 10^{-7}$ which is very small and again is indicative that observing all of the nodes in the network results provides a very high-accuracy estimate of the OD matrix.



(a) 1-4 (149 v) (b) 2-3 (139 v) (c) 1-3 (200 v) and 2-4 (18 v)

**Fig. 8.4** Routes estimated using the subset of monitored nodes discussed in Section 8.5.2. Here we see again that three components are estimated in the mixture model with the demonstrated transition matrices. Links with probability less than 0.05 are not shown for clarity. The link probability weight is denoted in the colorbar on the right. (a) Estimated routing matrix for estimated OD pair 1-4 (149 vehicles). (b) Estimated routing matrix for estimated OD pair 2-3 (139 vehicles). (c) Estimated routing matrix for estimated OD pairs 1-3 (200 vehicles) and 2-4 (18 vehicles).

### Monitoring a Subset of Nodes

Next we consider the case where sensors are only located at 21 of the nodes (1b, 1c, 1d, 2a, 2b, 2d, 3a, 3c, 3d, 4a, 4b, 4c, 5c, 6c, 7a, 7d, 8a, 8d, 9d, 9a). Then the state space $\Omega$ is half of the original state space reported above. We run the fitting routine and get a resulting mixture model again with three mixture components. The resulting mixture components are shown in Fig. 8.4. The RMSE on the estimated marginal OD matrix using only these 18 nodes is 0.3334. Also the relative $\ell_2$ error in the path-length distribution is 0.5665 which has grown from the previous scenario due to us not monitoring as many nodes. We can see in these plots that the estimated transitions are not necessarily the actual links of the network. This is because without knowing all the nodes of the network, the model estimates transitions between nodes it can observe. Therefore it can appear as though there are nodes which do not technically exist. This is not the problem in Fig. 8.3 because we monitor all nodes. In that case we can see the entire network structure.

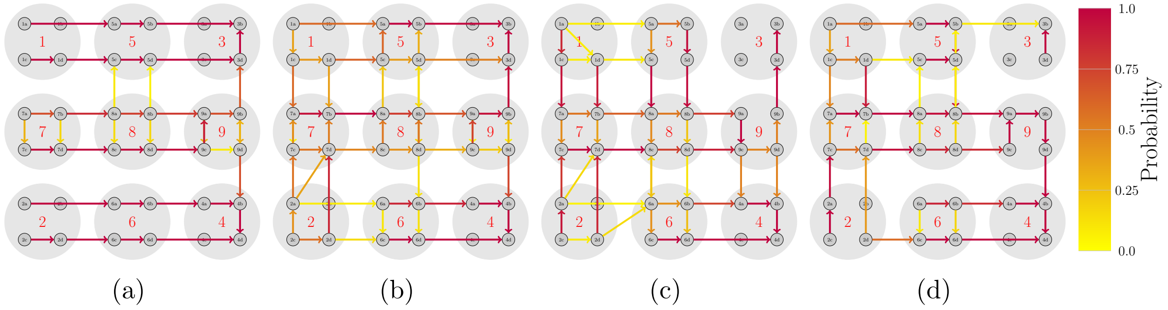


**Fig. 8.5** Routes estimated using a even smaller subset of nodes than the previous example. Specifically the set of nodes 1a,1c,1d,2a,2c,2d,3b and 3d-9d are observed. The four components estimated are demonstrating similar trends as before however we are now seeing duplicate route estimate matrices depending on which node inside a zone is estimated. The relevant OD pairs are excluded from the four figures for simplicity.

When monitoring an even smaller subset (1a,1c,1d,2a,2c,2d,3b and all nodes 3d-9d), the result is an estimate with the four mixture components shown in Fig. 8.5. The estimated marginal OD matrix has an RMSE of over 58. We attribute this to the fact that many of the origin and destination nodes are not observed when subsampling. The OD estimates show issues like zone 5 being a starting zone (where the ground truth is that only 1 and 2 are starting zones). However 5 is the first zone observed for some vehicles which started in zone 1. Therefore we see the same path behavior, but we are missing the correct origin leading to a high RMSE in the OD matrix estimate. We also can look at the relative $\ell_2$ error in the path-length distribution which is 0.7668 for this scenario. This demonstrates that even though we are missing some observed starting and terminating states we can still garner an accurate estimate according to the path-length distribution (as is also visible in the routing trends of Figs. 8.5(a), 8.5(b), 8.5(c), and 8.5(d)). As future work we will investigate methods to correct the OD estimate errors for this case by using additional traffic count data from sensors such as inductive loops to better estimate the in-flow and out-flow of a zone.

