Statistical modelling and inference for traffic networks

Katharina Parry
Institute of Fundamental Sciences
Massey University

A thesis submitted for the degree of
Doctor of Philosophy

2012

This thesis is my own account of my research and contains, as its main content, work that has not been previously submitted for a degree at any university.

Katharina Parry
September 2012

Acknowledgements

The world needs more people like Martin Hazelton. Many many thanks for the patience, guidance, encouragement, more encouragement and generally excellent supervision. I would like to say thank-you to Olly, who has been a great companion throughout the bad, not as bad and sometimes downright ugly stages of doing this PhD and for believing in me even when I did not. Many cheers to Leo, who knows how to makes things happen when I cannot get them to work, as well as moral support. Overall, I appreciate that nobody who was involved with me and this thesis in these past years has lost their sense of humour. And finally, I express my gratitude for the Marsden Fund scholarship as it provided the opportunity for me to do this research.

Abstract

There are two facets that are important in providing reliable forecasts from observed traffic data. The first is that the model used should describe and represent as many characteristics of the system as possible. The second is that the estimates of the model parameters need to be accurate. We begin with improved methods of statistical inference for various types of models and using various types of data; and then move onto the development of new models that describe the day-to-day dynamics of traffic systems.

Calibration of transport models for traffic systems gives rise to a variety of statistical inference problems, such as estimation of travel demand parameters. Once the ways in which vehicles move through the network are known, statistical inference becomes straightforward, however, at present, the data available are predominantly vehicle counts from a set of links in the network. The fundamental problem is that these vehicle counts do not uniquely determine the route flows, as there are a large number of possible route flows that could have led to a given set of observed link counts.

A solution to this problem is to simulate the latent route flows conditional on the observed link counts in a Markov Chain Monte Carlo sampling algorithm. This is challenging because the set of feasible route flows will typically be far too large to enumerate in practice, meaning that we must simulate from a set that we cannot fully specify. An innovative piece of work here was the extension of an existing sampling methodology that works only for linear networks to be applicable for tree networks. In simulation studies where we use the sample to estimate average route flows, we show that our method provides more reliable estimates than generalised least squares methods. This is to be expected given that our method exploits information available via second order properties of the link counts.

We provide another demonstration of how this generalised sampler can be applied whenever the need to sample from the set of latent route flows is pivotal for making statistical inference. We use the sampler to estimate travel demand parameters for day-to-day dynamic process models, an important class of model where the data has been collected on successive days and hence allows for inference using the evolution of the traffic flows over time.

A new type of data, route flows from tracked vehicles, is becoming increasingly available through emerging technologies. Our contribution was to develop a statistical likelihood model that incorporates this routing information into currently used link-count data only models. We derive some tractable normal approximations thereof and perform likelihood-based inference for these normal models under the assumption that the probability of vehicle tracking is known.

In our analysis we find that the likelihood shows irregular behaviour due to boundary effects, and provide conditions under which such behaviour will be observed. For regular cases we outline connections with existing generalised least squares methods. The theoretical analysis are complemented by simulation studies where we consider the tracking probability to be unknown and the effects on the accuracy in estimation of origin-destination matrices under estimated and/or misspecified models for this parameter.

Real link flow count data observed on a sequence of days can exhibit considerable day-to-day variability. A better understanding of such variability has increasing policy-relevance in the context of network reliability assessment and the design of intelligent transport systems. Conventional day-to-day dynamic traffic assignment models are limited in terms of the extent to which non-stationary changes in traffic flows can be represented.

In this thesis we introduce and develop an advanced class of models by replacing a subset of the fixed parameters in currently used traffic models with random processes. These resulting models are analogous to Cox process models. They are conditionally non-stationary given any realisation of the parameter processes. Numerical examples demonstrate that this new class of doubly stochastic day-to-day traffic assignment models is able to reproduce features such as the heteroscedasticity of traffic flows observed in real-life settings.

Publications arising from thesis

Parry, K. and Hazelton, M.L. (2012), ‘Estimation of origin-destination matrices from link counts and sporadic routing data’, *Transportation Research Part B* **46**,175-188.

Parry, K. and Hazelton, M.L. (2013), ‘Bayesian inference for day-to-day dynamic traffic models’. Accepted for publication in *Transportation Research Part B*.

Parry, K., Watling, D.P. and Hazelton, M.L. (2012),‘A new class of doubly stochastic day-to-day dynamic traffic assignment models’. Submitted for publication.

Contents

List of Figures

List of Tables

Chapter 1

Traffic network models

While the individual man is an insoluble puzzle, in the aggregate he becomes a mathematical certainty. You can, e.g., never foretell what any one man will be up to, but you can say with precision what an average number will be up to. Individuals vary, but percentages remain constant. So says the statistician.

(Arthur Conan Doyle)

1.1 Introduction

Traffic is everywhere, and it is on the rise. However, with higher volumes of motorised transport come higher levels of traffic congestion. Traffic jams are not only mentally frustrating, but also cause heightened levels of air pollution and an economic loss through extended travel times. This is hugely relevant for a country like New Zealand, which relies strongly on private transportation due to its low population density and large areas of urban sprawl. In an IBM survey (IBM, 2010), 47% of Auckland drivers reported that they had experienced a traffic jam of at least an hour or more in 2009. In the same survey, an accurate and timely road conditions information service was the third most popular demand behind more public transport and incentives for forgoing car trips altogether.

Traffic network models deliver this kind of information. They allow us to take a sample of vehicles observed in parts of the network at discrete points in time and uncover their relationship to the demand that generated the data. We use this information to estimate the model parameter values and their associated uncertainties which then provide important insights into road management and planning, especially with respect to road congestion control.

In this chapter we introduce and explain the notation of statistical network models in the context of transportation analysis. We begin with the two things that form the core of transport models, the network and the corresponding trip matrix. We then outline why statistical linear inverse problems provide a framework for studying inference for network-based models.

A network in transport modelling is a quantitative description of roads and junctions. Lines, called *links* , represent stretches of road; in complex networks these are often the main arteries in the network.

In this thesis, we deal with *directed* networks, where travel along a road is unidirectional and is indicated by an arrow in graphical representations. A two-way road is depicted as two opposite-direction links.

The lines connect circles, called *nodes*, which are junctions as well as points in the network where traffic originates and to which traffic is destined. The entry and exit nodes are called *origins* and *destinations*, respectively.

Once the set of nodes is defined, the movements through the network can be expressed in terms of the trip matrix, more commonly known as the *origin-destination (OD) matrix*. This matrix specifies the flow between every origin and every destination in the network. The ijth element of the OD matrix corresponds to the number of trips made from the ith origin node to the jth destination node. For convenience, we often order the OD pairs lexicographically and represent the OD matrix as a vector where the OD pairs are identified by a single index $o = 1, .., L$. We denote the actual OD flows as $\boldsymbol{w} = w_1, .., w_L$.

For illustration, we consider a toy example network with three nodes and four links, seen in Figure 1.1.

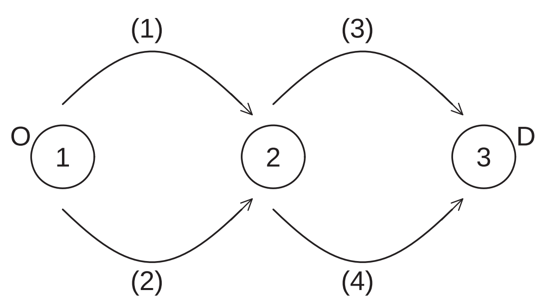


Figure 1.1: Illustrative network, with three nodes and four links.

We have a single OD pair, since the only origin is node 1 and all trips terminate at the third node. The OD flow vector is a single number, w, in this case.

However, we can see that there is more than one way to travel between the two nodes. The trips can be made via one of the four pairs of links (1,3), (1,4), (2,3) and (2,4).

A set of links that allow travel between a given OD pair is called a *route* or path. In principle, the number of routes that serve each OD pair can be quite large, however, it is unusual for more than six or seven routes to be employed for any given OD pair Bonsall et al. (1997). In practice, we will usually ignore unlikely routes in order to keep the number of routes N manageable from a computational perspective. We denote a given route flow as y_r, with $r = 1, ..., N$ and the N-vector of route flows as $\boldsymbol{y} = (y_1, \ldots, y_N)^\mathsf{T}$. Note that we index routes by a single integer, which may be defined through a lexicographical (partial) ordering based on the node sequences for each route within each OD pair.

Within the framework of OD matrix estimation we are interested in uncovering the expected movements from one point to another of all vehicles in the system over a long period of time. That is, instead of trying to reconstruct the L-vector of actual OD flows, we try to make inference about the L-vector of *mean OD flows*, $\boldsymbol{\mu} = \mathsf{E}(\boldsymbol{w})$ instead. The N-vector of *mean route flows* are denoted as $\boldsymbol{\lambda}$, where the average number of trips on a given route, λ_r are related to the oth mean OD flow, μ_o, as follows

$$\mu_o = \sum_{r \sim o} \lambda_r$$

where $r \sim o$ if and only if route r serves the oth OD pair. That is, each mean OD flow is an aggregation of the mean number of trips on all the routes that serve that OD pair. As an example, in the small network in Figure 1.1 with four routes serving the one OD pair, we have $\mu = \lambda_{(1,3)} + \lambda_{(1,4)} + \lambda_{(2,3)} + \lambda_{(2,4)}$.

In a network with *unique routing*, each OD pair corresponds to a single route only. Linear transit networks, as seen in Figure 1.2, are typical of unique routing networks in that there is only one option for travelling between each OD pair. For example, in the network seen below travel from origin node 1 to destination node 3 is restricted to the route using the first two links.

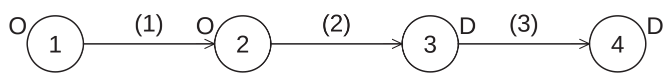


Figure 1.2: Network with property of unique routing: a linear network.

In this situation we have $\lambda_r = \mu_o$, where $r \sim o$. Consequently, for networks with unique routing the parameters of interest in OD matrix estimation are the mean route flows which are defined as $\lambda_r = \mathsf{E}(y_r)$ for $r = 1, ..., N$.

In more general networks without the unique routing property, we define ϱ_{or} as the probability of travelling between a given OD pair o using route r. We can then define the expected route flow rates $\boldsymbol{\lambda} = \mathsf{E}(\boldsymbol{y})$ in terms of the average OD flow rates $\boldsymbol{\mu}$ via the relationship $\lambda_r = \varrho_{or}\mu_o$. The route choice probabilities $\boldsymbol{\varrho}$ will typically be unknown and are considered a nuisance parameter in the model. In other words, the focus in OD matrix estimation is primarily on the estimation of the average route flow rates $\boldsymbol{\lambda}$.

The network seen in Figure 1.2 has more than one OD pair, O_1D_3, O_1D_4, O_2D_3 and O_2D_4. Due to the unique routing property of the network the OD pairs correspond to the four possible routes, y_{13}, y_{14}, y_{23} and y_{24}, respectively. Thus, we estimate a parameter vector of the four corresponding average route flow rates, $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)^\mathsf{T}$, where λ_1 is an estimate of the expected number of trips that begin at node 1 and end at node 3 and so on.

The mean route flows $\boldsymbol{\lambda}$ are by no means the only parameters of interest in traffic modelling. Information about the route flows can be used to calculate an array of other measures which describe the travel patterns observed. In other words, we can define a transport model in terms of a whole vector of unknown parameters that describe the traffic system, denoted herein as $\boldsymbol{\theta}$.

Mean route flow estimation is a special case where the vector $\boldsymbol{\theta}$ is equal to the expected route flows, $\boldsymbol{\lambda} = \mathsf{E}(\boldsymbol{y})$. Another statistical inference problem is the calibration of traffic assignment models, where the elements of $\boldsymbol{\theta}$ parametrise route choice probabilities (Sheffi, 1985).

Importantly, regardless of parameterisation, all transport modelling problems where the data involve traffic counts on network links, share a common structure and a common dilemma: although the route flows are ultimately of interest, they cannot be directly observed.

The emergence of cost-effective data collection technology such as GPS tracking and improved number plate scanning devices are promising developments, however, at present these advanced applications deliver at most partial information on travel movements.

At the time of writing, the main source of data are vehicle counts monitored at a low expense on a subset of M links in the network, $\boldsymbol{x} = (x_1, \ldots, x_M)^\mathsf{T}$. We assume that the locations of the links equipped with monitoring devices are chosen to be as informative as possible, see (Ehrlert et al., 2006; Yang and Zhou, 1998).

The link flows are linearly related to the route flows through the fundamental relationship

$$\boldsymbol{A}\boldsymbol{y} = \boldsymbol{x}, \tag{1.1}$$

where $\boldsymbol{A}$=(a_{lr}) the routing matrix with

$$a_{lr} = \begin{cases} 1, & \text{monitored link } l \text{ forms part of route } r \\ 0, & \text{otherwise.} \end{cases}$$

One example of a routing matrix is the matrix $\boldsymbol{A}$ affiliated with the network seen in Figure 1.1 with four links, where all four routes serve a single OD pair:

$$\boldsymbol{A} = \begin{array}{c} \\ \underset{1}{link} \\ \underset{2}{link} \\ \underset{3}{link} \\ \underset{4}{link} \end{array} \begin{array}{c} \begin{array}{cccc} \underset{1}{route} & \underset{2}{route} & \underset{3}{route} & \underset{4}{route} \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \end{array}.$$

Another example is the corresponding routing matrix of the transit network displayed in Figure 1.2, with three links and as many routes as OD pairs because of the unique routing property of the network:

$$\boldsymbol{A} = \begin{array}{c} \\ \underset{1}{link} \\ \underset{2}{link} \\ \underset{3}{link} \end{array} \begin{array}{c} \begin{array}{cccc} \underset{1}{route} & \underset{2}{route} & \underset{3}{route} & \underset{4}{route} \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \end{array}.$$

The routing matrix $\boldsymbol{A}$ summarises the network traffic structure in useful ways, e.g. the diagonal elements of $\boldsymbol{A}\boldsymbol{A}^{\top}$ count the number of routes passing through each link Tebaldi and West (1998).

In order to learn more about the route flows $\boldsymbol{y}$, we will want to invert the routing matrix $\boldsymbol{A}$ in 1.1 and then find all the non-negative solutions to the resulting set of equations for $\boldsymbol{y}$. The non-negativity constraint follows from the fact that there is no such thing as negative traffic flows. The inversion is the reason that problems of this type are known as linear inverse problems.

However, typically we will find that solving for $\boldsymbol{y}$ does not deliver a unique answer. The system of linear equations after inverting $\boldsymbol{A}$ in 1.1 is typically *underdetermined*, because there are more unknowns than linear equations.

In other words, traffic networks tend to contain more routes than links, thus the number of model parameters exceed the total number of data points and identifiability problems arise. After introducing different traffic assignment models from equilibrium models through to dynamic assignment, we will then discuss various methods used to make inference whilst dealing with the identifiability problem mentioned above.

1.2 Traffic assignment models

In *traffic assignment* we assume the number of trips between any OD pair is known, that is, we know the origin-destination matrix, and it is of interest to learn in what way the users are distributed among the possible routes associated with a given OD pair.

1.2.1 Equilibrium models

The notion of equilibrium in the analysis of urban transportation networks stems from the dependence of the traffic flows on the level of demand in the system. Since congestion increases with flow and trips are discouraged by congestion, there is an interaction that can be modelled as a process of reaching an equilibrium between congestion and travel decisions. Initially, academic research on traffic assignment models was focused on these equilibrium models, with two approaches being particularly common, the User-Equilibrium (UE) model and the Stochastic-User-Equilibrium (SUE) model.

User-Equilibrium model. This method is based on Wardrop's First Principle stating that 'no driver can unilaterally reduce travel costs by shifting to another route', see Wardrop (1952). It is reasonable to assume that every motorist will try to minimise the travel time when travelling from origin to destination. This does not mean that all travellers can be assigned to a single path. The travel time on each link changes with the flow and therefore, the travel time on several of the network paths changes as the link flows change. An equilibrium is reached when all users have minimised their travel costs and have no further incentive to change their route choices.

The UE model makes strong assumptions. It uses the idea that motorists have full information, that is, they know the travel time on every possible route, and that they consistently make the correct route choice decisions. It is also inherently implied that all individuals are identical in their behaviour, with the flows considered to be stable over the period of analysis. This can be true to a certain extent if e.g. observations are always made around the same period of time of the day, say during 'morning peak', but this model still does not allow for any variation between users or across time.

Stochastic-User-Equilibrium model. An extension of the UE model, first introduced by Daganzo and Sheffi (1977), which is defined as the state where 'no traveller can improve his or her *perceived travel time* by unilaterally changing routes'. The perceived travel time can then be looked upon as a random variable distributed across the population of drivers. In other words, each motorist may perceive a different travel time, over the same link.

The 'stochastic' element introduced are the random factors which are influencing choices given that now the users do not have complete knowledge of the travel conditions. Instead they are making 'educated guesses' (Sheffi, 1985). Equilibrium will be reached when no traveller believes that his travel time can be improved by unilaterally changing routes.

If the travel times were to be entirely accurate and all motorists could perceive the same travel time then the Stochastic-User-Equilibrium would become identical to the User-Equilibrium.

1.2.2 Deterministic/Stochastic

The mention of 'stochastic' in SUE is misleading here as both the conventional traffic assignment models UE and SUE described in the previous section are deterministic.

A deterministic model is one which performs the same way for a given set of initial conditions, that is for the same OD matrix parameters and other input, the algorithm generates identical equilibria each time. In other words, a deterministic model assumes that given some initial conditions there is a single outcome that, in principle at least, may be precisely determined. With this approach, it is possible to directly relate the difference in observed results to differences in the behavioural parameters or demand levels entered, because all other parameters can be kept fixed.

Consider for example a network with a single OD pair, where w is the overall demand in the network and ϱ_r, $r = 1, \ldots, N$, the probability of using the rth route serving the one OD pair given the congestion on that route, $\boldsymbol{c}$.

In a deterministic approach we model the route flows as $w\varrho_r$, $r = 1, \ldots, N$. Each route choice probability ϱ_r is a fixed value here, since we deduce the same route flows for any given time from the link counts initially observed, thus we deduce the same probability of users choosing to use route r. The lack of any variability in the parameter estimates means that the route choice probability has the same meaning as a route choice proportion here.

From a transport planning point of view a deterministic model will only be useful for making accurate predictions into the future when haphazard variation is not an important aspect of the network system's behaviour. Otherwise, if we want to have an idea of the variability in traffic flows as well as the degree of uncertainty in the estimated average flows then stochastic models are more useful (Watling, 2002a,b).