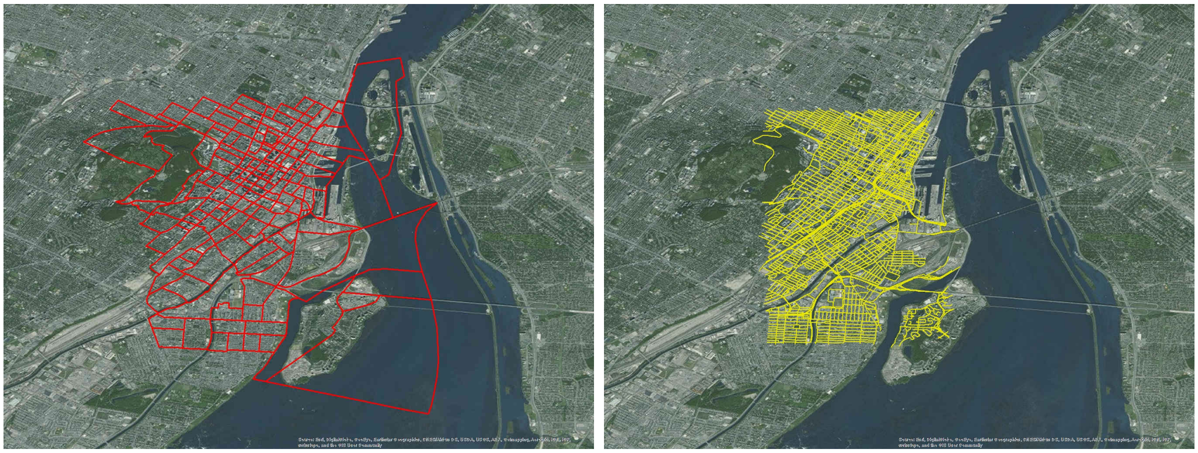
### 8.5.3 Application to the Montreal Regional Network

Next we study the performance of the proposed model and OD matrix estimation method using data simulated with a real road network based in downtown Montréal, Canada. A regional traffic assignment model for the Montreal metropolitan area is described in [20]. The model takes as input the 2008 Origin-Destination (OD) trip data for the Montréal region provided by Montréal's *Agence Métropolitaine de Transport* and assigns it on the network using a stochastic assignment in the PTV Visum platform [21]. The regional network consists of 127,217 road links and 90,467 nodes associated with over 1,500 traffic analysis zones. It also contains various road characteristics such as the type, length, speed limit, capacity, and number of lanes [20]. Note that this model has been validated using both traffic counts [22] and speed data collected using GPS [23].

The output from the traffic assignment procedure results in an array that contains a detailed account of all the paths connecting pairs of OD zones and the load on each path for every hour of the day. For this simulated analysis, we use only a portion of the network from [20] which represents the downtown core of Montréal. The subset of zones and links are shown in Figs. 8.6(a) and 8.6(b), respectively. We then use the path loads reported in [20] and simulate a population of 2 million vehicles (roughly the number of registered vehicles in the greater Montréal region). These vehicles are assigned randomly to zones and sent driving along the prescribed paths following the loads outlined in the simulations.

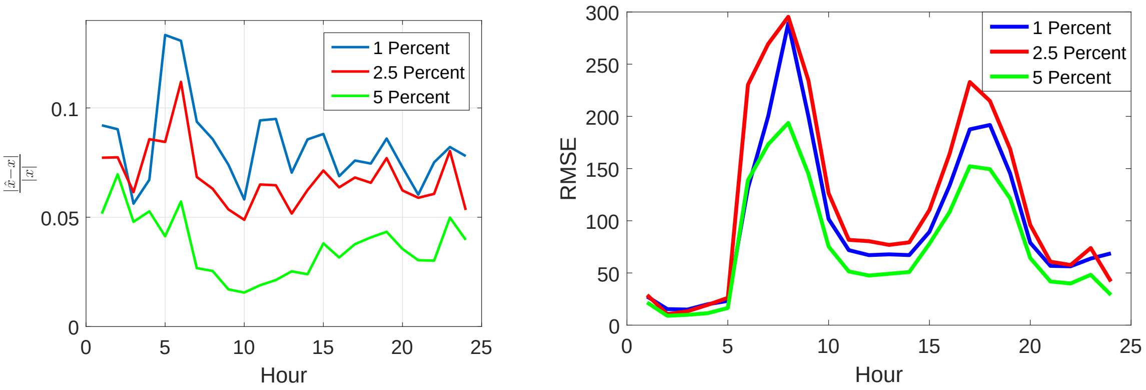
Artificial sensors are placed at links within the network at randomly sampled locations. In order to select a meaningful subset of the available links, we rank the 8146 links in this network according to how many paths each link appears in. For example a section of highway will appear in many more paths than a rural, dead-end road segment. This results in a vector of counts, $C$, where each element, $C_i$, is the path frequency count. We then normalize this vector so that it sums to one, making it a proportion of how frequently each link is utilized. We then sample 1% (81 sensors), 2.5% (204 sensors), and 5% (408 sensors) following these proportions. We call these collections of sensors $S_1$, $S_{2.5}$, and $S_5$ respectively. In the sampling procedure, we further enforce that $S_1 \subset S_{2.5} \subset S_5$. We then simulate populations of traffic traversing the road network, and for each subset of monitored links, when a vehicle traverses that link, we record the ID of the vehicle and the ID of the link traversed. These monitored links constitute the state space of the sensor network of our model.