To return to our example, we can imagine modifying the model so that one or both of the components of the above mentioned example are stochastic in nature. By way of illustration, we can model the route flows as a Multinomial distributed random variable with parameters w and $\boldsymbol{\varrho}$. A difficulty with this approach is that congestion $\boldsymbol{c}$ is a function of the link counts, and the link flows depend in part on the route choices made. Thus, we have a circular relationship, where the probability of using a given route, ϱ_r, is functionally related to whether or not the rth route is used. This issue can be circumvented by introducing a time dimension, as described in Hazelton (1998). As another example, Watling (2002a,b) provides a stochastic version of the above mentioned deterministic SUE by modelling traffic flows as random variables rather than fixed quantities.

In a landmark paper by Cascetta (1989), the idea of a temporal stochastic model is described in which the route flows are viewed as a Markov chain, that is, a system of transitions from one state to another between a finite amount of states in total. In a transport modelling setting, a state refers to the route a traveller chooses to use on a given day, where route choices on day t depend on costs experienced in previous periods. The author derives the idea from the fact that systems evolve over time, that is, flows observed at one point in time will not necessarily repeat themselves, hence he considers the successive states themselves as a process. Each time the simulation is run it will have a slightly different outcome, even if the inputs are the same. So, in contrast to deterministic systems, for a given initial condition there are many possible future trajectories, and we now estimate the probabilities with which these different trajectories may occur.

For simulating route flows in the long run, the primary advantage of a stochastic model is that such a model characterises the unpredictable nature of the variables more realistically, reflecting possible variations in demand on a day-to-day basis, for example. It also helps to ensure that observed results are not unduly biased by the particular set of observed link flows, as the resulting density estimates do not entirely depend on the initial starting points, as is the case with deterministic models. In other words, a stochastic simulation model is more appropriate for any real physical system whose behaviour has an intrinsically random component.

In other words, with stochastic simulation models, instead of the algorithm converging to a single stationary value, the estimate is a stationary probability distribution. This distribution can be unique and is independent of elapsed time and starting configurations in its history.

Cantarella and Cascetta (1995) as well as Davis and Nihan (1993) explore the relationship between deterministic and stochastic models. The two types of models can be linked through the fact that under the limiting process where the number of vehicles using a system is allowed to become infinitely large the stochastic process can be approximated by deterministic models.

1.2.3 Static/Dynamic

All equilibrium models, including the SUE model, are by nature static (Cascetta, 1989; Hazelton, 1998), that is, these models do not account for time as a factor. Although the theory for these set of models is well-understood, their ability to make accurate predictions is severely limited as transport systems do not necessarily remain in the same equilibrium state over time. Watling and Van Vuren (1993) in particular identify three important time-dependent effects that cannot be represented by a static model. The first being changes over time in route choice preferences, the second is spatial-temporal movements and the third is traveller's learning processes.

In contrast to static models, dynamic process models incorporate the idea that route choices made by travellers on a given day depend on previous travel experiences. In particular, fluctuations in the network system can be represented as a Markov chain (Cascetta, 1989). These kind of models deliver the advantage of allowing for interactions between traveller's experiences with events such as congestion and traveller's route choices (Hazelton and Watling, 2004).

There are essentially two types of dynamic models, the first represent within-day dynamics and the second allow for day-to-day changes to be modelled. Models which incorporate *within-day dynamical* aspects allow for variations in travel patterns due to traffic control systems and time of departure. *Between-day dynamical* or *day-to-day dynamical* models focus on fluctuations in traffic levels due to factors such as the type of day (weekend versus weekday for example) or learning patterns in response to changes in the network. Hazelton (2002) devises a 'coefficient of reactivity' summarising to what extent day-to-day variation influences user's reactions to changing levels of congestion, or other events, in contrast to variability due to underlying fluctuations.

The two dynamic models are closely related, where to a certain degree, the within-day dynamic models can be considered special cases of the between-day dynamic models. Cascetta and Cantarella (1991) propose an extension of the method put forward by Cascetta (1989) for day-to-day models to the case of within-day dynamic demand, and later on, Cantarella and Cascetta (1995) deliver a unifying framework that applies to both cases.

Furthermore, Cantarella and Cascetta (1995) introduce the notion that there is a clear link between dynamical Markov model systems and static equilibrium systems. The relationship is investigated in Watling and Hazelton (2003), where the authors find that the Markov dynamical models converge asymptotically to towards the SUE as demand increases in the network system.

1.3 Statistical methods of OD matrix estimation

This section gives an overview of the methods developed to find the relationship between traffic counts collected at discrete points in time and space, and the travel demand that has most likely generated this data within the framework of OD matrix estimation.

The inferential problems we consider in this thesis can relate to a variety of model parameters. However, OD matrix estimation is especially interesting both because of its importance to traffic modelling, and due to the very direct manner in which it relates to the linear inverse identifiability problem.

The described methods relate to OD estimation in the static context, where the counts are measured at one consistent time period each day and assume the rate of travel remains constant from observational window to observational window. Reference is made to car journeys but these methods apply to other transport modes analogously.

1.3.1 Maximum Entropy methods

One of the earlier approaches to estimating an OD matrix which reproduces the observed traffic counts was to use the minimum information (entropy maximising) model (Zuylen and Willumsen, 1980).

For simplicity, the method only applies to proportional assignment models in which the network is assumed to be uncongested, leading to an independence between traffic demand and which links users choose to take. The authors apply the maximum entropy method to a small artificial network with 15 nodes and 36 two-way links and found that it performed satisfactorily well.

The two techniques presented by Zuylen and Willumsen (1980), namely minimal information and maximum entropy respectively, attempt to find an OD matrix which is as similar as possible to an *a priori* or target matrix, while remaining consistent with the observed link counts. An example of such a target matrix could be an estimated OD matrix obtained from a previous survey. The two methods differ solely by the weights given to the counts for each link, that is they are formally the same thing.

Zuylen and Willumsen (1980) briefly discuss the problem that the link flows can appear inconsistent when errors in the link counts are accounted for in the model. The authors recommend the use of a maximum likelihood method devised by Hamerslag and Huisman (1978) to remove the inconsistencies.

Another problem with the maximum entropy approach is that when estimating the average route flows it uses only the *a priori* matrix as a target. Actually, maximum entropy methods are equivalent to assuming a uniform prior matrix in a Bayesian context. That is, the link counts are in a sense the only data considered, which may considerably weaken the reliability of the resulting estimates if the target matrix is also a dependable source of information.

When viewed together with the fact that this method cannot be applied to any congested network, it appears to be a method that is mainly used for convenience rather than having sound statistical properties and being applicable to real-life networks.

1.3.2 Generalised least squares estimation

A technique with sound statistical foundations is generalised least squares (GLS) estimation which, as the names suggests, involves minimising quadratic equations.

The aim of GLS in transport planning is to reproduce the observed route flows by finding an estimated OD matrix that minimises the distance to a target matrix as well as being compatible with the observed link counts. An example of a target matrix, also known as an *a priori* matrix may be an outdated OD matrix obtained from a previous survey. As Cascetta (1984) emphasises, the straightforward manner in which this kind of additional information can be integrated into the current analysis is one of the main attractions of this form of estimation.

The *a priori* data, that is, the prior estimates of the mean route flows are assumed to follow a Multivariate Normal (MVN) distribution with mean vector $\check{\boldsymbol{\lambda}}$ and dispersion matrix $\check{\Lambda}$, a diagonal matrix with entries equal to the elements of $\check{\boldsymbol{\lambda}}$. The link counts $\boldsymbol{x}$ are also considered to be MVN distributed, with mean $\boldsymbol{A\lambda}$ and variance-covariance $\boldsymbol{A}\Sigma\boldsymbol{A}^{\mathsf{T}}$, where Σ is the variance-covariance matrix of the route flows.

Overall, a function of the route flows can be defined as a combination of the two sources of information as seen in Equation 1.2. The GLS estimates are found by minimising the function $f(\boldsymbol{\lambda})$ in 1.2 with respect to the elements of $\boldsymbol{\lambda}$:

$$f(\boldsymbol{\lambda}) = \underbrace{\eta(\boldsymbol{\lambda} - \check{\boldsymbol{\lambda}})^{\mathsf{T}}\check{\Lambda}^{-1}(\boldsymbol{\lambda} - \check{\boldsymbol{\lambda}})}_{\text{prior information}} + \underbrace{(\boldsymbol{x} - \boldsymbol{A}\boldsymbol{\lambda})^{\mathsf{T}}(\boldsymbol{A}\Sigma\boldsymbol{A}^{\mathsf{T}})^{-1}(\boldsymbol{x} - \boldsymbol{A}\boldsymbol{\lambda})}_{\text{link flow model}} \tag{1.2}$$

where $\boldsymbol{A}$ is the routing matrix and η is a weighting parameter which controls the influence of the *a priori* information on the optimisation process. That is, η allows us take into account the reliability of these two data sources.

In Bell (1991)'s work, the parameter $\boldsymbol{\lambda}$ is subjected to non-negativity constraints in the minimisation algorithm as a way to address the paradox of negative traffic flows being possible. Cascetta (1984) on the other hand focuses on examining the properties of statistics that are defined explicitly in terms of the estimated mean route flows, the observed route flows and their dispersion matrices, hence did not impose any constraints. An important finding was that in the unconstrained case, the mean route flows estimated via GLS are the best linear unbiased estimators (BLUE).

Although this finding is crucial to our understanding of GLS estimates, in order for traffic model estimates to be realistic, the non-negativity constraint is essential and when these constraints are applied GLS estimates become biased, as discussed in chapter 3 (Parry and Hazelton, 2012). In addition, Cascetta (1984) makes two very strong assumptions, the first is that the starting estimates from the target matrix are unbiased and the second is that the errors of the assignment model are negligible, that is the model is correctly specified. Whether or not these two important assumptions hold can be easily checked via a comparison of estimator performance under criteria of risk and generalised minimum square error (MSE). Nevertheless, these are two very demanding criteria with a high chance of at least one being violated.

The GLS estimation is found to perform very similarly to the previously described maximum entropy approach in the event that the link counts were collected accurately (Bell, 1991), an assumption we use in all our traffic assignment models. In general, Cascetta (1984) finds that GLS estimators consistently have a lower MSE than the maximum entropy estimator. However, the GLS method relies heavily on an informative target matrix being available and the accuracy of the obtained estimates are highly dependent upon the quality of the *a priori* information (Hazelton, 2001b). In particular, Bell (1991) finds that unless the prior information is fairly error-free, the GLS suffers from severe under- or overestimation of large or low flows, respectively.

The poor estimation of flows in the presence of error-ridden priors is particularly strong in Bell (1991)'s work because the data considered were only one day's worth of observations and a lack of data leads to an inflation in the impact of the prior.

One important point needs to be clarified here, namely that OD matrix parameter estimation is often confused with OD matrix reconstruction. For instance, Bell (1991) is inconsistent in his differentiation between the expected route flows, $\boldsymbol{\lambda} = \mathsf{E}(\boldsymbol{y})$, and the realised route flows, $\boldsymbol{y}$. One reason for this lies in the fact that the two methods are thought to lead to equal results, however, Hazelton (2001b) presents examples which illustrate how much the outcomes can differ.

1.3.3 Maximum likelihood estimation

The general likelihood function for a random sample of counts is $\mathsf{L}(\boldsymbol{\lambda}) = \prod f(\boldsymbol{x}|\boldsymbol{\lambda})$. Maximum likelihood (ML) estimation involves finding the model parameters that are most likely given the data observed, that is, the parameters that maximise the likelihood function.

The expression for the likelihood function as well as its calculation is simple if the level of demand in the network is sufficiently large, that is, all mean route flows are high (Cao et al., 2000; Vardi, 1996). We can then use a normal approximation where the probability density function of $f(\boldsymbol{x}|\boldsymbol{\lambda})$ for a random sample of counts collected on M links follows a multivariate normal distribution with mean $\boldsymbol{A\lambda}$ and variance-covariance $\boldsymbol{A}\Lambda\boldsymbol{A}^{\mathsf{T}}$, as seen in Equation 1.3:

$$f(\boldsymbol{x}|\boldsymbol{\lambda}) = (2\pi)^{-\frac{M}{2}} \left|\boldsymbol{A}\Lambda\boldsymbol{A}^{\mathsf{T}}\right|^{-\frac{1}{2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{A\lambda})^{\mathsf{T}}(\boldsymbol{A}\Lambda\boldsymbol{A}^{\mathsf{T}})^{-1}(\boldsymbol{x}-\boldsymbol{A\lambda})}. \tag{1.3}$$

Note that the dispersion matrix Λ for the route flows is a diagonal matrix with the mean vector $\boldsymbol{\lambda}$ as its elements. The functional relationship between the mean and the variance enables second-order properties of the link counts, such as the covariances between link counts, to be incorporated into the mean route flow estimation. A covariance occurs because links which consistently share very similar numbers of vehicles are likely to actually have the same vehicles on them (Hazelton, 2003).

This connection is beneficial in countering the identifiability problem when determining the underlying mean route flows. It provides additional information that can be used to identify the realised route flows in situations where the actual route flows alone would otherwise be ambiguous (Cao et al., 2000; Hazelton, 2003).

The likelihood takes on a significantly simpler form when the link between the mean and the variance is severed, which can be achieved in two ways.

The first is to replace the dispersion matrix of the route flows Λ with dispersion matrix Σ whose diagonal elements are fixed values instead of elements of the mean route flow vector, $\boldsymbol{\lambda}$.

The second approach is to substitute the dispersion matrix of the link counts $\boldsymbol{A}\Lambda\boldsymbol{A}^{\mathsf{T}}$ in Equation 1.3 by the sample variance-covariance matrix of the link counts $\boldsymbol{S}$ so that the dispersions do not depend on the parameter vector $\boldsymbol{\lambda}$. The main advantage of simplifying the likelihood is the improvement in the computational efficiency in the estimation process.

The GLS estimation method is strongly associated with the ML estimation procedure (Nakayama et al., 2009), provided the likelihood is in quadratic form. While the likelihood in Equation 1.3 is not in a quadratic form because the variance-covariance matrix is a function of λ, both the simplified likelihoods are. In particular, Hazelton (2000) shows that ML estimation on the log-likelihood version of 1.3 with $\boldsymbol{S}$ instead of $\boldsymbol{A}\Lambda\boldsymbol{A}^{\mathsf{T}}$ is a special case of GLS estimation where the target matrix is considered completely redundant (η=0).

This second approach to simplifying the likelihood is problematic if the link counts are not reliable, for example because there is not a great quantity available or the counts contain a lot of measurement error. As the dispersion matrix is a function of the link counts only, inaccuracies in the counts will lead to an erroneous matrix $\boldsymbol{S}$. This leads to the potential of misleading ML estimates as a consequence. The ML estimator does not suffer from this issue if the variance-covariance matrix of the route flows depends on the parameter λ.

Another major drawback of a fixed dispersion matrix for the route flows, as illustrated in Hazelton (2000), is that only first order properties are considered. That is, GLS estimation (or ML estimation on the simplified likelihoods) focuses exclusively on reproducing the mean traffic flows. ML estimation methods on the other hand are capable of considering second order properties as well through the functional relationship of the variance and the mean in the matrix Λ.

The popularity of the ML estimator lies in its sound statistical properties, such as asymptotic unbiasedness, normality and efficiency. However one must bear in mind these are asymptotic properties, which means they are only valid for networks with a high traffic demand. Furthermore, efficiency is only given if the model assumptions are valid, which implies careful validation of any beliefs held (Watling, 2002a).

The form of the likelihood and the ML estimation procedure are considerably less straightforward when dealing with networks that do not have high levels of demand. In these cases we are dealing with discrete distributions, typically Poisson.

In the case of discrete route flow models, the evaluation of the likelihood $\mathsf{L}(\boldsymbol{\lambda})$ involves summing over all terms in the set of feasible route flows, defined as

$$\mathcal{Y}(\boldsymbol{x}) = \{\boldsymbol{y} : \boldsymbol{x} = \boldsymbol{A}\boldsymbol{y}, \boldsymbol{y} \geq 0\}, \tag{1.4}$$

that is, all route flows consistent with the observed link counts according to the system of linear equations 1.1. The set $\mathcal{Y}(\boldsymbol{x})$ is typically of substantial size which makes summation over all the elements problematic. This can become a computational black hole even for moderately sized networks. Another point at issue is that even when calculating a numerical approximation the need to enumerate possible elements in the set of feasible route flows typically presents itself as a challenging computational task for larger networks (Hazelton, 2000; Watling, 2002a,b).

1.3.4 Bayesian approach

In a Bayesian statistical analysis there are three components, the prior, the likelihood and the posterior. The posterior of the most likely mean route flows conditional on the observed link counts is calculated up to a normalising constant by multiplying the likelihood with the prior.

A controversial aspect of the Bayesian paradigm is the prior, a probability distribution placed on the parameters of interest. The matter of contention is the degree of relative weight assigned to prior beliefs in the model. The prior can be a reflection of previously known facts about these parameters. On the one hand one could incorporate information about differing levels of demand on the routes in the network or, on the other hand design the prior so it has no effect on the shape of the posterior.

As Hazelton (2010a) points out, the Bayesian approach has a number of advantages over other methodologies. For example, it can handle inference problems when dealing with latent variables, in this case the route flows, in a competent manner. It also has the ability to regularise the indeterminacy problems through the use of the prior, even if the prior is not very influential.

A major contribution to the introduction of Bayesian techniques for modelling traffic networks was made by Maher (1983), who develops a iterative algorithm which he tests on a small network with six OD pairs and four links. In this paper, the ratio of the amount of variability contributed by the data over the variance of the prior distribution is used as a measure of the influence through the prior, and if the lowest possible degree of belief in the prior is given, it can be shown that the estimation becomes equivalent to the entropy maximising approach.

However, arguably Maher (1983) makes two contradictory assumptions. On the one hand, the author uses a proportional assignment model which implies an uncongested network, and on the other hand models the traffic flow counts as approximately normally distributed, which requires high levels of demand. These two conjectures are inconsistent, as networks that have high levels of usage tend to develop problems with congestion.

An important extension of this initial work is the paper from Tebaldi and West (1998) who, following up on Vardi (1996)'s formulation of the OD estimation problem, stress the importance of using Bayesian analysis to avoid ambiguities in the route flow estimates inherent to other statistical methods. These ambiguities, which stem from the linear inverse structure, can lead to a severe over- or underestimation of route flows.