A common problem with OD matrix estimation without prior information is that not having sensors in a specific zone or not having a sensor on a specific artery will result is a high error in the OD matrix estimate. To illustrate this, consider two zones, 1 and 2, which are adjacent to each other. If zone 2 is the true destination of 20,000 vehicles originating



(a) (b)

**Fig. 8.6** (a) The zones which are considered in the PTV Visum simulations. Each zone is denoted by a red boundary line. There are 157 distinct zones in this downtown portion of Montreal, Quebec, Canada.(b) The links (road segments) which are bounded by the zones denoted in (a) for the downtown core of Montreal. In this case the yellow lines denote the links themselves where in (a) the lines denote zone boundaries. There are 8146 distinct links in this downtown Montreal network segment.



(a) (b)

**Fig. 8.7** (a) Plot of the relative decrease in the $\ell_2$ norm of the error on the path-length distributions. (b) The average RMSE between the true and estimated marginal OD matrices for the three scenarios of selected link proportions.

from zone 1, but there are no sensors in zone 2, then without prior information (a prior OD matrix) one would only estimate that the vehicles get as far as the last sensor available, so it would appear that the 20,000 vehicles end up in zone 1. Then when calculating a traditional OD error metric (such as RMSE) we would have an incredibly large error for a relatively small mis-estimation.

To address this, we examine the path-length distribution metric to asses the model fit. We first estimate the OD matrix between the monitored links in our simulation. Then we map the links to zones in order to get an OD matrix estimate between the zones of the network. With the zonal OD matrix estimate, we then compute the estimated distance between pairs of zones using the same histogram density estimate for the path-lengths as described above. We bin the path lengths in units of 0–1 km, 1–2 km, ..., etc. Then we compute the relative $\ell_2$ norm error which gives us a relative $\ell_2$ norm error for a density estimate on the estimated zonal OD matrix $\hat{x}(t)$ versus the true zonal OD matrix $x(t)$ for hour $t$. We plot the resulting $\ell_2$ norm errors averaged over 5 randomly generated datasets for each set of monitored links. The resulting plot of the error for the three different scenarios is shown in Fig. 8.7(a). Here we can see that the relative $\ell_2$ norm error for the 1% of the links is the worst error. We can also see that the error decreases as the number of monitored links increases which also agrees with what one would expect. We can see that these errors are quite reasonable, even with only using 1% of links in the estimation scheme. This method again requires no prior OD matrix, as opposed to more traditional OD matrix estimation methods.

In order to visualize how a reasonable path-length error may still have a very large RMSE error, we show the RMSE errors in Fig. 8.7(b) for the three sensor subsets. The scale of the average RMSE estimate error will reach nearly 300, which is quite high. Both Fig. 8.7(a) and Fig. 8.7(b) again are demonstrating the average error for 5 random trials of each scenario of selected link proportions. Therefore we can see the hour-by-hour the comparison between the path-length error estimate and the RMSE. We can see that though the RMSE error will spike with the morning and afternoon rush-hours, the path-length distribution will remain much more stable showing that the estimated model is still accurate when estimating origin to destination areas even though the *exact* origin and destination zones may be mis-estimated yielding a high RMSE value.

The proposed method provides a way of estimating routing information for vehicle mediums which are not traditionally available or difficult to gather, for exampling trucking routes since this requires coordination from the carrier company. The proposed method allows a way of estimating the routes associated with OD pairs beyond just traditional truck counts without coordinating with the carrier company. Even in the estimation scheme used with this more accurate simulation of Montréal downtown traffic, there is a wealth of additional information encoded in the model. At each hour, a mixture model, as previously defined, has been fit. This model allows a transportation planner to investigate specific OD pairs which naturally separate from the others in the algorithm and can further view the routes of specific OD pairs providing a greater wealth of information to traffic engineers than available with traditional OD estimation techniques.

## 8.6 Conclusion

This article proposes a new approach to model and infer static and time-varying OD matrices using AVI data. In addition to demonstrating that with sufficient sensor coverage we can estimate an OD matrix accurately, or at least one that is highly correlated with the true OD matrix, we can also perform the estimation without the need to provide a prior OD matrix. As a by-product of the proposed model and estimation technique, we decompose the OD matrix into elements which categorize different groups in the data. These groups are formed according to vehicles which follow similar routes. By utilizing this decomposition we can correlate OD matrix elements with the routes that are estimated by each component. Automatic vehicle identification data proves to contain a wealth of additional information from which future traffic estimation techniques can leverage.

### 8.6.1 Future Work

A future avenue of research is to incorporate traffic count based sensors to try and correct errors in the OD estimates by this AVI-only method. One approach may be to have traffic count sensors at the in-flow and out-flow links of zones which are not monitored by AVI sensors. Therefore the volume in and out of the zone can be readily calculated. If the volume shows that vehicles do not originate at the specified zone (because the in-flow matches the out-flow) and the estimation technique states that the zone is an origin or destination then we know we are incorrect. Therefore the errors in the proposed, AVI-only based method, could be corrected or adjusted to provide more accurate OD matrices still without the need to provide a prior OD matrix.