Tebaldi and West (1998) use an example of a real network, the Monroe NC transportation network with 13 nodes and 12 two-way links to illustrate how the Bayesian estimation improves significantly as the prior is given more influence on the posterior, provided that an informative prior is available. In a comment written in response to this paper, Vardi points out that not only do the authors use an informative prior, they also examine data which has been collected over a single time period. In order to address the issue raised in the paper of needing an influential prior, Vardi recommends collecting data repeatedly over several observational periods. The heightened amount of data would prevent the excessive influence of the prior as described by the authors.

While developing a method to combine a traffic assignment model with a Bayesian network model in order to make predictions, Castillo et al. (2008b) mention a vital advantage of the Bayesian approach, namely that estimates can be updated in an elegant manner when new data becomes available. The reason is that a Bayesian analysis inherently never returns a final result; in the sense that a posterior can become the future prior which when multiplied with the likelihood forms a future posterior and so on and so forth.

In addition, as Hazelton (2008) points out, the Bayesian framework also allows for the prospect of being able to calculate confidence and prediction intervals as measures of precision for the obtained estimates, because the output of the Bayesian analysis are distributions. The main difficulty with this approach is that the process of sampling from the posterior distribution makes this method computationally expensive, hence finding efficient algorithms needs much attention. To deal with this problem, Hazelton (2001b) considers applying a multivariate approximation to the link counts in systems with high demand, which renders the inference considerably less computationally demanding.

Another suggestion put forward by Li (2005) is the use of an expectation- maximisation (EM) algorithm to overcome the computing difficulties, however, this approach demands a very informative prior.

1.4 Structure of thesis

At the moment there are two significant gaps in the statistical methodology of traffic flow estimation. The first is developing methods capable of considering as many characteristics of the traffic network system as possible, while providing accurate estimation in the presence of higher numbers of parameters. The second is deriving techniques which are computationally efficient, in order for them to apply to real networks, which are larger and more complex than the systems so far considered.

In chapter 2 we consider networks with low levels of demand, where a normal approximation is not suitable, and we need to assume discrete models such as the Poisson distribution when modelling the route flows. The likelihood for a discrete counts model is however mathematically intractable, instead probability distribution sampling methods are needed to evaluate it.

We will discuss the challenges we encounter with this approach and develop an extension of an improved Markov Chain Monte Carlo sampling technique for estimation of traffic flows for linear networks to systems with a tree-like structure. We supply a detailed description of the algorithm and a combinatorial simulation study that draws comparisons to the generalised least squares approach. Programming code too detailed for the main text is given in the appendix.

In chapter 3, we adapt this new methodology to apply to cases where the data, instead of being observed at intermittent time points, is measured at successive observational periods and the evolution of the traffic flows over time becomes the process of interest. The day-to-day dynamic models we develop in this chapter are more effective at capturing features such as travellers' learning experiences over time as well as other habitual attributes that contribute to route choice decision making.

In chapter 4 we turn our attention towards intelligent transport systems where the movements of vehicles can be tracked via emerging technologies such as electronic toll tagging, registration plate scanning and mobile traffic sensors. This is a subject that has drawn attention in recent years (Castillo et al., 2008a; Watling, 1994), as it allows for the development of models whose assumptions are not as strong as those discussed in the previous two chapters.

We present an initial investigation of making inference when using other sources aside from link counts, although we note that the classical link count data only models still have the greatest practical applicability at this point (Hazelton, 2008).

In chapter 5 we move away from the focus on obtaining good estimates of model parameters and move towards another equally important aspect of transport research, namely the development of models that are capable of representing as many features of traffic systems as possible. We address the problem of statistical modelling when the observed link flows exhibit unusual features such as changes in the mean and the variance. We investigate the broadening of current statistical traffic models that are capable of capturing these non-stationary features.

Finally, in chapter 6 we summarise all the results developed in this thesis and suggest future avenues of research.

Chapter 2

Improved MCMC sampling for tree networks

Now everybody's sampling.

(Missy Elliot)

2.1 Introduction

Occasionally in traffic network modelling, we come across the dilemma that the probability distribution of interest is mathematically intractable. In these cases, the answer is to come up with an approximate solution instead.

Markov Chain Monte Carlo (MCMC) methods, a class of algorithms designed to sample from a probability distribution using constructed Markov chains, have gathered momentum in line with the rate of technological advances in computational power. With faster and more powerful machines, the calculations necessary for MCMC can be dealt with, however computational efficiency still remains the biggest challenge in this area of statistical research.

MCMC samplers can be applied to both Bayesian and frequentist inference problems, but have been more relevant to the former (Besag, 2001). The Bayesian MCMC approach is particularly appropriate for statistical linear inverse problems in traffic network-based modelling, as Bayesian methods are well suited to handle missing data and latent variables (Hazelton, 2010b).

An discussed in chapter 1, a particular problem we face is the OD estimation problem, where we attempt to reproduce the travel demand that would have led to the link counts we observe.

Knowledge about the route flows can then be used to make inference about a variety of other parameters that describe the travel behaviour. We now describe the basic elements - the likelihood, the prior and the posterior - needed to make inference using the Bayesian paradigm.

The likelihood. The likelihood distribution is defined as a probability function of the data $\boldsymbol{x}$, given the values of the model parameters $\boldsymbol{\theta}$. When the data are vehicle counts in a traffic system with low demand we model travel flows using a discrete model. We assume that the parameters in the vector $\boldsymbol{\theta}$ relate to the route flows, $\boldsymbol{y}$, in the sense that $\boldsymbol{\theta}$ is conditionally independent of $\boldsymbol{x}$ given $\boldsymbol{y}$. The form of the likelihood for a discrete flow model is

$$\mathsf{L}(\boldsymbol{\theta}) = f(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{\boldsymbol{y}} f(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta}) f(\boldsymbol{y}|\boldsymbol{\theta}) \;=\; \sum_{\boldsymbol{y}} \mathbb{I}_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} f(\boldsymbol{y}|\boldsymbol{\theta}) \;=\; \sum_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} f(\boldsymbol{y}|\boldsymbol{\theta}) \tag{2.1}$$

where the set $\mathcal{Y}(\boldsymbol{x})$ is defined as the set of feasible route flows $\{\boldsymbol{y} : \boldsymbol{x} = A\boldsymbol{y} \text{ and } \boldsymbol{y} \geq 0\}$; the inequality ensures that all elements of $\mathcal{Y}(\boldsymbol{x})$ are non-negative. The equality in Equation 2.1 follows from $f(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})$ being a indicator function, which takes on values 1 if the route flow belongs to the set $\mathcal{Y}(\boldsymbol{x})$ and 0 otherwise.

For example, if we assume the route flows follow a Poisson distribution with mean $\mathsf{E}(\boldsymbol{y}) = \boldsymbol{\lambda}$, we get the following likelihood for the link flows

$$\mathsf{L}(\boldsymbol{\lambda}) \;=\; \sum_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} f(\boldsymbol{y}|\boldsymbol{\lambda}) \;=\; \sum_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} \prod_{r=1}^{N} \frac{\lambda_r^{y_r} e^{-\lambda}}{y_r!}. \tag{2.2}$$

Alternatively, in cases of high demand, where we observe high numbers of vehicles on all routes in the system, it is possible to use a normal approximation to the Poisson model. In this scenario we assume the route flows to be normal distributed with mean $\boldsymbol{\lambda}$ and covariance Λ, where Λ is a diagonal matrix with the mean vector as its elements. The main advantage of modelling the route flows as continuous is that the sum in Equation 2.1 turns into a mathematically tractable integral and the likelihood becomes

$$\mathsf{L}(\boldsymbol{\lambda}) = \int_{-\infty}^{\infty} f(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\lambda}) f(\boldsymbol{y}|\boldsymbol{\lambda}) d\boldsymbol{y} = \Phi_{\boldsymbol{A}\Lambda\boldsymbol{A}^{\mathsf{T}}}(\boldsymbol{x} - \boldsymbol{A}\boldsymbol{y}) \tag{2.3}$$

where $\Phi_{\boldsymbol{A}\Lambda\boldsymbol{A}^{\mathsf{T}}}$ denotes a multivariate normal density with zero mean vector and variance-covariance matrix $\boldsymbol{A}\Lambda\boldsymbol{A}^{\mathsf{T}}$. Without the summation over the route flows in Equation 2.1 we can evaluate the distribution of the flows in a closed form.

However, setting the mean equal to the variance can be potentially invalid, as real-life data is often found to be overdispersed (Hazelton, 2001a, 2008). In this case we can apply a more flexible model, where the distribution of $\boldsymbol{y}$ is $\mathsf{Norm}(\boldsymbol{\lambda}, \upsilon\Lambda)$ where υ is a scalar less than 1 in the case of underdispersion and greater than 1 when representing overdispersion in the route flows.

The prior. Before collecting the data, we may have information about what drives the observed traffic patterns or have access to facts about the flows on each of the routes. We can incorporate this *a priori* knowledge about the values of the canonical parameters, $\boldsymbol{\theta}$, into the analysis via the prior probability function.

If we have no information about the parameters of interest, we choose a prior which is described as *uninformative*. A more suitable word would be *vague*, which more accurately reflects the fact that the prior has little impact on the likelihood (Gelman et al., 2004).

To return to our example of the Poisson model, where the parameters of interest are the average route flows, $\boldsymbol{\lambda}$, we can model our beliefs via a density which is defined on an interval, because $\boldsymbol{\lambda}$ is a vector of continuous parameters. We may not have any knowledge about whether certain routes are more frequented than others, in which case a vague prior is appropriate. A suitable prior in this case would be the uniform distribution,

$$f(\boldsymbol{\lambda}) = \frac{1}{U^N}\mathbb{I}_{[0,U]^N}, \tag{2.4}$$

with lower bound equal to 0 and an arbitrary upper bound U. The upper bound does not usually need to be specified in practice as long as it is suitably large. The superscript N indicates the fact that the priors for each component of $\boldsymbol{\lambda}$ multiple together. This is a vague prior, because the density gives all possible flow rates equal probability and has little influence on the posterior.

The posterior. Overall, we make inference for the parameter vector $\boldsymbol{\theta}$ in a Bayesian analysis by summarising the posterior distribution, for example we can use the mode or mean as a point parameter estimate. The posterior probability for the parameter $\boldsymbol{\theta}$ is obtained, up to a constant, by multiplying the prior by the likelihood

$$f(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{f(\boldsymbol{\theta})f(\boldsymbol{x}|\boldsymbol{\theta})}{f(\boldsymbol{x})} = \frac{f(\boldsymbol{\theta})f(\boldsymbol{x}|\boldsymbol{\theta})}{\int f(\boldsymbol{\theta})f(\boldsymbol{x}|\boldsymbol{\theta})d\boldsymbol{\theta}}, \tag{2.5}$$

where $f(\boldsymbol{x})$ is the marginal distribution for $\boldsymbol{x}$, which acts as the normalisation constant.

A particular version of the posterior for our example, with Poisson distributed route flows and a uniform prior is

$$
\begin{aligned}
f(\boldsymbol{\lambda}|\boldsymbol{x}) = \frac{f(\boldsymbol{\lambda})f(\boldsymbol{x}|\boldsymbol{\lambda})}{f(\boldsymbol{x})} &= \frac{\frac{1}{U^N}\mathbb{I}_{[0,U]^N} \cdot \sum_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} \prod_{r=1}^{N} \left(\frac{\lambda_r^{y_r} e^{-\lambda_r}}{y_r!} \right)}{\int_{[0,U]^N} \frac{1}{U^N}\mathbb{I}_{[0,U]^N} \cdot \sum_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} \prod_{r=1}^{N} \left(\frac{\lambda_r^{y_r} e^{-\lambda_r}}{y_r!} \right) d\boldsymbol{\lambda}} \\
&= \frac{\sum_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} \prod_{r=1}^{N} \left(\frac{\lambda_r^{y_r} e^{-\lambda_r}}{y_r!} \right)}{\int_{[0,U]^N} \sum_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} \prod_{r=1}^{N} \left(\frac{\lambda_r^{y_r} e^{-\lambda_r}}{y_r!} \right) d\boldsymbol{\lambda}},
\end{aligned}
\tag{2.6}
$$

where the cancellation of the uniform density demonstrates that if the prior has little influence on the overall shape of the resulting distribution we get a posterior that is proportional to the likelihood.

Obtaining the exact posterior in Equation 2.5 is often problematic for two reasons. First, the evaluation of the likelihood function requires enumeration of all feasible flows in order to evaluate the sum in Equation 2.1 (Vardi, 1996). This is problematic as the number of elements in the set of feasible flows, $\mathcal{Y}(\boldsymbol{x})$, can become extremely large, even for moderate sized network systems. Thus enumeration is computationally challenging for anything but very small networks.

Second, the integral in the denominator of Equation 2.5 involves computing the normalisation constant $f(\boldsymbol{x})$. This will generally not be possible in closed form because this requires integration of a high-dimensional function.

A notable solution to the first problem is given by Tebaldi and West (1998), who develop a sampling methodology that can be implemented without specifying the full set of feasible route flows. However it cannot be applied to realistically sized networks due to computational inefficiency. In this chapter, we develop an alternative technique that is applicable to tree networks, which efficiently samples from the set $\mathcal{Y}(\boldsymbol{x})$. It is founded on some recent work of Li (2008), who presents a Bayesian approach to OD matrix estimation in a transit network, where the data consists of the number of passengers boarding and leaving a bus at each stop. Li (2008)'s idea of modelling the changes in passenger numbers from stop to stop as a Markov process is picked up by Hazelton (2010b) and incorporated into a two-stage MCMC sampling algorithm. Our work is an extension of this line of work from transit networks to tree networks.

The use of Markov Chain Monte Carlo methods addresses both of the problems mentioned above. Firstly, a MCMC algorithm provides a sampling based approximation to the posterior which requires only that we sample from the set of feasible route flows rather than enumerate the entire set. Secondly, the need to calculate the normalisation constant in the denominator of Equation 2.5 can be circumvented by summarising the complex non-standard distribution $f(\boldsymbol{\theta}|\boldsymbol{x})$ via iterative techniques such as MCMC samplers.

In the next sections we will first discuss Markov Chain Monte Carlo techniques in more detail. We will then describe the two-stage sampling algorithm first introduced by Hazelton (2010b) and finally talk through the improved sampler for tree networks.

2.2 Markov Chain Monte Carlo sampling

Back in 1906, Markov first introduced the concept of Markov chains when showing that independence was not essential for the law of large numbers, which he illustrated via an example of the alternation of consonants and vowels in Pushkin's *Eugene Onegin*.

The physicists Metropolis, Ulam and von Neumann initiated the emergence of Monte Carlo methods in Los Alamos, New Mexico during the Second World War while working on the neutron diffusion needed to create nuclear weapons; the name 'Monte Carlo' was the code name for the then confidential research.

Then, in 1953 Metropolis and Ulam published a paper which can be seen to first describe a MCMC sampler, the Metropolis algorithm Metropolis et al. (1953). The most widely used MCMC sampler, the Gibbs sampler, was first introduced in the field of image processing by Geman and Geman (1984).

It was not until the early 1990s that the usefulness of MCMC samplers gained recognition in the statistical world, but it has since received considerable impetus and now solutions to problems that appear to be computational nightmares can be solved in a straightforward manner. The emphasis here is that one can assume that 'simulated' solutions are adequate approximations of the 'exact' solutions, see Robert and Casella (2011).

In the name Markov chain Monte Carlo, the Monte Carlo part refers to integration by simulation. The original Monte Carlo approach was a method developed to use random number generation to compute complex integrals of the form $\displaystyle\int_{l^*}^{u^*} h(x)dx$. The general approach is to decompose the function $h(x)$ into the product of a function $g(x)$ and a probability density function $f(x)$ defined over the interval (l^*, u^*).

Then the integral can be expressed as an expectation of $g(x)$ over the density $f(x)$:

$$\int_{l^*}^{u^*} h(x)dx = \int_{l^*}^{u^*} g(x)f(x)dx = \mathsf{E}_{f(x)}\left(g(x)\right). \tag{2.7}$$

The next step is to draw a finite sample of R values, $x_1, ..., x_R$, from the density function $f(x)$ and invoke the Weak Law of Large Numbers to give

$$\mathsf{E}_{f(x)}\left(g(x)\right) = \operatorname*{plim}_{R\to\infty} \frac{1}{R}\sum_{a=1}^{R} g(x_a) \backsimeq \frac{1}{R}\sum_{a=1}^{R} g(x_a). \tag{2.8}$$

In a Monte Carlo simulation, the draws are assumed to be independent of each other, whilst in a MCMC sampler, the previously drawn sample value is used to randomly generate the next sample value, and a sequence of draws generates a Markov chain.

A Markov chain is a stochastic process where in each step the system may change from the current state to another, or remain the same, according to a certain probability distribution which is functionally dependent on the most recent state.

Let Z^t denote the value of a random variable at time t and let the *state space* refer to the range of possible values of Z. The changes of state are called *transitions* and the probabilities associated with various state-changes are called *transition probabilities*. The random variable Z becomes a *d-order* Markov process when the transition probabilities between different states in the state space functionally depend only on the last d random variable's states, that is

$$P(Z^{t+1}|Z^1, ..., Z^{t-1}, Z^t) = P(Z^{t+1}|Z^{t-d+1}, ..., Z^t), \tag{2.9}$$

where commonly we assume d=1, in which case we would be dealing with a *first-order* Markov process

$$P(Z^{t+1}|Z^1, ..., Z^{t-1}, Z^t) = P(Z^{t+1}|Z^t). \tag{2.10}$$

In the case of a first-order Markov process, the only information about the past needed to predict the future is the current state of the random variable, any further knowledge of other previous values of Z do not have any impact on the transition probability.

A Markov chain is formed when a sequence of random variables $Z^0, ..., Z^n$ are generated through a Markov process. A particular chain is defined by specifying a starting vector, $P(Z^0 = i^*)$, of the state space probabilities at time 0, that is the probabilities that the chain begins in a given state i^*, one of the possible values of Z .

Often all the elements of $P(Z^0 = i^*)$ are 0 except for a single element, corresponding to the process starting in that particular state. As the chain progresses, the probability values get spread out over the state space.

The marginal probability of the chain being in state i^* at time $t{+}1$ is given by the Chapman-Kolomogrov equation, which sums over the probability of currently being in a particular state and the transition probability from that state to another:

$$P(Z^{t+1} = i^*) = \sum_{j^*} P(Z^{t+1} = i^* | Z^t = j^*) \cdot P(Z^t = j^*). \tag{2.11}$$

We denote the state probability $P(Z^{t+1} = i^*)$ as ϖ^{t+1} and the NxN-probability transition matrix as $\mathcal{M}$. $\mathcal{M}$ is defined so it's i^*j^*th element is $P(Z^{t+1} = i^* | Z^t = j^*)$, the probability of moving to state i^* from state j^*. Under these settings we can write Equation 2.11 in the compact form

$$\varpi^{t+1} = \varpi^t \mathcal{M}. \tag{2.12}$$

Using this notation, we can show that successive iterations of the chain in 2.11 describe the evolution of the chain. If we continue deconstructing the probability of being in state i^* at time t in the following manner [1]

$$\varpi^{(t)} = \varpi^{(t-1)}\mathcal{M} = (\varpi^{(t-2)}\mathcal{M})\mathcal{M} = \varpi^{(t-2)}\mathcal{M}^2$$

we eventually end up with

$$\varpi^{(t)} = \varpi^{(0)}\mathcal{M}^t. \tag{2.13}$$

Under suitable conditions (e.g. ergodicity), once a Markov chain has evolved long enough, a stationary distribution ϖ^* can be found for which the vector of probabilities of being in any particular given state has become independent of the initial condition. The equation 2.13 then becomes

$$\varpi^* = \varpi^* \mathcal{M}. \tag{2.14}$$

2.2.1 Metropolis-Hastings algorithm

The Metropolis-Hastings (MH) sampler is an extension of the Metropolis algorithm by Hastings (1970), where the stationary distribution of the Markov chain is the posterior distribution.

[1] superscript for time index is in brackets as a distinction from the exponents of $\mathcal{M}$

The first step in a MH sampler is to find some initial values for the parameters of interest, $\boldsymbol{\theta}$. The choice of these values may be completely at random, however, choosing reasonable starting points will improve the sampler performance. This can be achieved by generating initial values from the prior distribution, or selecting values based on frequentist estimates derived from a current data set.

The algorithm begins with these initial values and then specifies a rule for simulating the next value in the sequence, $\boldsymbol{\theta}^{t+1}$, given the current value in the sequence, $\boldsymbol{\theta}^{t}$. In order to implement this rule we need a proposal density q which must be known, so we can assign probabilities to the candidate values. The rule then consists of simulating a candidate value $\boldsymbol{\theta}^{\dagger}$ and the computation of an acceptance probability α that indicates the probability that the candidate value will be accepted as the next value in the sequence.

The candidates sampled from the proposal distribution are accepted if a value drawn from the $\mathsf{Unif}(0,1)$ distribution is less than the acceptance probability α. The probability of accepting a candidate value is

$$\alpha = \min\left(1, \frac{f(\boldsymbol{x}|\boldsymbol{\theta}^{\dagger})f(\boldsymbol{\theta}^{\dagger})q(\boldsymbol{\theta}^{t})}{f(\boldsymbol{x}|\boldsymbol{\theta}^{t})f(\boldsymbol{\theta}^{t})q(\boldsymbol{\theta}^{\dagger})}\right) \tag{2.15}$$

is essentially a ratio of the posterior for the candidate value of $\boldsymbol{\theta}$ and the posterior of the formerly accepted value. The multiplication by the proposal density function values ensures that the stationary distribution matches the requisite posterior.

The proposal density must be symmetric in the original Metropolis algorithm, while the more general MH algorithm allows for a non-symmetric proposal density. The minima function is used to solve the problem that although the ratio is meant to be a probability it can take on values greater than 1.

The main advantage of MCMC sampling becomes apparent if we look at the ratio

$$\frac{f(\boldsymbol{\theta}^{\dagger}|y)q(\boldsymbol{\theta}^{t})}{f(\boldsymbol{\theta}^{t}|y)q(\boldsymbol{\theta}^{\dagger})} = \frac{\dfrac{f(\boldsymbol{x}|\boldsymbol{\theta}^{\dagger})f(\boldsymbol{\theta}^{\dagger})}{\int f(\boldsymbol{x}|\boldsymbol{\theta})f(\boldsymbol{\theta})} \cdot q(\boldsymbol{\theta}^{t})}{\dfrac{f(\boldsymbol{x}|\boldsymbol{\theta}^{t})f(\boldsymbol{\theta}^{t})}{\int f(\boldsymbol{x}|\boldsymbol{\theta})f(\boldsymbol{\theta})} \cdot q(\boldsymbol{\theta}^{\dagger})}. \tag{2.16}$$

The normalisation constant $f(\boldsymbol{x}) = \int f(\boldsymbol{x}|\boldsymbol{\theta})f(\boldsymbol{\theta})$ cancels out in Equation 2.16, leaving the familiar form seen in Equation 2.15 and thus the calculation of this integral is avoided.

If the prior $f(\boldsymbol{\theta})$ is reused as the proposal distribution $q(\boldsymbol{\theta})$ then the acceptance probability α reduces down to the likelihood ratio

$$\min\left(\frac{f(\boldsymbol{x}|\boldsymbol{\theta}^{\dagger})}{f(\boldsymbol{x}|\boldsymbol{\theta}^{t})}, 1\right) \tag{2.17}$$

because the $f(\boldsymbol{\theta})$ and $q(\boldsymbol{\theta})$ terms cancel each other out.

The next step is to update the parameter values of interest. If the candidate value $\boldsymbol{\theta}^{\dagger}$ is accepted, it is redefined as the new parameter value $\boldsymbol{\theta}^{t+1}$. If the candidate value is found to be too unlikely to be coming from the posterior distribution it is rejected and the updated parameter value is reset to the current value $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^{t}$.

We then increment t to t+1 and repeat the steps described above of sampling a new candidate value and applying the acceptance rule. We repeat this process as many times as practically possible, for example R=100,000 times, as the greater the number of iterations, the greater the probability that the Markov chain will have reached convergence. In theory the chain will always converge, however the number of iterations needed to reach convergence depends on a number of factors. The most important factor is the choice of proposal distribution.

In the case of an *independence chain MH algorithm*, we sample from a proposal distribution that is functionally independent of the current parameter value $\boldsymbol{\theta}^{t+1}$. For example, sampling from the prior leads to independent samples.

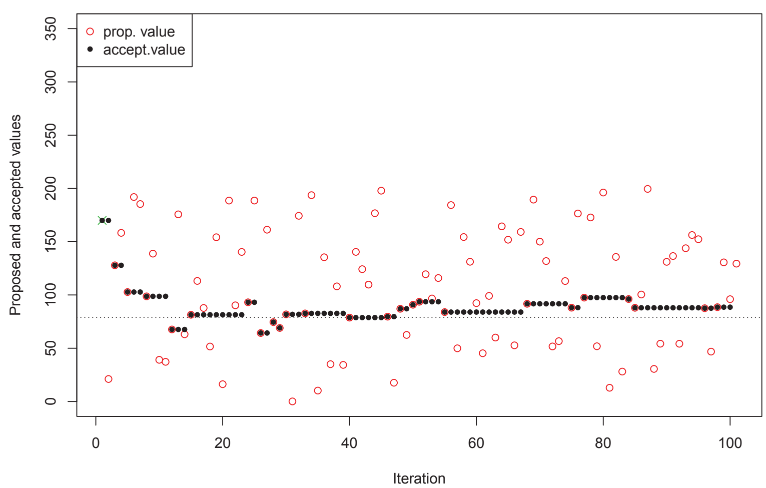


Figure 2.1: Simple independence chain. True parameter value indicated by dotted line and green cross represents initial candidate value.

The simple independence sampler is not very computationally efficient unless the proposal distribution $q(\boldsymbol{\theta})$ already shares certain properties with the posterior distribution $f(\boldsymbol{x}|\boldsymbol{\theta})$ we are trying to sample from, e.g. the same support. If the proposal density approximates the posterior density well you get a high acceptance of candidate values. The independence chain in Figure 2.1 for example has an acceptance rate of less than 20%, which indicates that the proposal density is approximating the posterior only moderately well.

An alternative sampler, the *Random-Walk MH algorithm* is more suitable if little is known about the posterior. The proposal distribution is $q(\theta^{t+1}|\theta^t)$, which in contrast to the independence chain now depends on the previously accepted value. This is noticeable in Figure 2.2 where the proposed values remain in the vicinity of previous candidate values.

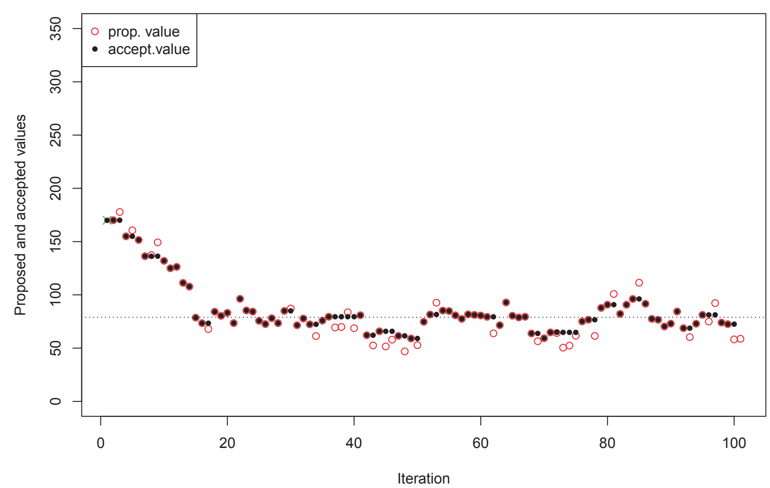


Figure 2.2: Random walk chain. True parameter value indicated by dotted line and green cross represents initial candidate value.

A larger proportion of proposed candidates were accepted for the Random-Walk MH sampler, see Figure 2.2, than for the simple independence case, see Figure 2.1. The considerably higher acceptance rate indicates a significant improvement in efficiency.

Another important factor that can influence the efficiency of a MCMC sampler is how close the initial values are to the actual values of the parameters that we are estimating. If the sampler is not mixing well, that is, the MCMC chain is not moving around in the parameter space enough, then the choice of initialisation can become relevant (Gamerman and Lopes, 2006).

2.2.2 Gibbs algorithm

We can use the Gibbs method to sample from the joint distribution when we know the full conditional distribution of each parameter. The Gibbs sampler can be viewed as a special case of the MH sampler, where the randomly sampled value is always accepted. In other words, we have acceptance probability α=1. The approach to constructing a Markov chain in this case is by considering only univariate conditional distributions where all the random variables of interest but one have been fixed.

For example, consider a joint distribution $f(x, y)$ where y is a nuisance parameter and the aim is to make inference about $f(x)$. The Gibbs algorithm starts with some initial value for y, y_0, and generates an initial value of x, x_0, from the conditional $f(x|y = y_0)$. The sampler then uses this generated value of x to draw a new y value from the second conditional $f(y|x = x_0)$.

The sampler continues to simulate further y's from its conditional distribution and then simulate the next x's conditional on all the updated parameters, y. That is, the sampler proceeds to generate y's and x's in the ath MCMC cycle where

$$x_a \sim f(x|y = y_{a-1}) \quad \text{and} \quad y_a \sim f(y|x = x_a)$$

in the iterative manner described. Repeating these steps R times creates a Gibbs sequence of length R, where the first B simulations are discarded as a burn-in period (Gamerman and Lopes, 2006). Only draws from iteration $B + 1$ onwards are retained as a reliable sample from the full joint distribution.

The advantage of the Gibbs algorithm is its improved efficiency in cases where it is easier to sample from a sequence of conditional distributions that are standard known distributions rather than from the full unknown joint distribution directly. The 100% acceptance rate can become a problem, however, as this can lead to slower mixing.

A common example is the poor performance of the Gibbs method when sampling from a bivariate normal with high correlation, in which case the posterior becomes very elongated. The Gibbs sampler often gets stuck in one part of the narrow ridge of the posterior because all proposed moves depend on the previous position and are accepted, hence they are highly correlated (Gamerman and Lopes, 2006).

If we are dealing with a joint distribution of more than two variables, we can consider using a variant called the *Block-Gibbs* sampler. In this case, we update a set of parameters as a group by sampling from their joint distribution when conditioned on all the other parameters. This sampler will typically mix better than the component-wise update in the same situation.

2.2.3 MH-within-Gibbs algorithm

As mentioned before, the MH sampler avoids the problem of needing to calculate the normalisation constant seen in equation 2.5. It is however problematic that the algorithm as described in 2.2.1 relies on the crucial fact that it is possible to enumerate the entire set of feasible route flows, $\mathcal{Y}(\boldsymbol{x})$.

Enumeration of the set of feasible route flows is straightforward in a small network, however, it quickly becomes fraught with difficulty in larger and/or more complex networks. Let us for instance imagine a network with 50 nodes, where 30 and 18 travellers enter the system at nodes 1 and 2 respectively without anyone leaving at the second node. If we then assume a single trip ends at each of the remaining 48 stops, the feasible set $\mathcal{Y}(\boldsymbol{x})$ is generated in a combinatorial manner by selecting the 30 stops where trips started at node 1 could terminate. Thus $\mathcal{Y}(\boldsymbol{x})$ will have $\binom{48}{30} \approx 7 \cdot 10^{12}$ elements. This is a large number of elements considering the simplicity of the example. This makes it obvious that determining all solutions to the Equation 1.1 is typically impractical in more realistic settings, thus providing the incentive to sample the feasible route flows from a suitable proposal distribution instead.

The multidimensional posterior we want to sample from is

$$f(\boldsymbol{\theta}, \boldsymbol{y}|\boldsymbol{x}) = f(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{x}) f(\boldsymbol{y}|\boldsymbol{x}) \tag{2.18}$$

using a two-stage Metropolis-Hastings-within-Gibbs sampler as developed by Hazelton (2010b). This algorithm proceeds by iteratively sampling from the full conditionals, where the full conditional of $\boldsymbol{y}$ is:

$$f(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{x}) = \frac{f(\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{\theta})}{f(\boldsymbol{x}|\boldsymbol{\theta})} = \frac{f(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\theta}) f(\boldsymbol{y}|\boldsymbol{\theta})}{f(\boldsymbol{x}|\boldsymbol{\theta})} = \frac{\mathbb{I}_{\boldsymbol{x}=A\boldsymbol{y}} f(\boldsymbol{y}|\boldsymbol{\theta})}{f(\boldsymbol{x}|\boldsymbol{\theta})} \tag{2.19}$$

and the full conditional of $\boldsymbol{\theta}$ is:

$$f(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{x}) = f(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{f(\boldsymbol{y}|\boldsymbol{\theta}) f(\boldsymbol{\theta})}{f(\boldsymbol{y})}. \tag{2.20}$$

In the first step, we sample latent route flows, $\boldsymbol{y}$. The challenge here lies in finding an algorithm that allows us to sample candidate route flows without explicitly specifying the entire feasible set $\mathcal{Y}(\boldsymbol{x})$. Evading the potential computational black-hole of enumerating all feasible route flows would greatly improve the efficiency of the overall MCMC sampler.

In the second step of the algorithm, we sample the parameters of the OD matrix conditional on the route flows. In principle, this is a straightforward procedure, as the conditional distribution may be a known standard distribution.

We return to our example where we consider the rate of route flows $\boldsymbol{y}$ to be Poisson distributed with common mean vector $\boldsymbol{\lambda}$. If we use standard conjugate Gamma priors, $f(\boldsymbol{\lambda}) = \mathsf{Gam}(\breve{a}, \breve{b})$, then the posterior will be a standard distribution we can directly sample from: $f(\boldsymbol{\lambda}|\boldsymbol{y}) = \mathsf{Gam}(\breve{a} + \boldsymbol{y}^{\cdot}, \breve{b} + T)$, where $\boldsymbol{y}^{\cdot} = \sum_{t=1}^{T} \boldsymbol{y}^t$ and T is the number of observational periods.

The overall algorithm of the MH-within-Gibbs sampler is as follows:

- step 0: Initialise both $\boldsymbol{\theta}$ and $\boldsymbol{y}$.

  For each cycle $a = 1, .., R$

- step 1: Sample $\boldsymbol{y}_a$ conditional on $\boldsymbol{\theta}_a$ via MH.
- step 2: Sample $\boldsymbol{\theta}_a$ conditional on $\boldsymbol{y}_{a-1}$ via MH or Gibbs.

We generate samples from each of the conditionals using the MH algorithm in order to avoid evaluating the normalisation constant. If the model is conjugate, as is the case in our example, then we use a Gibbs sampler in step 2.

For the overall MH-within-Gibbs algorithm, the Gibbs sampler component lies in the repetition of steps 1 and 2 for a suitably large number of iterations R. It generates samples from the joint posterior $f(\boldsymbol{\theta}, \boldsymbol{y}|\boldsymbol{x})$ after convergence has been reached.

2.3 Methods for sampling route flows

In the first step of the MH-within-Gibbs sampler described in 2.2.3, modellers have until now failed to produce an efficient way to sample feasible candidates directly from the set of non-negative route flows that are consistent with the observed link counts.

2.3.1 General solution

A sampling method was presented by Tebaldi and West (1998) which applies to any kind of traffic network system. The idea of their method was to essentially to decompose the problem into two smaller problems. The algorithm works as follows: first, reorder the columns of the routing matrix so the partition of $\boldsymbol{A} = (\boldsymbol{A}_{[1]}, \boldsymbol{A}_{[2]})$ has a non-singular square matrix $\boldsymbol{A}_{[1]}$. Then partition the matrix of route flows $\boldsymbol{y} = (\boldsymbol{y}_{[1]}, \boldsymbol{y}_{[2]})$ accordingly.

The key purpose of this partition lies in the fact that we can reduce the computational cost considerably. According to the relationship

$$\boldsymbol{y}_{[1]} = \boldsymbol{A}_{[1]}^{-1}(\boldsymbol{x} - \boldsymbol{A}_{[2]}\boldsymbol{y}_{[2]}), \tag{2.21}$$

we need only sample feasible $\boldsymbol{y}_{[2]}$'s instead of the full set of $\boldsymbol{y}$'s. A set of $\boldsymbol{y}_{[2]}$ is considered feasible if it creates a non-negative $\boldsymbol{y}_{[1]}$ in Equation 2.21. We now sequentially sample *each element* of $\boldsymbol{y}_{[2]}$, conditional on the other elements in the vector, from a narrowing sequence of uniform intervals at each step. The upper limit of the uniform intervals is determined by the smallest traffic count on a link that is part of the route in question.

The feasibility of the sampled $\boldsymbol{y}_{[2]}$ are tested by computing the implied route flows $\boldsymbol{y}_{[1]}$ from 2.21 and checking for impossible (negative) values.