## 8.7 Glossary

**Term** – **Definition**

OD Matrix – Origin-Destination Trip Table or Origin-Destination Matrix

AVI – Automatic Vehicle Identification

LPR – License Plate Recognition

$\Omega$ – The state space of the Markov model (sensors in the network)

$C$ – The number of sensors in the network

$n^i$ – The number of observations of vehicle $i$

$x^i_j$ – The state where vehicle $i$ was observed in its $j^{th}$ observation

$X^i$ – The transition-count matrix for vehicle $i$

$x^i$ – The observation sequence of vehicle $i$

$t$ – Time-window

$N(t)$ – The number of vehicles observed (equiv. the number of distinct observation sequences) in time-window $t$

$X(t)$ – The collection of observation sequences $x^i, i \in \{1, ..., N(t)\}$ in time-window $t$

RMSE – Root mean squared error

$\pi$ – The initial state distribution for a discrete-time Markov chain

$P$ – The transition matrix for a discrete-time Markov chain

$b$ – Matrix of parameters to Bernoulli distributions which govern the terminating/destination state for a vehicle given its starting state

$\phi$ – The collection of parameters describing a component in the mixture model ($\phi = \{\pi, P, b\}$)

$M(t)$ – The number of components in the mixture model in time-window $t$

$m$ – Denotes the $m^{th}$ mixture component ($m \in \{1, ..., M(t)\}$)

$\alpha(t)$ – The vector of mixture weights for the mixture model in time-window $t$

$\mathbf{\Theta}(t)$ – The collection of all model parameters in time-window $t$

Dir() – The Dirichlet distribution

**Dir**() – A vector of Dirichlet distributions

Beta() – The Beta distribution

**Beta**() – A matrix of Beta distributions

**Term – Definition**

Pois() – The Poisson distribution

$\lambda(t)$ – The parameter to the Poisson distribution governing component birth in time-window $t$

$n_b(t)$ – The number of components born in time-window $t$

$\alpha'(t)$ – The *rescaled* $\alpha$ vector after births and deaths have occurred

$\widehat{OD}^{(m)}(t)$ – The estimated OD matrix for mixture component $m$ in time-window $t$

$\widehat{OD}(t)$ – The estimated *marginal* OD matrix in time-window $t$

$C_a$ – The link capacity

$c_a(0)$ – The link free-flow rate

$S_a(x)$ – The average travel time of a vehicle on link $a$

$f(\mathbf{v})$ – The objective function for the user equilibrium optimization problem

$v_a$ – The volume of traffic on link $a$

$\alpha_{ij}^{ar}$ – The link-path identifying variable

$x_{ij}^{r}$ – The number of vehicles on path $r$ from origin $i$ to destination $j$

$L_{KL}$ – The KL divergence limit for the Automatic Hard EM algorithm

$L_\alpha$ – The mixture weight trimming limit for the Automatic Hard EM algorithm

# References for *A Mixture Model Approach to Origin-Destination Matrix Estimation with Routing Information*

[1] M. Gu, B. Xu, and Y. Hu, "Urban public transportation network planning method based on transit-oriented strategy," in *2012 International Conference on Computer Application and System Modeling, Processding of the*, ser. Advances in Intelligent Systems Research, 2012.

[2] W. A. O'Neill, "Origin-destionation trip table estimation using traffic counts," Ph.D. dissertation, University of New York, Buffalo, 1987.

[3] J. P. Rodrigue, C. Comtois, and B. Slack, *The Geography of Transport Systems*, 3rd ed. Routledge, 2009.

[4] L. G. Willumsen, "Estimation of an O-D matrix from traffic counts - a review," Institute of Transport Studies, Tech. Rep. 99, 1978.

[5] E. Castillo, P. Jiménez, J. M. Menéndez, and M. Nogal, "A Bayesian method for estimating traffic flows based on plate scanning," *Transportation*, vol. 40, no. 1, pp. 173–201, 2013.

[6] H. D. Sherali, A. Narayanan, and R. Sivanandan, "Estimation of origin-destimation trip-tables based on a partial set of traffic link volumes," *Tranprn Res.-B*, vol. 37, no. 9, pp. 815–836, Nov. 2003.

[7] A. Peterson, "The origin-destinaton matrix estimation problem - analysis and computations," Ph.D. dissertation, Linköping University, Linköpings universitet, SE-601 74 Norrköping, Sweden, 2007.

[8] E. Bas, A. M. Tekalp, and F. S. Salman, "Automatic vehicle counting from video for traffic flow analysis," in *2007 IEEE Intelligent Vehicles Symp.*, June 2007, pp. 392–397.

[9] N. V. D. Zijpp, "Dynamic OD-matrix estimation from traffic counts and automated vehicle identification data," *Tranprn Res. Record: J. of the Transprn Res. Board*, vol. 1607, pp. 87–94, 1997.

[10] M. Dixon and L. Rilett, "Population origin-destination estimation using automatic vehicle identification and volume data," *J. Transp. Eng.*, vol. 131, no. 2, pp. 75–82, Feb. 2005.

[11] E. Castillo, J. M. Menéndez, and P. Jiménez, "Trip matrix and path flow reconstruction and estimation based on plate scanning and link observations," *Transportation Research Part B*, vol. 42, pp. 455–481, 2008.

[12] G. Michau, N. Pustelnik, P. Borgnat, P. Abry, A. Nantes, A. Bhaskar, and E. Chung, "A primal-dual algorithm for link dependent origin destination matrix estimation," April 2016, [arxiv preprint arXiv:1604.00391 [math.OC]].