Tebaldi and West (1998) were able to show that the support for each of these single path flows is a bounded interval, so that selecting an infeasible element of $\boldsymbol{y}_{[2]}$ is followed by revising the bounds on the proposal distribution and then generating another candidate route flow.

Once a feasible route flow is generated, it is accepted or rejected using the appropriate Metropolis-Hastings acceptance probability depending on the full model likelihood.

Illustration of candidate route flow generation Consider a short linear transit network, for example a bus route, as depicted in 2.3 with four stops, where the vector $\boldsymbol{x}$=(100,68,40) are the number of travellers on the bus as it approaches stops 2, 3 and 4 respectively.

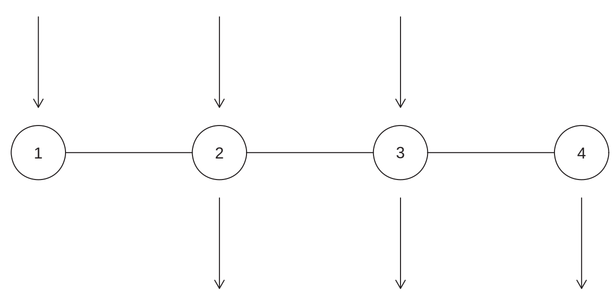


Figure 2.3: Transit network with four stops.

The first stop is an origin only and the last stop a destination only, as we assume nobody alights at the same stop as they boarded. The routing matrix $\boldsymbol{A}$ in this case is

$$\boldsymbol{A} = \begin{array}{c} \\ \underset{1}{link} \\ \underset{2}{link} \\ \underset{3}{link} \end{array} \begin{array}{c} \begin{array}{cccccc} \underset{12}{route} & \underset{13}{route} & \underset{14}{route} & \underset{23}{route} & \underset{24}{route} & \underset{34}{route} \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \end{array}.$$

The partition leads to the first three columns of the matrix $\boldsymbol{A}$ forming the matrix $\boldsymbol{A}_1$ and the last three the matrix $\boldsymbol{A}_2$ with the corresponding route flow vectors $\boldsymbol{y}_{[1]} = (y_{12}, y_{13}, y_{14})$ and $\boldsymbol{y}_{[2]} = (y_{23}, y_{24}, y_{34})$.

We now need only draw candidate route flow vectors of length 3 instead of formerly length 6, thus the problem has been made algebraically simpler. It is not considerably easier in this case, however, we note that this simplification is more pronounced in larger networks. □

The critical drawback of this method for characterising elements of the set of feasible route flows is that it is computationally inefficient on larger networks. In the algorithm we sample one component y_i of the set of feasible route flows at a time and cancel the outcome if this results in an unfeasible route flow vector. This can lead to many discarded attempts.

Another problem is the need to constantly refine the intervals for sampling the components of $\boldsymbol{y}_{[2]}$, as previously accepted values heavily influence the possible values within which further candidate values can lie, this implementation also increases computational demand.

In the simulation process we need a feasible non-negative route flow to initiate the sampler. Obviously the simple approach of finding a unique solution for the $\boldsymbol{Ay} = \boldsymbol{x}$ equation by using the Moore-Penrose generalised inverse of $\boldsymbol{A}$ cannot be used as solutions to the equation system do not necessarily fulfil the non-negativity criteria.

An elegant alternative is to regard finding solutions to $\boldsymbol{Ay} = \boldsymbol{x}$ subject to non-negativity constraints as a linear integer problem instead. However, starting points found using this technique are very unlikely cases and the sampler needs a larger number of iterations until convergence is reached; this is another factor that makes this algorithm computationally inefficient.

In addition, we can see in Figure 2.4 that the sampler moves very slowly, this is to be expected as the current candidate value depends very strongly on the former candidate value.

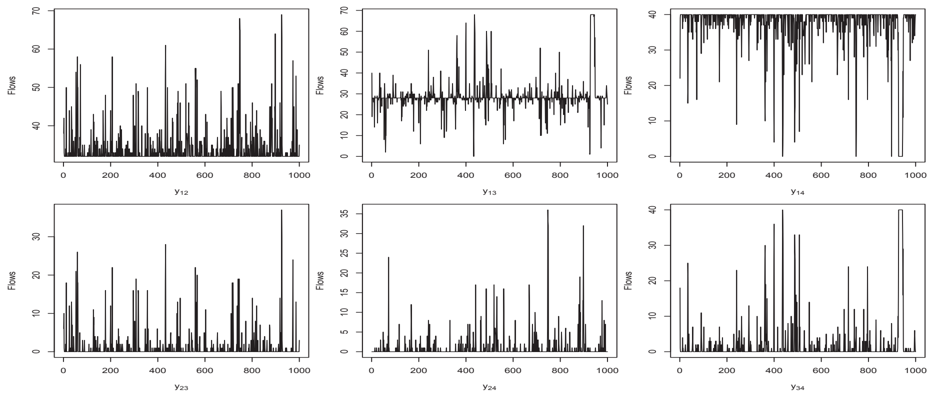


Figure 2.4: Generated route flows using Tebaldi and West algorithm.

In some cases, especially with very low traffic flows, this method can fail entirely in its attempt to find feasible solutions, see (Hazelton, 2010b). For example, if the number of people using this transit system were only 12 instead of 143 as before, the sampler fails to move at all, see Figure 2.5.

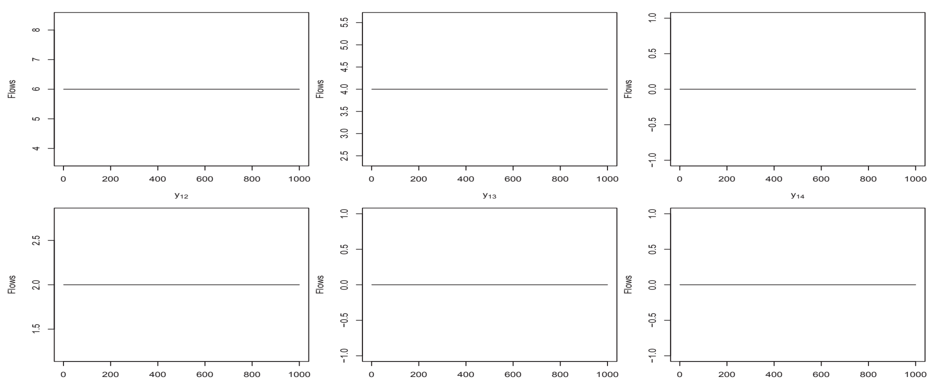


Figure 2.5: Lack of movement of Tebaldi and West sampler in case of very low demand.

2.3.2 Improvement for linear transit networks

A method which is applicable for large networks has been put forward based on an idea first put forward by Li (2008) to model the number of trips between two nodes i and j as a Markov process. Hazelton (2010b) incorporated this process into the MCMC sampler as a hidden Markov chain, from which the candidate values of $\boldsymbol{y}$ could be sampled.

Li (2008) considered general linear transit networks as depicted in Figure 2.6 with $M+1$ nodes, where the vectors $\boldsymbol{b} = (b_1, b_2, ...b_{M+1})$ and $\boldsymbol{v} = (v_1, v_2, ...v_{M+1})$ are the number of traveller's entering and exiting the network, respectively.

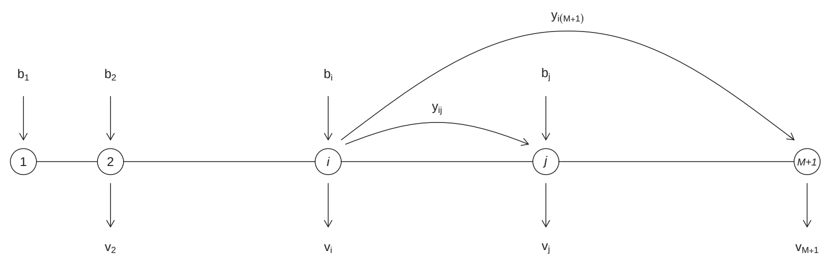


Figure 2.6: General linear network.

We begin by describing the simple model of travel behaviour following Li (2008). First, we assume the network is acyclic, that is, travel is unidirectional with trips from node i to node j impossible if i is greater than j. Further, a trip cannot end at the same node as it began. Within the framework of the OD estimation problem, the acyclic and unique routing properties lead to the OD matrix for this kind of problem being an upper-triangular matrix as follows

$$\begin{array}{cc} & \textit{Destination nodes} \\ \begin{array}{c}\textit{Origin}\\ \textit{nodes}\end{array} & \begin{pmatrix} 0 & \lambda_{1,2} & \lambda_{1,3} & \lambda_{1,4} & ... & \lambda_{1,M+1} \\ 0 & 0 & \lambda_{2,3} & \lambda_{2,4} & ... & \lambda_{2,M+1} \\ 0 & 0 & 0 & \lambda_{3,4} & ... & \lambda_{3,M+1} \\ ... & ... & ... & ... & \ddots & ... \\ 0 & 0 & 0 & 0 & 0 & \lambda_{M,M+1} \end{pmatrix} \end{array}$$

where the ijth element describes the estimated mean flow rate on the route between nodes i and j. We note that, because estimation of the boarding rates λ_i is trivial since the data are observed directly in the form of the counts $\boldsymbol{b}$, we redefine the elements as $\lambda_{ij} = \lambda_i \cdot \rho_{ij}$, where ρ_{ij} the probability of travelling from node i to j. The target for inference is then ρ_{ij}.

The second important premise is that users finish their trips in a random fashion. Thus, we assume that travellers do not take into account where they entered the system when exiting. In a bus system setting this means that someone who caught the bus at the second stop and is still on the bus as it approaches the sixth stop is just as likely to depart at the upcoming stop as somebody who got on at the fifth stop.

The assumption of people essentially forgetting where they came from when travelling allows us to calculate departure probabilities conditional on the presence in the network at the previous node instead of the origin of travel. We realise that this may appear an unrealistic premise for an application in real-life network models. We make two comments on this point. On the one hand, as Hazelton (2010b) argues, it can be shown that this travel behaviour can be motivated in terms of standard types of random utility models, for example, a logit route choice model. On the other hand, we note that for our adaption of Li (2008)'s model, we assume random travel behaviour primarily for the purpose of generating a candidate route flow vector. These candidates are then rejected or accepted in the general MCMC algorithm.

Under the two aforementioned assumptions, we can model the link counts $\boldsymbol{x}$ as a Markov process and use the Markov property to determine the transition probabilities ρ_{ij}, that is, the probability of any given user entering at node i will leave at node j, as follows

$$\rho_{ij} = \varphi_j \prod_{k=i+1}^{j-1} (1 - \varphi_k) \quad \big(1 \leq i < j \leq (M+1)\big) \tag{2.22}$$

where φ_j is the probability of exiting at node j, which is constant due to the probability of exiting being independent of where the user entered the network. Furthermore, we are able to estimate how many people leave at node j as a binomially distributed random variable $v_j \sim \mathsf{Binom}(x_{j-1}, \varphi_j)$.

Overall, the likelihood of travelers leaving the system conditional on who was present on the link leading up to the destination node j as well as the probability of leaving at that node becomes

$$f(v_j|x_{j-1}, \varphi_j) = \prod_{i=1}^{M+1} \varphi_i^{v_i} (1 - \varphi_i)^{x_{i-1} - v_i} \tag{2.23}$$

modulo irrelevant multiplicative constants. The use of this Binomial likelihood instead of the probability function in Equation 2.1, has the benefit of not requiring summation over all the elements of the set of feasible route flows. However, the likelihood in Equation 2.23 is mainly relevant for making inference about the parameter φ, the probability of leaving the system at a given node. We, on the other hand, adapt Li (2008)'s Markov model to the case where we assume we have observed both the number of vehicles joining the network at each origin node, and the number leaving at each destination node. Consequently, instead of simulating the choice of destination for each traveller conditional on the number of vehicles entering at each node, we simulate the route flows $\boldsymbol{y}$ conditional on the observed traffic counts of entering and exiting vehicles.

To this end we introduce z_{ij}, the number of travellers who board at node i and are still present in the system on the link leading up to node j, which is defined as

$$z_{j,j+1} = b_j \tag{2.24}$$

$$z_{i,j+1} = z_{i,j} - y_{i,j}, \quad \text{for } i = 1, ..., (j-1), \tag{2.25}$$

that is, the $\boldsymbol{z}$'s keep track of the origins of travellers currently in the system as they approach node j. We note from Equations 2.24 and 2.25 that the route flows $\boldsymbol{y}$ can be derived from the values of the $\boldsymbol{z}$ process conditional on $\boldsymbol{b}$. The derivation of the $\boldsymbol{y}$'s from the $\boldsymbol{z}$'s is a one-to-one transformation, which means that sampling a set of $\boldsymbol{z}$'s is equivalent to sampling a set of $\boldsymbol{y}$'s. Furthermore, we can use the proposal probabilities associated with the $\boldsymbol{z}$'s for the corresponding $\boldsymbol{y}$'s. We can now define an augmented conditional probability mass function of the auxiliary variables $\boldsymbol{z}_j = (z_{1,j}, z_{2,j}, ..., z_{\mathrm{M}+1,j})^{\mathsf{T}}$ as

$$f(\boldsymbol{z}_j|\boldsymbol{z}_{j-1}, v_j, x_{j-1}) = \prod_{i=1}^{j} P(z_{i,j}|z_{i,(j-1)}, v_j, x_{j-1}), \tag{2.26}$$

where $P(z_{i,j}|z_{i,(j-1)}, v_j, x_{j-1})$ are the transition probabilities of the hidden Markov chain which is formed by the evolution of $\boldsymbol{z}_j$, the number of vehicles that are still on the road just before it reaches node j.

In contrast to the Binomial transition probabilities in Li (2008)'s original Markov model, we are now dealing with a conditional process where we simulate the number of departures at a given node conditional on the number of people known to exit at that node. The transition probabilities are now hypergeometric instead, with parameters $z_{i,(j-1)}, v_j$ and x_{j-1}:

$$P(z_{i,j}|z_{i,(j-1)}, v_j, x_{j-1}) = \frac{\prod_{i=1}^{j-1} \binom{z_{i,j}}{y_{i,j}}}{\binom{x_{j-1}}{v_j}} \tag{2.27}$$

where $z_{i,j}$ the number of users approaching node j from nodes $i = 1, \ldots, j-1$, $y_{i,j}$ the number of users travelling from node i to node j, x_{j-1} the number of travellers on the link leading up to node j and v_j the observed number of users exiting at node j.

The hidden Markov chain of the underlying travel behaviour, $\boldsymbol{z}$, forms the proposal distribution in step 1 of the MH-within-Gibbs algorithm.

In view of the fact that the Markov model we use to define the proposal distribution can be motivated by a random utility model, it is plausible that the proposal distribution will somewhat resemble the posterior distribution. This delivers a healthy acceptance rate in the MH sampling part of the MH-within-Gibbs sampler, which greatly improves the computational efficiency.

We now show the construction of a candidate route and the calculation of its corresponding transition probabilities.

Illustration of algorithm We return to the linear transit network discussed in section 2.3.1 of a bus route with four stops, as seen in Figure 2.3, where the arrival and departure count vectors observed are $\boldsymbol{b} = (100, 10, 33, 0)$ and $\boldsymbol{v} = (0, 42, 61, 40)$, respectively. For each stop we are approaching, that is stop 2 through to stop 4, we randomly draw the number of passengers as dictated by the departure counts from the passengers currently present and record the origin of the selected leavers. For instance:

1. We record the 100 people getting on at the first bus stop as a hundred 1's: 1,1,...,1.
2. 42 1's get off the bus at the second stop. 10 people get on which are termed as ten 2's: 2,2...,2.
3. Of the 58 1's and 10 2's on the bus when approaching the third stop, we randomly draw 61 who get off. This results in 52 1's and all 2's apart from 1 leaving. 33 get on, which are labelled 3,3,...,3.
4. At the fourth and final stop all 40 people get off: 6 1's, 1 from stop 2 and the 33 3's. □

For this particular scenario, the constructed route flow vector, where the elements are kept in lexicographical order, would be $\boldsymbol{y}^{\dagger} = (42, 52, 6, 9, 1, 33)$. For example, we find that 42 passengers travel from the first to the second bus stop, 52 from the first to the third stop and so on. The corresponding number of passengers from previous stops still on the bus as it approaches the next stop is $\boldsymbol{z} = (100, 58, 6, 10, 1, 33)$. For example, the first three numbers refer to the passengers that got on at the first stop: 100 are on the bus as it approaches the second stop, 58 are still left as it approaches the third stop and 6 remain as the bus heads towards the final stop.

Once we have sampled a candidate route flow in the fashion as illustrated above, we can assign the transition probabilities. Everyone on the bus as it approaches the second stop must have come from the first stop, so the first transition probability is 1.

Next, we can calculate the probability of users that boarded at the first two stops alighting at the third stop as the hypergeometric weight

$$P(z_{1,3}, z_{2,3}|z_{1,2}, v_3, x_2) = \frac{\binom{z_{1,3}}{y^{\dagger}_{1,3}}\binom{z_{2,3}}{y^{\dagger}_{2,3}}}{\binom{x_2}{v_3}} = \frac{\binom{58}{52}\binom{10}{9}}{\binom{68}{61}} = 0.42$$

for the candidate route flow vector $\boldsymbol{y}^{\dagger}$ drawn. Analogously we calculate that travellers who boarded at the first three stops and are still on the bus as it heads towards the fourth stop alight at the next stop with probability

$$P(z_{1,4}, z_{2,4}, z_{3,4}|z_{1,3}, z_{2,3}, v_4, x_3) = \frac{\binom{z_{1,4}}{y^{\dagger}_{1,4}}\binom{z_{2,4}}{y^{\dagger}_{2,4}}\binom{z_{3,4}}{y^{\dagger}_{3,4}}}{\binom{x_3}{v_4}}$$

which will be equal to 1 for our example, as the fourth stop was the final stop.

Overall, for any given set of candidate routes sampled in this manner, we can calculate the proposal probability as follows:

$$q(\boldsymbol{y}) = \prod_{j=2}^{J} P(z_{i,j}|z_{i,(j-1)}, v_j, x_{j-1}) \tag{2.28}$$

where J denotes the number of time periods on which data was collected multiplied by the number of nodes in the network. The probability $P(z_{i,j}|z_{i,(j-1)}, v_j, x_{j-1})$ is defined as in Equation 2.27. In our example the proposal probability for the candidate route flow vector $\boldsymbol{y}^{\dagger}$ is $q(\boldsymbol{y}^{\dagger}) = 0.42$, with a single observational period, $T = 1$.

2.3.3 Extension to tree networks

In network topology, tree networks are defined as structures of linked nodes, as seen in Figure 2.7, where each node has a unique predecessor but may have multiple successors and the nodes follow a specific order. The ordering means that each node in the network has a higher number than its predecessor. The branching is where the name of these kind of networks originates from. In particular, we focus on traffic systems with at most two successors for each node, that is, links bifurcate at some of the nodes and travellers choose to turn left or right. Exiting the network counts as one of the two possibilities.

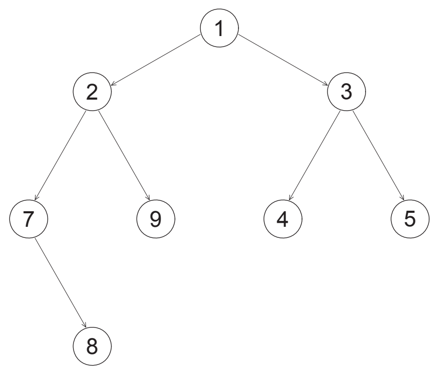


Figure 2.7: Example of a tree network.

We note that while we describe the technique for the case of binary trees, we can readily extend this method to trees where we may face more than two options at a given node by introducing dummy nodes. For example, if the links fork at a destination node, we split that node into two dummies nodes to allow for the three options of turning left, right and leaving.

In this section we will develop an extension of Hazelton (2010b)'s work from transit to tree networks. The hidden Markov model for the case of tree networks is remarkably similar to that of the linear networks because these systems share the important property that every node can only be approached through a single node.

As each node in a tree network has a unique predecessor, we can take a random sample from users present at each node and track how people progress within the network as before with linear networks. This is not the case with directed acyclic networks, where it is possible for travellers arriving at a given node to have more than one origin. In this case, a random draw at a given node may include travellers who have already been sampled which renders the combinatorial calculation of the proposal probabilities incorrect.

The extension of the route flow sampling method from linear networks to tree networks mainly involves some pre-processing before applying the MCMC sampler. For each current state up to two possible successive states are possible, it therefore becomes necessary to record how many travellers turn left and right respectively at each intersection, as constrained by the observed link counts. In our computer implementation of this methodology, we coded the algorithm so that entering the network is considered turning right and leaving the system at a terminal node is considered as turning left.

Illustration of pre-processing. The simplest example of a tree network is displayed in Figure 2.8, with nodes 1 and 2 as origins and 3 and 4 as destinations. For each node

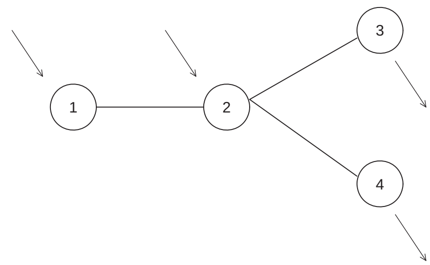


Figure 2.8: Simple tree network.

we record the predecessor (`pred`), whether this node is approached via a left turn or not (`left`) and whether or not it is a destination node (`term`) or an intersection node (`split`). It is important that successor nodes are always labelled with higher numbers, as the ordering is then used when simulating the candidate route flows.

`node`	`pred`	`left`	`term`	`split`
1	0	0	0	0
2	1	0	0	1
3	2	1	1	0
4	2	0	1	0

Table 2.1: Network topology of four-node tree network seen in Figure 2.8.

Using the information in Table 2.1, the pre-processing for this example would be:

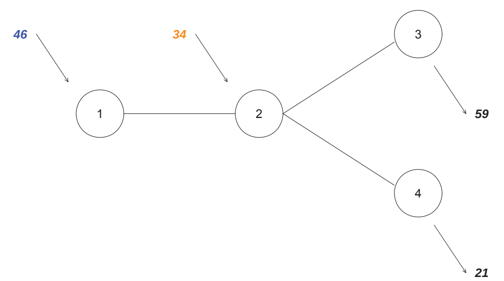
- We begin with the final node, node 4. We determine that the number of people leaving is equivalent to the number of people on the link leading up to it.
- The same is the case for node 3 where the people leaving are equivalent to those observed on the preceding link.
- The technicalities arise for node 2 as it is an intersection. We determine for which two nodes it is the unique predecessor, which are nodes 3 and 4. We assign node 3 as being reached by going ‘left’, and node 4 the destination node after turning ‘right’. The number of people travelling on the link leading up to this node are calculated as the combined number of people who we recorded as leaving at the final two nodes minus the number of arrivals at node 2.
- Lastly, node 1 is a special case where it suffices to record the number of people observed arriving.

For example, observing link counts $\boldsymbol{x} = (46, 59, 21)$ will result in the passengers choices at each node as seen in Table 2.2, where the columns correspond to the nodes in the network and the rows reflect the two branches of every node.

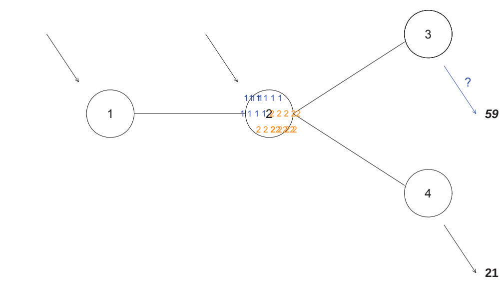
	node 1	2	3	4
turn left	46	21	0	0
turn right	0	59	59	21

Table 2.2: Table of binary choices possible for observed link counts $\boldsymbol{x} = (46, 59, 21)$.

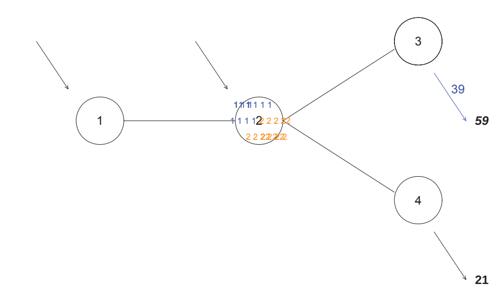
With the network topology 2.1 as well as the overview of intersection choices 2.2, we have completed all pre-processing necessary for embarking on the generation of candidate route flows.



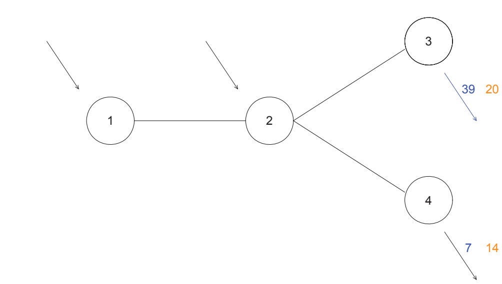
(a) As dictated by the link counts, 46 trips begin at node 1 and 34 at node 2. This results in a total of 80 vehicles at node 2.



(b) We draw a random sample of 59 vehicles who will turn left towards node 3. We record how many of the vehicles selected entered the network at the first node.



(c) In our random draw we found that 39 of the selected vehicles are from node 1.



(d) As the network is very simple, this is the final step needed to reconstruct the entire route flow vector. It is $\boldsymbol{y} = (39, 7, 20, 14)$.

Figure 2.9: Illustration of the steps in the construction of a candidate route flow $\boldsymbol{y}$.

The method for sampling a candidate route now accounts for the two outcomes possible at each junction by randomly drawing the number of people to turn 'right', at each node instead of the number that are exiting the network altogether as in the linear network case.

The most important step in Figure 2.9 was step (b) where the overall value of the route flow vector is determined by the random sample drawn. Random samples such as these occur more often in larger networks, and the number of possible candidate route vectors is therefore considerably larger than in this simple case.

Now that we have finished describing the procedure of producing a candidate route flow vector for the overall MCMC sampler, we consider the calculation of the transition probabilities. As described before with linear transit networks, a hypergeometric density provides the distribution of the resulting candidate route flow vectors; in the example given we have the following hypergeometric distribution:

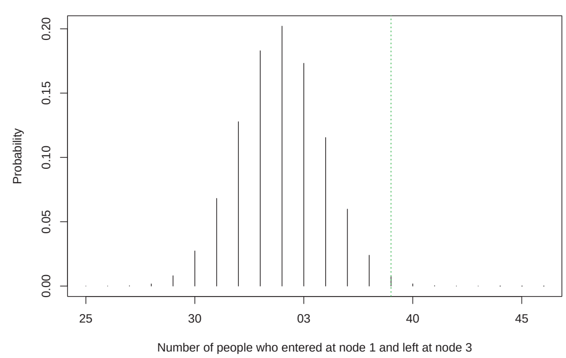


Figure 2.10: Hypergeometric probabilities of associated candidate route flow vectors. Green dashed line represents probability of generated $\boldsymbol{y}^{\dagger} = (39, 7, 20, 14)$.

As before, as seen in Equation 2.28, we use probabilities calculated in this manner to determine the proposal probabilities used in the general MCMC algorithm.

2.4 Simulation study

A major competitor to the Bayesian MCMC method is the generalised least squares (GLS) inferential approach. The main advantage of the GLS approach is the computational efficiency even for larger networks and the guaranteed convergence of the algorithm (Bell, 1991). However, as discussed before, the GLS method does rely heavily on the choice of the target matrix.

A comparison in Bell (1991) shows the GLS estimation produces very similar results to the Bayesian inference approach as described by Maher (1983). However, in both these articles, the authors consider networks with high traffic volumes.

They apply a multivariate normal approximation with a fixed dispersion matrix, which is independent of the mean flows when estimating the parameters of the OD matrix. Yet, so far no comparisons of these two methods have been made when a normal approximation of the counts is not possible.

In the following sections, we conduct a simulation study to demonstrate the performance of the estimates obtained using the Bayesian MH-within-Gibbs sampler in comparison to estimates obtained via GLS estimation for networks with low as well as high volumes of traffic.

The advantage of a simulation study, where we simulate the data, is that it enables us to evaluate how well the estimated average route flows match up with the average rates originally used to create the generated link counts. We consider a combinational experimental design with three factors, network topology, level of demand in the system and the number of observational periods. All permutations of the latter two factors lead to a total of eight scenarios for each network considered, as seen in Figure 2.11.

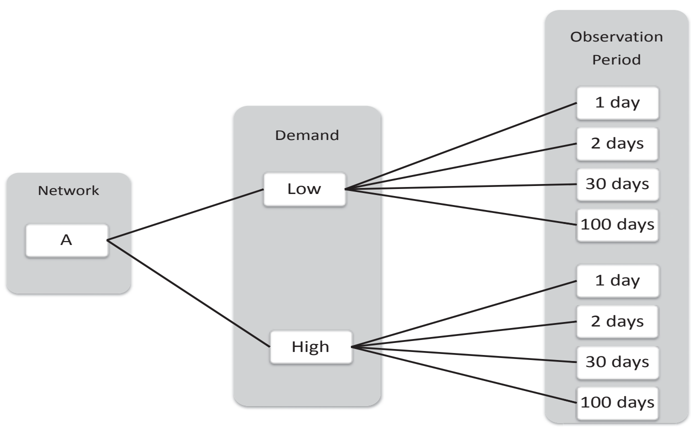


Figure 2.11: Overview of the eight possible scenarios for each network, e.g. A.

The first factor is the complexity of the network layout, here represented by three network topologies, A, B and C. We choose this type of network as a means to illustrate our methodology. Topology A is the simplest kind of tree network, with only four nodes, as seen in Figure 2.8. Topology B is a tree network with eight nodes and the final network is taken from a section of the state highway 16 in Auckland, New Zealand and has 14 nodes.

The second factor is the network demand with two levels, high and low. Including lower levels of activity is of interest as this highlights any problems the sampler may run into when dealing with the effect of sparse data, something that our developed sampler was designed to deal with to a better extent.

We generate the observations corresponding to the two levels of demand as follows. First, we randomly draw values from a uniform distribution. In the case of low levels of travel, the boundaries of the uniform distribution were 0 to 5, and for high demand 1 to 100. Secondly, we assume these values to be the mean rate of a Poisson distribution which we sample from. The random sample of Poisson distributed numbers constitute the link counts.

The third and final factor, the observational time frames, has four levels corresponding to an increasing number of days on which data was collected, with the options being 1,2, 30 and 100 days.

Data collection from the network is assumed to be standardised during the analysis period, that is data is collected in the same locations and at the same times. Similarly, the pattern of activities that generates the trips is assumed to be kept fixed, e.g if the time slot happens to be around peak hour, higher activity levels are expected.

For each of the eight scenarios for networks A, B and C respectively, we measure the accuracy of the resulting estimates via a root mean squared error standardised by the total demand

$$\epsilon = \frac{\sqrt{\sum_{r=1}^{N}(\hat{\lambda}_r - \lambda_r)^2}}{\sum_{r=1}^{N}\lambda_r}, \tag{2.29}$$

where the mean of the posterior $\hat{\lambda}_r$ is the estimated mean route flow of route r. Since we repeat this procedure for 100 simulated data sets each element in the table of errors represents an average over the resulting 100 error terms.

A moderate number of iterations by MCMC sampler standards, a total of one million (100,000 runs with thinning of every 10th value), was considered as the computational burden was otherwise too high. The simulation of estimates for 100 generated data sets took an average of about 1 1/2 days to complete for the simplest case of low demand in network A and data observed on a single day. We also performed a convergence analysis, that is, we checked whether the chain had converged after the one million iterations. If the MCMC chain is mixing badly, it follows that it will take more iterations to converge, as the chain is moving around the parameter space slowly.

We remove the first 20% of MCMC runs as burn-in. The analysis of convergence diagnosis (Geweke, 1992) showed this was adequate. Finally, we repeated the estimation using the GLS method as described in section 1.3.2. We model the *a priori* estimates of the mean route flows $\check{\lambda}_r$ as $\mathsf{Gam}(0.1, 0.1)$ distributed values. As we obtain estimates using the same simulated link counts, we can directly compare the recorded error terms to the errors from the MCMC method.

2.4.1 Topology A

Topology A is the simple four node network seen in Figure 2.8. The simplicity of this network makes it possible to visually explore the properties of the simulated traffic flows using the improved MCMC sampler for tree networks. The histograms of the estimated average route flows for each model parameter are shown in Figure 2.12.

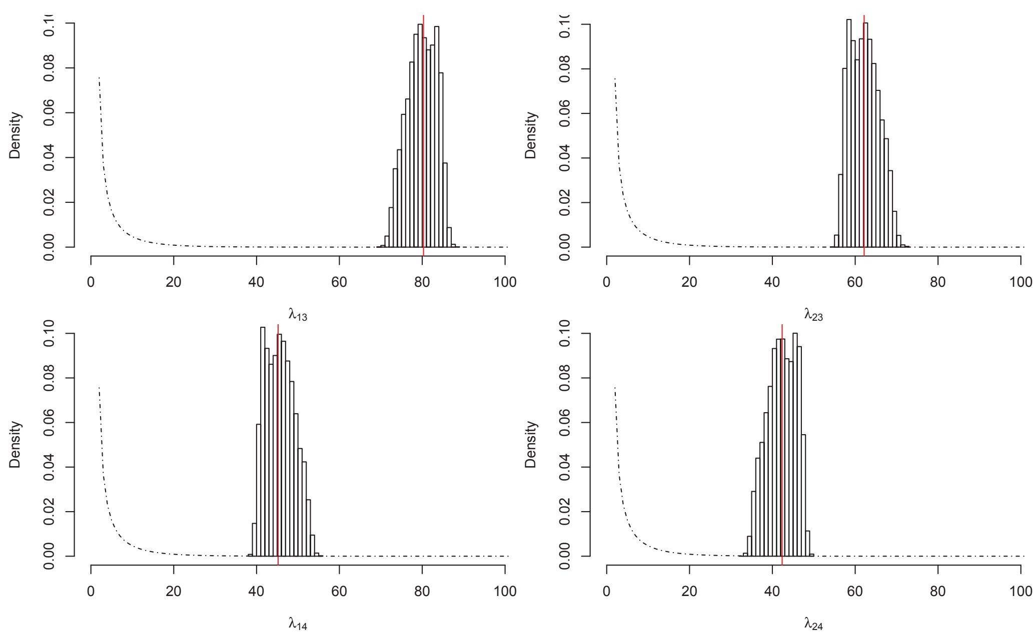


Figure 2.12: Histogram of estimated traffic flows on each of the four routes for high demand over 100 days. The dashed line represents the $\mathsf{Gam}(0.1{,}0.1)$ prior.

These histograms are the estimated marginal posterior densities of $\lambda_{13}, \lambda_{23}, \lambda_{14}$ and λ_{24}. The prior densities are superimposed as a dashed line. The simulations were, with reference to Figure 2.11, based on data generated for a network experiencing high levels of demand and data collected on 100 days in total. The mean route flow value used to generate the link counts is displayed as a vertical red line in each plot.

Clearly, the modes of the histograms are centred around the true value, which confirms that the sampler is doing well. In particular, we can see that the prior appears to have little influence on the resulting posterior distributions. Furthermore, there are two features that are noteworthy. The first point is that there are signs of an interdependence between the parameters. We observe that the average number of trips from node 1 to 3, λ_{13}, tends to be underestimated more frequently while λ_{14} appears to be overestimated more often. A linkage between the route flow rates is to be expected as an artefact of the simplicity of the network. If travellers who enter at node 1 do not leave at node 3 they must exit at node 4. Ergo, if the number predicted to leave at node 3 is too high, then too few will be predicted to have left at node 4.

The impression of a relationship between the parameters λ_{13} and λ_{14} manifests itself when we look at the bivariate histogram in Figure 2.13. We can see that when the flows from node 1 to node 3 are high, the flows from node 1 to node 4 are lower.

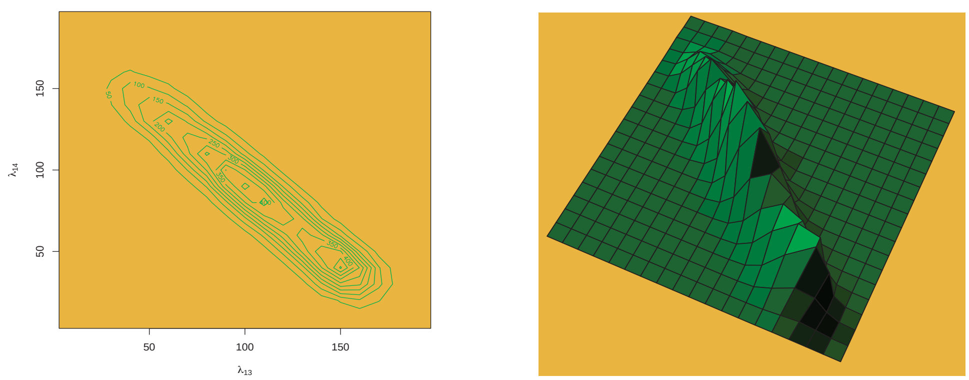


Figure 2.13: Contour and perspective plots of the bivariate histogram of λ_{13} and λ_{14}

The second aspect is the slight hint of bi-modality. The cause for this may be that there are (at least) two overlapping distributions of equally likely scenarios. In other words, the posterior may be a distribution comprising of two or more components.