[13] Y. Feng, J. Sun, and P. Chen, "Vehicle trajectory reconstruction using automatic vehicle identification and traffic count data," *J. of Adv. Transportation*, vol. 49, pp. 174–194, Feb. 2014.

[14] H. Yang, "Heuristic algorithms for the bilevel origin-destination matrix estimation problem," *Transportation Research Part B: Methodological*, vol. 29, no. 4, pp. 231–242, Aug. 1995.

[15] S. Lawlor and M. G. Rabbat, "Estimation of time-varying mixtures of Markov chains: An application to road traffic modeling," *T. on Sig. Proc.*, vol. PP, no. 99, pp. 1–1, September 2016.

[16] M. Stephens, "Bayesian analysis of mixture models with an unknown number of components- an alternative to reversible jump methods," *The Annals of Statistics*, vol. 28, pp. 40–74, 2000.

[17] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. of the Royal Statatistics Society Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[18] S. C. Dafermos and F. T. Sparrow, "The traffic assignment problem for a general network," *J. of Res. of the National Bureau of Standards*, vol. 73B, no. 2, pp. 91–118, April 1969.

[19] J. G. Wardrop and J. I. Whitehead, "Correspondence. some theoretical aspects of road traffic research," *Proceedings of the Institution of Civil Engineers*, vol. 1, no. 5, pp. 767–768, 1952.

[20] T. Sider, A. Alam, M. Zukari, H. Dugum, N. Goldstein, N. Eluru, and M. Hatzopoulou, "Land-use and socio-economics as determinants of traffic emissions and individual exposure to air pollution," *J. of Transport Geography*, vol. 33, no. 0, pp. 230 – 239, 2013.

[21] PTV Vision, *PTV Vision, 2009*, VISUM 11.0 Basics ed., PTV AG, Karlsruhe, Germany, 2009.

[22] T. Sider, A. Alam, W. Farrell, M. Hatzopoulou, and N. Eluru, "Evaluating vehicular emissions with an integrated mesoscopic and microscopic traffic simulation," *Canadian J. of Civil Eng.*, vol. 41, no. 10, pp. 856–868, Aug. 2014.

[23] A. Alam, G. Ghafghazi, and M. Hatzopoulou, "Traffic emissions and air quality near roads in dense urban neighborhoods: Using microscopic simulation for evaluating effects of vehicle fleet, travel demand, and road network changes," *J. of the Transportation Research Board*, vol. 2427, pp. 83–92, 2014.

# Chapter 9

# Summary and Future Work

The overwhelming theme of this thesis is the utilization of AVI data, initially for the identification of convoys in traffic and then further to improve estimation of traffic flows. The growing use of this class of sensors provides an opportunity to gain more in-depth information into traffic than is currently available. The work in this thesis is only an initial foray into the wealth of information this data can provide.

The first work in this thesis, presented in Chapter 4, proposes a novel organization of the AVI data for use with sequential detection and estimation. It further proposed a mathematical model for what it means to be a convoy of vehicles, i.e. two or more vehicles which follow dependent routes through a network of sensors with dependent transition times between the states of the network. It also proposes a model which fits nicely into the Markov chain framework allowing for sequential detection to be performed under the Markov assumptions of the model.

The proposal of a mixture of first-order Markov models was then extended by allowing the mixture model to vary with time. This proposed time-varying mixture model (TVMM) of discrete-time Markov chains (DTMC) allows for a dynamic model for estimating the highly variant traffic patterns which emerge in real networks. In addition to the model presented in Chapter 6, I propose a novel approximate inference algorithm which is a modified hard-EM algorithm for choosing the correct mixture model order as well as estimating the parameters of the mixture components. In addition the utilization the KL divergence as a penalization on the model order complexity, I believe, possibly may open new avenues of research into model order estimation with algorithms that generatively select the model order instead of annealing-type algorithms.

Lastly this research concludes in Chapter 8 with a modification of the TVMM DTMC model of Chapter 6 to account for the state a vehicle is last seem at or terminates at within the sensor network. This further allows the computation of a mixture of OD matrices which includes information on the routes vehicles are following between OD pairs. Breaking an OD matrix into a more flexible mixture of sub OD matrices has not been performed to the best of my knowledge and allows for a wealth of additional information which can be utilized to more accurately visualize traffic information for city planners and law enforcement. In addition these mixture models still include the time-varying aspect of the TVMM DTMC model and therefore allow city planners and engineers to visualize the changes in their traffic structures over time. In addition to the extensions mentioned in the three manuscripts in Chapters 4, 6, and 8 I present in the following sections a selection of possible avenues for future research which can be extended from the work of this thesis.

## 9.1 Convoy Detection Algorithm of More Than Two Vehicles

Adjusting the tracking algorithm presented in Chapter 4 to track groups of vehicles larger than 2 is not a trivial task. This is because for each possible convoy vehicle pair, a tracking state is created. If we track groups of varying size, the number of vehicle combinations required to be tracked will grow exponentially. Therefore we present a post-analysis step to detect if a group of vehicles larger than two are traveling as a convoy.

However a future task of deriving a tracking algorithm where each tracked group has a variable and dynamic number of vehicles in the convoy may help improve the algorithm's performance detecting larger groups. It would require starting a tracking state for a large group of vehicles seen together and as time continues annealing the vehicles in the group to only those which continue travelling as the group. I do not believe this to be a trivial task since a major problem which would need to be solved is the problem of not starting tracking states for all combinations of possible convoy groups.