Further evidence that we are dealing with a mixture of distributions comes from an examination of traceplots. The degree to which the MCMC chain is mixing or moving around the parameter space can be visually inspected using a traceplot for every parameter, see (Geweke, 1992). A traceplot is a plot of the accepted parameter value for each iteration plotted against the iteration number.

The traceplot in Figure 2.14 for one parameter of the OD matrix, λ_{14}, confirms that there are mixing issues. The trace shows clear signs of clustering, that is, the chain appears to get stuck in certain areas of the parameter space.

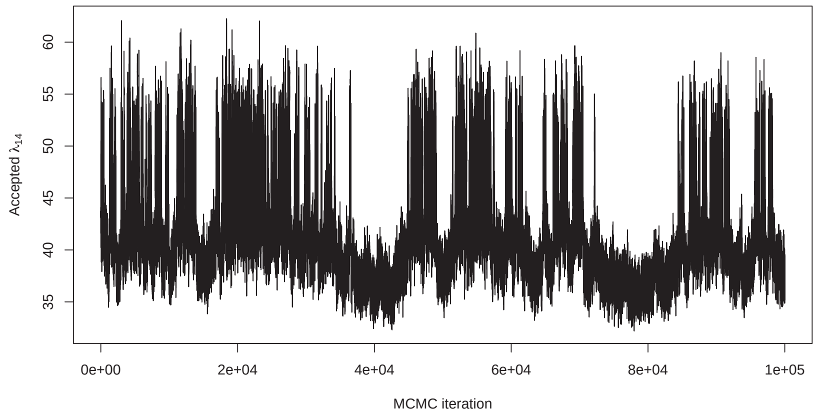


Figure 2.14: Traceplot for λ_{14} in network A, where we assume high demand and an observational period of 100 days.

It is important to emphasise that the MCMC chain has converged to a unique posterior, yet it is a posterior that is bimodal- or at least a mixture of at least two components.

We now consider the Bayesian estimation errors, denoted as ϵ_{MCMC_A}, for the four node tree example seen in 2.8, which are shown in Table 2.3.

		Number of data collection days			
		1	2	30	100
Demand	*low*	0.34 (0.14)	0.29 (0.13)	0.08 (0.03)	0.04 (0.02)
	high	0.13 (0.07)	0.12 (0.07)	0.08 (0.06)	0.06 (0.05)

Table 2.3: Mean values and associated Monte Carlo standard errors (in brackets) of MCMC estimation errors ϵ_{MCMC_A}.

The level at which the estimated mean flows match up with the mean flows used to generate the data, on average, significantly improves with the increase in the number of days on which data was collected. In fact, it appears to be an even stronger factor than whether the traffic flow levels were high or not, as the level of accuracy is similar for both low and high levels when the observation period reaches 30 or more days.

Nonetheless, overall we find that the additional data available with traffic systems experiencing high demand does improve estimation, especially if only one or two days worth of counts are available the rate is more than halved (0.13 and 0.12 when demand high as opposed to 0.34 and 0.29, respectively, when demand low).

The level of accuracy of the GLS estimates, seen in Table 2.4, is similar to that of the MCMC estimates when traffic volumes are modest and the observation period short. However, the trend of progressively more accurate estimates that we see in the MCMC estimates for longer observation periods is non-existent in the GLS case. We discuss why this is to be expected in section 2.5.

		Number of data collection days			
		1	2	30	100
Demand	*low*	0.38 (0.18)	0.36 (0.15)	0.31 (0.16)	0.30 (0.16)
	high	0.28 (0.16)	0.27 (0.16)	0.27 (0.14)	0.27 (0.16)

Table 2.4: Mean values and associated Monte Carlo standard errors (in brackets) of GLS estimation errors $\epsilon_{\mathrm{GLS}_A}$.

The biggest disadvantage of the MCMC method is that, even in the case of the improved algorithm introduced in this thesis, the computations needed vastly exceed those required to perform the GLS estimation.

In this simple four node case, the MCMC simulations continue for over a day, whilst the GLS produces results after quarter of an hour. Yet the vastly inferior learning ability offsets the computational advantage of the GLS method in cases where data from many observation periods are available, as in this case the estimation errors of the GLS are nearly fivefold larger than their MCMC counterparts, for example, with high traffic volumes and 100 observation periods the error rates are 0.27 versus 0.06, respectively.

2.4.2 Topology B

Now we repeat the same procedure for a slightly more complex network which can be seen in Figure 2.15. The first node is considered an origin only, nodes 2-4 are both origin and destination nodes and the remaining nodes 5-8 are all destinations only.

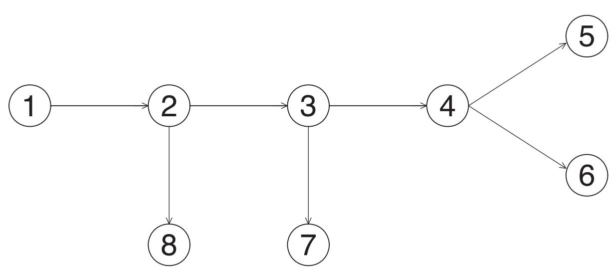


Figure 2.15: Network B with eight nodes connected by seven links.

The resulting Bayesian estimation errors, $\epsilon_{\mathrm{MCMC}_B}$, for this network are given in Table 2.5.

		Number of data collection days			
		1	2	30	100
demand	*low*	0.22 (0.05)	0.20 (0.06)	0.10 (0.04)	0.06 (0.02)
	high	0.11 (0.03)	0.11 (0.04)	0.10 (0.03)	0.09 (0.04)

Table 2.5: Mean values and associated Monte Carlo standard errors (in brackets) of MCMC estimation errors $\epsilon_{\mathrm{MCMC}_B}$.

Again, we see a trend of increasingly better MCMC estimations for larger numbers of observation periods in the case of only low traffic demand in the system. This trend is noticeably less pronounced when the traffic system is undergoing high demand. Nonetheless, when demand is high, the accuracy of the MCMC estimated is considerably improved, with error rates nearly half of their equivalents in the low demand case when data was only collected once or twice.

The GLS estimate errors, $\epsilon_{\mathrm{GLS}_B}$, seen in Table 2.6 stay fairly constant with both factors, number of collection days and demand levels, showing no strong effect on the overall level of accuracy of the estimation results.

		Number of data collection days			
		1	2	30	100
demand	*low*	0.22 (0.06)	0.21 (0.05)	0.19 (0.05)	0.19 (0.05)
	high	0.18 (0.06)	0.18 (0.06)	0.18 (0.05)	0.18 (0.06)

Table 2.6: Mean values and associated Monte Carlo standard errors (in brackets) of GLS estimation errors $\epsilon_{\mathrm{GLS}_B}$.

The average rate of 0.2 for estimation using sparse data, where many routes have only low levels of traffic flows, does not diverge notably from the average rate of 0.18 for estimates based on data with high traffic flows in the system. The impression that the GLS estimation method does not benefit from increasing amounts of information derived from link counts to the same extent as the MCMC sampler gathers more support in network B.

2.4.3 Topology C

We now consider a tree network, the state highway 16 in Auckland, New Zealand. See Figure 3.9 for a schematic representation of the route.

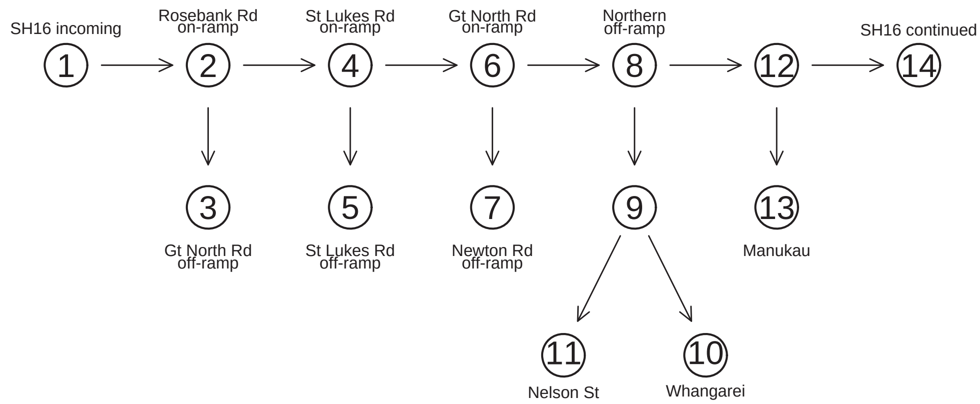


Figure 2.16: Section of State Highway 16, Auckland, New Zealand.

The section that we consider is northbound, starting at the Rosebank Rd on-ramp and progressing along Newton Road to the Whangarei/Nelson Street interchange. At the interchange, the highway splits into two sections, one leading to the Whangarei and Nelson Street off ramps and the other section to the Manukau/Hamilton off ramp. In this case, we simulated the data as Poisson distributed with mean rates based on average flows measured on the state highway 16 in 2010.

In Table 2.7 we see the results of the MCMC estimation. We note that the computational burden of the MCMC sampling process is exceedingly large, with the iterations now running over a period of nearly four days for this network (in the 30 data collection days scenario).

		Number of data collection days			
		1	2	30	100
demand	*low*	0.14 (0.022)	0.12 (0.019)	0.09 (0.016)	0.06 (0.011)
	high	0.09 (0.007)	0.08 (0.006)	0.08 (0.004)	0.07 (0.004)

Table 2.7: Mean values and associated Monte Carlo standard errors (in brackets) of MCMC estimation errors $\epsilon_{\mathrm{MCMC}_C}$.

When comparing Tables 2.7 and 2.8, we see that the level of accuracy for one or two observation periods and low demand is similar for the two methods. This is to be expected considering that in the absence of data the Bayesian approach is strongly influenced by the prior, and as previously discussed the GLS method always relies heavily on the choice of the target matrix.

In all other cases, the GLS estimates do not reach the same level of accuracy as the MCMC estimates, with the difference becoming more pronounced the more the number of data collection days increases.

		Number of data collection days			
		1	2	30	100
demand	*low*	0.14 (0.03)	0.14 (0.03)	0.13 (0.02)	0.13 (0.02)
	high	0.13 (0.02)	0.13 (0.02)	0.13 (0.02)	0.13 (0.02)

Table 2.8: Mean values and associated Monte Carlo standard errors (in brackets) of GLS estimation errors $\epsilon_{\mathrm{GLS}_C}$.

One reassuring element we see for this third example, which represents the most complex topology considered, is that the larger the network, the more accurate the estimates. This may be explained by the fact that correlation structures provide additional information to the extent that we observe a further reduction of estimation errors.

2.5 Conclusion

When making Bayesian inference for traffic networks, we often find ourselves confronted with the the difficulty of obtaining the exact posterior. The issue at hand is twofold. Firstly, we cannot evaluate the normalisation constant in the denominator of the posterior. This first problem can be solved by using a MCMC sampler, such as the Metropolis-Hastings sampler, where the normalisation constant is not needed anymore. Secondly, in the case of low traffic flow counts in particular, we assume the traffic flows are discrete, for example following a Poisson distribution.

For discrete traffic flow models, explicitly evaluating the likelihood involves specifying the set of all route flows that are consistent with the observed link counts. As this set can contain such a large number of elements that even identification is computationally difficult, it is advisable to sample from the set of feasible route flows instead.

Existing work still lacks a computationally efficient sampling methodology for sampling from the set of feasible route flows. The technique developed by Tebaldi and West (1998) is noteworthy in that it is applicable to general networks, but is highly computationally inefficient.

The most important contribution in this chapter is the expansion of a MCMC sampler developed by Hazelton (2010b), based on work by Li (2008) where a hidden Markov process of the underlying travel behaviour is used to sample candidate feasible route flows. This MCMC sampler was originally developed for linear networks only and has now been expanded to be applicable to tree networks.

Linear and tree networks cover a large range of real-life network systems, from transit systems to highways, but obviously there are still many traffic networks that do not belong to either one of these two classes of network topologies. One future course of investigation could be to extend our proposed methodology to be applicable to a wider range of network types. The concept of the MCMC sampler introduced in this chapter relies completely on natural ordering and is therefore not extendable to 'unordered' networks, but could potentially be developed for acyclic networks.

Overall, this method is especially useful when a network system is experiencing low levels of demand, which means we need to work with the discrete models, such as the Poisson distribution. In this case maximum likelihood estimation becomes computationally intractable. Our method is still less computationally expensive than other methods available, with the only drawback being that it is very restricted in terms of the network topologies it can be applied to.

Our method is general enough to also apply to the case when the traffic counts are high and a multivariate normal approximation can be used. In this case, however, inference via ML or GLS estimation is more advisable, as either of these approaches require considerably less computer power than the MCMC sampler.

In the case of large enough counts, Hazelton (2001a) compares the ML with the GLS estimation method and found the ML method performed significantly better in the absence of prior information or in the presence of measurement error, but otherwise the GLS approach returned similar results whilst being computationally more feasible.

In this chapter, we compare the GLS approach to our MCMC method, as both methods can be applied in both low and high demand cases. Surprisingly, we found that even for the high demand case, the MCMC sampler returns more accurate estimates, but more importantly, we demonstrate that our sampler significantly improves the accuracy estimation when data was collected on a higher number of days, while the GLS estimates show no progress.

As Hazelton (2001a) already noted, the GLS estimates have the disadvantage of not being able to incorporate the extra information given by increased numbers of observation periods.

The reason for this lies in the fact that the GLS method attempts to minimise the distance between the average observed link counts, $\bar{\mathbf{x}}$ and the average rates given by, say, a formerly estimated model, $\bar{\mathbf{x}} = \mathbf{A}\hat{\boldsymbol{\lambda}}$. As the GLS estimator $\mathbf{A}\hat{\boldsymbol{\lambda}}$ is a linear function of $\bar{\mathbf{x}}$, the correlation structure over the days is ignored. This makes solving the identifiability problem more difficult, because, as Hazelton (2003) points out, there is an advantage when using the extra information available through the variability in the data.

However, the computational burden of the MCMC sampler, while an improvement over other currently available MCMC methods, is still very high. Further work on implementing an even more efficient sampler that can compete in terms of computational needs with methods such as the GLS technique, is another course for future work in this area.

Chapter 3

Inference for Day-to-Day Traffic Assignment Models

Think ahead. Don't let day-to-day operations drive out planning.

(Donald Rumsfeld)

3.1 Introduction

Network-based models play a critical role in many planning and management activities for road transport systems. For many years deterministic equilibrium models were dominantly used, however, while such models are mathematically tractable, they are by nature static and therefore do not account for changes in travel patterns over time. This leaves them ill suited for analysis of traffic data sets with vehicle counts gathered from a sequence of observation periods, for example a succession of days (Peeta and Ziliaskopoulos, 2001).

Thus, interest has shifted towards more dynamic aspects of traffic systems and research in the transport community has progressed towards models which focus on the evolution of travel patterns over time (Watling and Van Vuren, 1993).

There are two types of these kind of models, within-day and day-to-day dynamical models. Within-day dynamical models aim to capture the dependence of route choice on the time of departure, while taking the proper temporal and spatial evolution of congestion across the network into account. Typically, the evolution of a traffic system over a short amount of time is of interest, for example in the work from Nie (2011) the analysis period is 100 seconds.

Day-to-day dynamic models in contrast, work with data that is typically observed on time intervals *over a number of successive days*, which puts the development of travel choices and traffic congestion over days at the centre of attention. In other words, these kind of models focus on the response of drivers from one day to another to exogenous information.

An important type of such models is described in Cascetta (1989), where the evolution of a transport system is represented as a Markov chain. In this type of model, travellers make route choices on a given day and the decision-making process is linked through a learning model based on past events such as the travel costs experienced. Extensions of this work are given in Cantarella and Cascetta (1995) and Cascetta and Cantarella (1991), which focus on providing an overall modelling framework which applies to both categories of dynamical models.

In addition to providing a representation of the traffic system dynamics, the properties of the stationary distribution of the Markov chain, such as the mean and the variance, describe interesting characteristics of the system of interest. On this note, Watling and Hazelton (2003) highlight the important relationship between dynamic and traditional equilibrium models, for example, as demand increases, the mean equilibrium flow of any Markov model will converge to SUE under certain mild conditions.

Day-to-day models show considerable potential, but they tend to have many parameters, for example, we not only need to estimate travel demand parameters (typically in the form of an origin-destination matrix) but also parameters that calibrate traveller behaviour (for instance, the scaling parameter in logit route choice models). At present, parameters in these models are typically estimated by ad hoc means. It is not uncommon for values for the parameters describing traveller behaviour to be borrowed from other studies, even when it is far from clear as to whether they are appropriate for the case at hand.

An elegant statistical approach to this parameter estimation problem is to use a modification of the overall MH-within-Gibbs algorithm we developed in section 2.2.3.

Herein, we focus on two aspects. Firstly, we face an exponential rise in the complexity of the sampling problem with the increase in the number of days for which we have data. We seek to decompose the problem of sampling route flows for all days simultaneously in one iteration run into a sequence of conditional sampling problems for each day in turn. Secondly, we discussed in the previous chapter the issue of designing a computationally efficient algorithm for sampling feasible route flows within the overall MCMC sampler, and described two algorithms in particular that were developed to address this problem.

In this chapter we perform a simulation study where we compare the performance of these two route flow sampling algorithms; the technique proposed in Tebaldi and West (1998)'s work and the method we developed which generates candidates from the set of feasible flows using a proposal process based on a simple Markov model of traveller behaviour.

3.2 The day-to-day model likelihood

In day-to-day modelling, traffic counts from a *sequence* of T days are collected, this means we can no longer assume the link flows, $\boldsymbol{x}$, to be independent. That is, traffic flows from a succession of days will lead to a certain degree of inter-day dependence. We can imagine this as akin to weather patterns, where certain conditions on one day tend to influence conditions in the days thereafter. We denote the link counts as $\boldsymbol{x} = (x_1, ..., x_M)$, and now add a superscript t to index the day of observation, i.e. $\boldsymbol{x}^t$ are the link counts on day t, t=1, . . . , T.