## 9.2 Mixture of Gaussian Distributions

Future work on the inference algorithm presented in Chapter 6 includes extending this algorithm to mixtures of distributions other than Markov chains. The Automatic Hard

EM algorithm could be used to estimate a time-varying mixture of Gaussian distributions. In this scenario, the prior distribution for a mixture of Guassian distributions can still be fully specified. Specifically the covariance matrix of a component from time-window $t-1$ becomes the inverse-scale matrix of a Gaussian-inverse-Wishart distribution in time-window $t$ and the mean of the component from $t-1$ becomes the mean of a Gaussian-inverse-Wishart distribution. This evolution would look like

$$\left(\mu^{(m)}(t), \Sigma^{(m)}(t)\right) \sim NIW(\mu^{(m)}(t-1), 1, \Sigma^{(m)}(t-1), D-1) \tag{9.1}$$

where $D$ is the dimensionality of the Gaussian distributions. This would yield MAP parameter estimates at each hard EM step of [Mur07]

$$\hat{\mu}^{(m)}(t) = \frac{\mu^{(m)}(t-1) + n\bar{x}}{1+n} \tag{9.2}$$

$$\hat{\Sigma}^{(m)}(t) = \Sigma^{(m)}(t-1) + C + \frac{n}{1+n}\left(\bar{x} - \mu^{(m)}(t-1)\right)\left(\bar{x} - \mu^{(m)}(t-1)\right)^T \tag{9.3}$$

where $\bar{x}$ is the sample mean, $n$ is the number of datapoints, and [Mur07]

$$C = \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T. \tag{9.4}$$

By extending this algorithm to a mixture of single or multivariate Gaussian distributions, we could compare this algorithm with existing work in the literature on estimating a time-varying, finite model-order, mixture model. The application presented in this thesis, of a mixture of Markov chains, appears to have had limited focus making an assessment of the algorithm's performance compared to alternative algorithms difficult. With a wider field of previous study, comparison to other algorithms becomes much more available. Ideally we would like to demonstrate that this algorithm is useful for any time-varying mixture model of distributions where a conjugate prior distribution is available so a MAP estimate based on the previous time-window can be defined.

## 9.3 Deeper Investigation Into the KL Divergence Between Mixture Model Components

Chapter 6 presented a novel approach using the KL divergence as a measure of when a mixture model estimate is overfitting. In the Automatic Hard EM algorithm, the KL divergence is computed between all mixture components after the algorithm converges in each time-window. These KL divergences are then investigated to make sure that none of the components have a KL divergence which is too small as this would indicate that those two components are describing the same group of data. This is a common scenario in overfitting mixture models, where one collection of data which is present is explained by more components than are necessary. The KL divergence is unique in that it allows us to evaluate the distance between probability distributions and therefore is suitable for the task at hand.

The KL divergence between two random Markov chains is a random variable parametrized by the transition matrix ($P$) and initial state distribution ($\pi$) of two different Markov chains. The closer the two Markov chains are the smaller the resulting KL divergence between the two components will be. The shape of this distribution however has not yet been studied. A future avenue of research is to examine if this distribution can be described in a closed-form. If a closed-form distribution can be derived, then a statistical anomaly detection approach to choosing the limit $L_{KL}$ could be utilized to determine a $p$-value for if the estimated KL divergence is a valid value or is anomalous (too close to 0, and therefore the components should be merged).

## 9.4 Addition of Traffic Count Data in Origin-Destination Estimation

Another avenue of research which can be derived from this thesis is the consideration of additional forms of data into the origin-destination matrix estimation schemes proposed in Chapter 8. One proposed form of data which would be helpful to include is traffic count data gleamed from inductive loops. As stated in Chapter 8, inductive loops are deployed below roads and return the number of vehicles which pass on that road segment, or in other words the traffic count on a road. This class of traffic sensor is already widely deployed and even today is a popular form of traffic sensor being deployed. By extending the TVMM

DTMC model to include this type of data, we believe that we can improve the OD matrix estimates in accuracy and correct scale estimate errors present when utilizing only AVI data as our method does. In addition, due to the more readily available nature of data gathered from inductive loop sensors, we can additionally extend the coverage area with additional routing information therefore increasing the information contained in the estimated model.

## 9.5 Handling Errors in Recorded LPR Data

All of the models for traffic estimation presented in this thesis assume a perfect read-rate of LPR data, or any AVI-type of data. This is likely infeasible in real scenarios due to real-world conditions. For example, in LPR data the possibility of having the OCR algorithm mistake a *B* for an *8* is frequent. Even with an advertised rate in some regions exceeding 95% accuracy, there is still room for a lot of read errors. Therefore we propose handing this by forming a distribution over the expected errors. Again looking at the B-8 example, if during training the OCR algorithm the percentage of B's recorded as 8's is 10% with 90% of the time a B is read accurately, that yields a small multinomial distribution. Now when computing the path of a vehicle, we assign weighted probabilities based on these multinomial distributions.

As a small example, consider the identifier *ABC*. Now given an estimated multinomial on the error of reading a B from before, we assign any observation of ABC to 10% as A8C and 90% as ABC. This would further mean that the transition count matrix used in many of the proposed estimation schemes would contain non-integer values based on these multinomial distributions. By creating a distribution over the error space of an LPR read, the input to the algorithm would no longer be an integer count-matrix but a weighted distribution over count matricies. However due to the form of the generative model, we can simply sum these partial, weighted count matrices as if they were whole count matrices to garner the same estimation technique. The difficulty would simply being deriving the distribution over the state-space of possible LPR read errors.

## 9.6 Optimal Selection of AVI Sensor Locations

Another interesting future research field is to compute the optimal (or sub-optimal) placement of AVI sensors on a road network to maximize the convoy detection or traffic estimation probabilities. Intuitively the optimal placements are likely large road-crossing in an urban environment, however computing additional sensor placements which allow for the largest gain in estimation and detection information would allow the entity installing the sensors to minimize their costs for the largest gains.

## 9.7 Extending the First-Order Markov Model

The works presented in Chapters 4, 6, and 8 all utilize first-order Markov model estimates for how a vehicle's path through the sensor network is governed. In a future work, I would like to investigate how extending this model to a $n$-th order Markov model impacts the estimates for traffic flow throughout the day. I suspect that having higher-order model orders will increase the average model-order complexity however it may be worth the added complexity for the gains in the estimation accuracy of the TVMM model.

# References

[ABT$^+$15] A. Alam, B. Besselink, V. Turri, J. Mårtensson, and K.H. Johansson. Heavy-duty vehicle platooning for sustainable freight transportation: A cooperative method to enhance safety and efficiency. *IEEE Cont. Sys. Mag.*, 35(6):34–56, Dec. 2015.

[AGH14] A. Alam, G. Ghafghazi, and M. Hatzopoulou. Traffic emissions and air quality near roads in dense urban neighborhoods: Using microscopic simulation for evaluating effects of vehicle fleet, travel demand, and road network changes. *J. of the Transportation Research Board*, 2427:83–92, 2014.

[Agr13] Alan Agresti. *Categorical Data Analysis*. Wiley Series in Probability and Statistics. Wiley, 2013.

[Aka74] H. Akaike. A new look at the statistical model identification. *IEEE Trans. on Automatic Control*, 19(6):716 – 723, 1974.

[BBA97] S. T. Buckland, K. P. Burnham, and N. H. Augustin. Model selection: An integral part of inference. *Biometrics*, 53(2):603–618, 1997.

[CB01] A. Corduneanu and C. M. Bishop. Variational Bayesian model selection for mixture distributions. In *Intl. Conf. on Artificial Intelligence and Statistics*, pages 27–34. Morgan Kaufmann, Jan. 2001.

[CDD07] F. Caron, M. Davy, and A. Doucet. Generalized Polya urn for time-varying Dirichlet process mixtures. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence, UAI*, July 2007.

[CK08] J. Chen and A. Khalili. Order selection in finite mixture models with a nonsmooth penalty. *J of the American Statistical Association*, 103(484):1674–1683, 2008.

[CMJ08] E. Castillo, J. M. Menéndez, and P. Jiménez. Trip matrix and path flow reconstruction and estimation based on plate scanning and link observations. *Transportation Research Part B*, 42:455–481, 2008.

[DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Statatistics Society Ser. B*, 39(1):1–38, 1977.

[DR05] M. Dixon and L. Rilett. Population origin-destination estimation using automatic vehicle identification and volume data. *J. Transp. Eng.*, 131(2):75–82, Feb. 2005.

[Eva70] A.W. Evans. Some properties of trip distribution methods. *Transportation Research*, 4(1):19–36, April 1970.

[FJ02] Mario A.T. Figueiredo and A.K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3):381–396, March 2002.

[Her10] A.O. Hero. Geometric entropy minimization (GEM) for anomaly detection and localization. In *Conf. on Neural Information Processing Systems*, Vancouver, Canada, Dec. 2010.

[HHZ$^+$11] A. Homayounfar, A.T.S. Ho, N. Zhu, G. Head, and P. Palmer. Multi-vehicle convoy analysis based on ANPR data. In *Intl. Conf. on Imaging for Crime Detection and Prevention*, pages 1–5, Nov. 2011.

[How07] Ronald A. Howard. *Dynamic Probabilistic Systems : Semi-Markov and Decision Processes*. Dover Publications, 1 edition, 2007.

[HPZ13] T. Huang, H. Peng, and K. Zhang. Model selection for Gaussian mixture models. *ArXiv e-prints*, Jan. 2013.

[JLO07] Christian S. Jensen, D. Lin, and Beng Chin Ooi. Continuous clustering of moving objects. *IEEE Trans. on Knowledge and Data Eng.*, 19(9):1161–1174, Sept 2007.

[JYZ$^+$08] Hoyoung Jeung, Man Lung Yiu, Xiaofang Zhou, Christian S. Jensen, and Heng Tao Shen. Discovery of convoys in trajectory databases. In *Intl. Conf. on Very Large Data Bases*, pages 1068–1080, Auckland, New Zeland, Aug. 2008.

[KMB05] Panos Kalnis, Nikos Mamoulis, and Spiridon Bakiras. On discovering moving clusters in spatio-temporal data. In *Intl. Symp. on Spatial and Temporal Databases*, pages 364–381, Angra dos Reis, Brazil, Aug. 2005.

[Koc02] W. Koch. Information fusion aspects related to GTMI convoy tracking. In *Fifth Int. Conf. on Information Fusion*, volume 2, pages 1038–1045, 2002.

[Law13] Sean Lawlor. Detecting convoys in networks of short-ranged sensors. Master's thesis, McGill University, Montreal, Quebec, 2013.

[Mur07] Kevin P. Murphy. Conjugate Bayesian analysis of the Gaussian distribution. Technical report, University of British Columbia, Department of Computer Science, 2007.

[O'N87] W. A. O'Neill. *Origin-Destionation trip table estimation using traffic counts*. PhD thesis, University of New York, Buffalo, 1987.

[Per09] Theodore J. Perkins. Maximum likelihood trajectories for continuous-time Markov chains. In *Advances in Neural Information Processing Systems 22*, pages 1437–1445. 2009.

[Pet07a] A. Peterson. *The origin-destinaton matrix estimation problem - analysis and computations*. PhD thesis, Linköping University, Linköpings universitet, SE-601 74 Norrköping, Sweden, 2007.

[Pet07b] Anders Peterson. *The Origin-Destination Matrix Estimation Problem - Analysis and Computations*. PhD thesis, Institutionen fr teknik och naturvetenskap, Linkpings universitet, SE-601, 74 Norrkping, Sweden, 2007.

[PPR09] Evangeline Pollard, Benjamin Pannetier, and Michele Rombaut. Convoy detection processing by using the hybrid algorithm (GMCPHD/VS-IMMC-MHT) and dynamic Bayesian networks. In *Int. Conf. on Information Fusion*, volume 12, Seattle, WA, USA, July 2009.

[PRP10] E. Pollard, M. Rombaut, and B. Pannetier. Bayesian networks vs. evidential networks: An application to convoy detection. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Dortmund, Germany, Jun. 2010.

[PTV09] PTV Vision. *PTV Vision, 2009.* PTV AG, Karlsruhe, Germany, VISUM 11.0 Basics edition, 2009.

[Ris78] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[SAF$^+$14] T. Sider, A. Alam, W. Farrell, M. Hatzopoulou, and N. Eluru. Evaluating vehicular emissions with an integrated mesoscopic and microscopic traffic simulation. *Canadian J. of Civil Eng.*, 41(10):856–868, Aug. 2014.

[SAZ$^+$13] T. Sider, A. Alam, M. Zukari, H. Dugum, N. Goldstein, N. Eluru, and M. Hatzopoulou. Land-use and socio-economics as determinants of traffic emissions and individual exposure to air pollution. *J. of Transport Geography*, 33(0):230 – 239, 2013.

[Sch78] G. Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464, 1978.

[SK10] C. Scott and E. Kolaczyk. Nonparametric assessment of contamination in multivariate data using generalized quantile sets and FDR. *J. of Computational and Graphical Stat.*, 19(2):439–456, Jun. 2010.

[SLHC10] Mei-Chiun Shih, Tze Leung Lai, Joseph F Heyse, and Jie Chen. Sequential generalized likelihood ratio tests for vaccine safety evaluation. *Stat. in Medicine*, 29(26):2698–2708, November 2010.

[SNS03] H. D. Sherali, A. Narayanan, and R. Sivanandan. Estimation of origin-destination trip-tables based on a partial set of traffic link volumes. *Tranprn Res.-B*, 37(9):815–836, Nov. 2003.

[Ste00] M. Stephens. Bayesian analysis of mixture models with an unknown number of components- an alternative to reversible jump methods. *The Annals of Statistics*, 28:40–74, 2000.

[TJ03] M. Thottan and C. Ji. Anomaly detection in IP networks. *IEEE Trans. on Sig. Proc.*, 51(8):2191–2204, Aug. 2003.

[vJD15] S. van de Hoef, K. Johansson, and D. Dimarogonas. Fuel-optimal centralized coordination of truck-platooning based on shortest paths. In *American Control Conf.*, Chicago, IL, Jul. 2015.

[VVK03] J.J. Verbeek, N. Vlassis, and B. Krose. Efficient greedy learning of Gaussian mixture models. *Neural Computation*, 15(2):469–485, 2003.

[Wal66] Abraham Wald. *Sequential Analysis*. Wiley Publication In Statistics. John Wiley & Sons, Inc., 1966.

[WBH12] T. Weiherer, E. Bouzouraa, and U. Hofmann. A generic map based environment representation for driver assistance systems applied to detect convoy tracks. In *IEEE Intl. Conf. on Intelligent Transportation Systems*, Anchorage, AK, Sep. 2012.

[YD12] Jeremy Yeoman and Matt Duckham. Decentralized network neighborhood information collation and distribution for convoy detection. In *Seventh Int. Conf. on Geographic Information Science*, Columbus, OH, September 2012.

[Zij97] N. Van Der Zijpp. Dynamic OD-matrix estimation from traffic counts and automated vehicle identification data. *Tranprn Res. Record: J. of the Transprn Res. Board*, 1607:87–94, 1997.