Again, the superscript t is used to index the day for $\boldsymbol{y} = (y_1, ..., y_N)$, the corresponding route flow vector, where N is the number of routes in the network.

Under the general class of day-to-day Markov traffic models studied by Cascetta (1989) and Cantarella and Cascetta (1995), the distribution of $\boldsymbol{y}^t$ is defined conditionally on the finite history from the past d days, $\boldsymbol{y}^{t-1}, \ldots, \boldsymbol{y}^{t-d}$, and some model parameters $\boldsymbol{\theta}$ governing travel demand and traveller behaviour. That is, we can model the evolution of route flows day-to-day using a d-step Markov process with transition probabilities

$$f(\boldsymbol{y}^t|\boldsymbol{y}^{t-1}, \boldsymbol{y}^{t-2}, ..., \boldsymbol{y}^{t-d}, \boldsymbol{\theta}) \tag{3.1}$$

where f denotes a probability distribution as defined by its arguments and $\boldsymbol{\theta}$ a vector of model parameters.

We model the dependence of $\boldsymbol{y}^t$ for a given day t on $\boldsymbol{y}^{t-1}$ through the use of link cost functions. To this end we assume that the flow x_l on link l generates a link specific travel cost $c(x_l)$. Travel costs are defined as a combination of monetary elements such as the amount of fuel needed (e.g. on a hilly stretch as opposed to a road on a plain), congestion charges or tolls, and non-monetary factors such as the time needed to travel along the link as well as the appeal of the road environment (e.g. a scenic route in contrast to using the motorway) or ease of driving. The cost of travelling along a route is then given by

$$\boldsymbol{k} = \boldsymbol{A}^{\mathsf{T}}\boldsymbol{c} \quad \text{where} \quad \boldsymbol{c} = (c(x_1), \ldots, c(x_M))^{\mathsf{T}}. \tag{3.2}$$

Statistical inference for models of the form seen in Equation 3.1 are difficult because they require us to explore a high-dimensional space of route flows, $\{\boldsymbol{y}^t, \boldsymbol{y}^{t-1}, ..., \boldsymbol{y}^{t-d}\}$. A natural way to simplify the expression in Equation 3.1 is to assume that decisions made on day t depend on the previous states of the system only through the travel costs on the preceding d days. In this case:

$$\begin{aligned} f(\boldsymbol{y}^t|\boldsymbol{y}^{t-1}, \boldsymbol{y}^{t-2}, ..., \boldsymbol{y}^{t-d}, \boldsymbol{\theta}) &= f(\boldsymbol{y}^t|\boldsymbol{c}^{t-1}, \boldsymbol{c}^{t-2}, ..., \boldsymbol{c}^{t-d}, \boldsymbol{\theta}) \\ &= f(\boldsymbol{y}^t|\boldsymbol{x}^{t-1}, \boldsymbol{x}^{t-2}, ..., \boldsymbol{x}^{t-d}, \boldsymbol{\theta}). \end{aligned} \tag{3.3}$$

Logit route choice model We present a specific class of such d-step Markov models as described in Hazelton and Watling (2004), in part to clarify some of the notation introduced as well as to provide a framework within which we perform our simulation study later in this chapter.

We suppose that there is a vector $\boldsymbol{\mu} = (\mu_1, ..., \mu_L)$ of average OD flow rates and that the realised vector of OD flows on day t is given by $\boldsymbol{w} = (w_1^t, ..., w_L^t)^\mathsf{T}$ with w_o^t following a $\mathsf{Pois}(\mu_o)$ distribution independently of $w_{o^*}^s$ for $o^* \neq o$ or $s \neq t$.

We apply the logic route choice model to calculate the probability that a traveller will select route r on day t for a journey between OD pair o as follows:

$$\varrho_{or}^t = \frac{e^{\psi u_r^t}}{\sum_{r^* \sim r} e^{\psi u_{r^*}^t}} \tag{3.4}$$

where $r^* \sim r$ if and only if routes r^* and r serve the same oth OD pair and ψ is the logit model parameter.

In Equation 3.4, we model the travel utility u^t on day t as a function of travel costs experienced by drivers previously, in the sense that heavy congestion on one day is liable to lead to reduced demand the next. Specifically, u_r^t is defined by the rth element of the linear filter

$$\boldsymbol{u}^t = -\sum_{s=1}^{d} \delta^s \boldsymbol{k}^{t-s}. \tag{3.5}$$

The powers of δ provide exponentially decreasing weights, with $\delta > 0$ chosen so that $\sum_{s=1}^{d} \delta^s = 1$. The traveller learning process is hence determined by the length of memory d and the rate δ at which the travellers consider past experiences decays with time.

The lth element of $\boldsymbol{c}$, the vector of link costs, could be defined here via the commonly used Bureau of Public Roads (1964)'s link cost function,

$$c(x_l) = a_l^* \left(1 + 0.15 \left(\frac{x_l}{b_l^*}\right)^4\right) \tag{3.6}$$

where a_l^* is the free-flow travel time and b_l^* a measure of road capacity, for $l = 1, ..., M$. We assume that, conditional on past costs and the realised demand on day t, travellers choose their routes independently. It follows that the conditional distribution of route flow choices for any given OD pair on day t follows a Multinomial distribution:

$$\begin{aligned} f(\boldsymbol{y}_r^t | \boldsymbol{w}^t, x^{t-1}, x^{t-2}, ..., x^{t-d}, \boldsymbol{\theta}) &= \mathsf{Mn}(\boldsymbol{w}^t, \boldsymbol{\varrho}^t) \\ &= \frac{w_o^t!}{\prod\limits_{r^* \sim r} y_r^t} \prod_{r^* \sim r} (\varrho_{or}^t)^{y_r^t} \end{aligned} \tag{3.7}$$

where $r^* \sim r$ if and only if routes r^* and r serve the same OD pair and ϱ_{or} the route choice probability corresponding to the oth OD pair. Our model is of the form of Equation 3.3 with travel demand parameter vector $\boldsymbol{\theta} = (\psi, d, \delta, \boldsymbol{\mu}^\mathsf{T})^\mathsf{T}$.

3.3 Modification of MCMC algorithm

We seek to draw inference in a similar manner as described in chapter 2, with the difference that we regard the travel demand parameter vector $\boldsymbol{\theta}$ based on a sequence $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^T$ of link counts instead of flows observed at intermittent time points. In the absence of useful prior information we will employ a vague prior for the probability distribution of $\boldsymbol{\theta}$ before observing the link counts. Within the Bayesian paradigm the prior distribution is combined with the model likelihood, $L(\boldsymbol{\theta})$, to give the posterior distribution for $\boldsymbol{\theta}$ as follows:

$$f(\boldsymbol{\theta} | \boldsymbol{X}) = \frac{L(\boldsymbol{\theta}) f(\boldsymbol{\theta})}{\int \mathsf{L}(\boldsymbol{\theta}) f(\boldsymbol{\theta})} \tag{3.8}$$

where $\int \mathsf{L}(\boldsymbol{\theta}) f(\boldsymbol{\theta})$ is the marginal probability function of the data, which acts as a normalising constant.

There are a number of problems that will make direct computation of the model likelihood infeasible in all but toy examples.

To illustrate these difficulties, suppose that the memory length is just $d = 1$ day, and that we have data for only $T = 2$ days. Then the full likelihood for a $d-$step Markov model from Equation 3.3 is given by

$$\begin{aligned} \mathsf{L}^*(\boldsymbol{\theta}) &= f(\boldsymbol{X}|\boldsymbol{\theta}) \\ &= f(\boldsymbol{x}^1, \boldsymbol{x}^2|\boldsymbol{\theta}) \\ &= f(\boldsymbol{x}^1|\boldsymbol{\theta}) f(\boldsymbol{x}^2|\boldsymbol{x}^1, \boldsymbol{\theta}). \end{aligned} \tag{3.9}$$

where the matrix $\boldsymbol{X}$ is a concatenation of the daily link flow vectors, $\boldsymbol{X} = (\boldsymbol{x}^1, ..., \boldsymbol{x}^t)^\mathsf{T}$.

The term $f(\boldsymbol{x}^1|\boldsymbol{\theta})$ is problematic because it requires evaluation of the unconditional probability of $\boldsymbol{x}^1$, while our Markov model is defined in terms of conditional (i.e. transition) probabilities and we have no previous link counts to condition on. In order to even define the likelihood in Equation 3.9 we need to make assumptions about this unconditional distribution. Arguably it is simplest to assume that the Markov process is stationary, as a consequence the distribution of $\boldsymbol{x}^1$ is defined by the equilibrium distribution of the Markov chain. However, this distribution is intractable although asymptotic approximations are sometimes available (Hazelton and Watling, 2004).

Turning to the second term on the right-hand side of Equation 3.9, we must condition on $\boldsymbol{y}^2$ because, in accordance with Equation 3.3, the parameter vector $\boldsymbol{\theta}$ is defined in terms of the distribution of the route flows. Therefore we obtain

$$f(\boldsymbol{x}^2|\boldsymbol{x}^1, \boldsymbol{\theta}) = \sum_{\boldsymbol{y}^2} f(\boldsymbol{x}^2|\boldsymbol{x}^1, \boldsymbol{y}^2, \boldsymbol{\theta}) f(\boldsymbol{y}^2|\boldsymbol{x}^2, \boldsymbol{\theta}) = \sum_{\boldsymbol{y}^2 \in \mathcal{Y}(\boldsymbol{x}^2)} f(\boldsymbol{y}^2|\boldsymbol{x}^1, \boldsymbol{\theta}). \tag{3.10}$$

where $\mathcal{Y}(\boldsymbol{x}^t) = \{\boldsymbol{y}^t : \boldsymbol{x}^t = \boldsymbol{A}\boldsymbol{y}^t, \boldsymbol{y}^t \geq 0\}$ is the set of feasible route flows on day t. Direct calculation of the expression 3.10 is typically not possible, as discussed in the previous chapter, because it involves summation over all the elements of the set of feasible route flows on day t. This in turns renders summarising the posterior distribution seen in Equation 3.9 impossible.

We use MCMC samplers to get an approximation of the posterior, in particular, we propose modifying the MCMC algorithm we developed in chapter 2. In order to obviate the need to enumerate the feasible route flow sets $\{\mathcal{Y}(\boldsymbol{x}^t)\}$ we incorporate a route flow sampler into the MCMC algorithm. This means we sample from the joint posterior distribution $f(\boldsymbol{Y}, \boldsymbol{\theta}|\boldsymbol{X})$, where the matrix $\boldsymbol{Y}$ is a concatenation of the daily route flow vectors, $\boldsymbol{Y} = (\boldsymbol{y}^1, ..., \boldsymbol{y}^t)^\mathsf{T}$, instead of directly sampling from the posterior $f(\boldsymbol{\theta}|\boldsymbol{X})$ and then make inference for $\boldsymbol{\theta}$ using the sampled values of this parameter vector alone.

Factorisation of the joint posterior distribution The joint posterior distribution $f(\boldsymbol{Y}, \boldsymbol{\theta}|\boldsymbol{X})$ can be factorised conveniently using the Markov property described in Equation 3.3, which we show explicitly in the case of $d = 1$ as follows.

$$\begin{aligned} f(\boldsymbol{Y}, \boldsymbol{\theta}|\boldsymbol{X}) = &\ f(\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^t, \boldsymbol{\theta}|\boldsymbol{X}) \\ = &\ f(\boldsymbol{y}^t, \boldsymbol{\theta}|\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^{t-1}, \boldsymbol{X}) \cdot f(\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^{t-1}, \boldsymbol{\theta}|\boldsymbol{X}) \\ &\ \vdots \\ = &\ f(\boldsymbol{y}^t, \boldsymbol{\theta}|\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^{t-1}, \boldsymbol{X}) \cdot f(\boldsymbol{y}^{t-1}, \boldsymbol{\theta}|\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^{t-2}, \boldsymbol{X}) \\ &\ \cdots f(\boldsymbol{y}^3, \boldsymbol{\theta}|\boldsymbol{y}^2, \boldsymbol{y}^1, \boldsymbol{X}) \cdot f(\boldsymbol{y}^2, \boldsymbol{\theta}|\boldsymbol{y}^1, \boldsymbol{X}) \cdot f(\boldsymbol{y}^1, \boldsymbol{\theta}|\boldsymbol{X}). \end{aligned} \tag{3.11}$$

The expression 3.11 can be simplified due to the first-order Markov chain property of all previous states beyond $t-1$ being independent of the state t:

$$f(\boldsymbol{y}^t, \boldsymbol{\theta}|\boldsymbol{y}^{t-1}, \boldsymbol{X}) \cdot f(\boldsymbol{y}^{t-1}, \boldsymbol{\theta}|\boldsymbol{y}^{t-2}, \boldsymbol{X}) \cdots f(\boldsymbol{y}^3, \boldsymbol{\theta}|\boldsymbol{y}^2, \boldsymbol{X}) \cdot f(\boldsymbol{y}^2, \boldsymbol{\theta}|\boldsymbol{y}^1, \boldsymbol{X}) \cdot f(\boldsymbol{y}^1, \boldsymbol{\theta}|\boldsymbol{X}). \tag{3.12}$$

Importantly, the dependence of $\boldsymbol{y}^t$ on $\boldsymbol{y}^{t-1}$ is only through the costs from the previous day, $\boldsymbol{c}^{t-1}$, which in turn depend only on $\boldsymbol{x}^{t-1}$, not $\boldsymbol{y}^{t-1}$. In addition, $\boldsymbol{y}^t$ depends on the flows from the current day through the linear relationship 1.1. From this it follows that

$$f(\boldsymbol{y}^t, \boldsymbol{\theta}|\boldsymbol{y}^{t-1}, \boldsymbol{X}) = f(\boldsymbol{y}^t, \boldsymbol{\theta}|\boldsymbol{y}^{t-1}, \boldsymbol{x}^{t-1}, \boldsymbol{x}^t) = f(\boldsymbol{y}^t, \boldsymbol{\theta}|\boldsymbol{x}^{t-1}, \boldsymbol{x}^t). \tag{3.13}$$

Overall, we have thus shown that

$$\begin{aligned} f(\boldsymbol{Y}, \boldsymbol{\theta}|\boldsymbol{X}) = &\ f(\boldsymbol{y}^t, \boldsymbol{\theta}|\boldsymbol{x}^t, \boldsymbol{x}^{t-1}) \cdot f(\boldsymbol{y}^{t-1}, \boldsymbol{\theta}|\boldsymbol{x}^t, \boldsymbol{x}^{t-1}) \cdots \\ &\ f(\boldsymbol{y}^3, \boldsymbol{\theta}|\boldsymbol{x}^t, \boldsymbol{x}^{t-1}) \cdot f(\boldsymbol{y}^2, \boldsymbol{\theta}|\boldsymbol{x}^t, \boldsymbol{x}^{t-1}) \cdot f(\boldsymbol{y}^1, \boldsymbol{\theta}|\boldsymbol{x}^t, \boldsymbol{x}^{t-1}), \end{aligned} \tag{3.14}$$

that is, that the route flows are independent conditional on the link flows and we can sample from $\mathcal{Y}(\boldsymbol{x}^1), \ldots, \mathcal{Y}(\boldsymbol{x}^T)$ in turn, rather than sampling en bloc from the cross product of these sets for the number of days given.

We show the factorisation in the case of $d = 1$ for simplicity of the presentation of results, although extensions to larger values of d are straightforward in principle.

The importance of being able to factorise the posterior in this manner lies in the fact that if this property was not given, we would need to either accept or reject the whole set of $\boldsymbol{y}$'s as a whole, which would give a very low acceptance rate, in particular if we have data from many days. This would make the sampling algorithm highly inefficient as a low acceptance rate leads to bad mixing.

Finally, before modifying our algorithm, we need to consider how to handle data from the first d days as they are not fully specified in our model.

We will consider the d traffic patterns to be fixed, independent of $\boldsymbol{\theta}$ and condition on the link counts from the first day of collection, $\boldsymbol{x}^{d+1}$, onwards since we do not want to make assumptions (e.g. stationarity) about the process prior to the time of data collection. Under these settings, the joint distribution in Equation 3.14 for general $d > 1$ expands as

$$\begin{aligned} f(\boldsymbol{Y}, \boldsymbol{\theta} | \boldsymbol{X}) &= f(\boldsymbol{y}^1, \boldsymbol{\theta} | \boldsymbol{x}^1) f(\boldsymbol{y}^2, \boldsymbol{\theta} | \boldsymbol{x}^2, \boldsymbol{x}^1) \cdots f(\boldsymbol{y}^d, \boldsymbol{\theta} | \boldsymbol{x}^{d-1}, \ldots, \boldsymbol{x}^1) \\ &\quad \times \prod_{t=d+1}^{T} f(\boldsymbol{y}^t, \boldsymbol{\theta} | \boldsymbol{x}^{t-1}, \boldsymbol{x}^{t-2}, \ldots, \boldsymbol{x}^{t-d}), \end{aligned} \tag{3.15}$$

simplifies to the conditional posterior

$$f(\boldsymbol{Y}, \boldsymbol{\theta} | \boldsymbol{X}) = \prod_{t=d+1}^{T} f(\boldsymbol{y}^t, \boldsymbol{\theta} | \boldsymbol{x}^{t-1}, \boldsymbol{x}^{t-2}, \ldots, \boldsymbol{x}^{t-d}). \tag{3.16}$$

In the case that $d = 1$ the conditional posterior in Equation 3.16 based on a first-order Markov chain reduces to

$$f(\boldsymbol{Y}, \boldsymbol{\theta} | \boldsymbol{X}) = \prod_{t=2}^{T} f(\boldsymbol{y}^t, \boldsymbol{\theta} | \boldsymbol{x}^{t-1}). \tag{3.17}$$

The factorisation of the posterior in 3.16 suggests a two-stage sampling algorithm in which we iterate between sampling from the conditional distributions $f(\boldsymbol{Y} | \boldsymbol{\theta}, \boldsymbol{X})$ (stage 1) and $f(\boldsymbol{\theta} | \boldsymbol{Y}, \boldsymbol{X})$ (stage 2). We recall the overall MH-within-Gibbs algorithm from section 2.2.3, which in this case is as follows:

Overall MH-within-Gibbs sampler for day-to-day inference:

- step 0: Initialise both $\boldsymbol{\theta}_0$ and $\boldsymbol{Y}_0$.

  For each cycle $a = 1, .., R$

- step 1: Sample $\boldsymbol{Y}_a$ conditional on $\boldsymbol{\theta}_{a-1}$ via MH.
- step 2: Sample $\boldsymbol{\theta}_a$ conditional on $\boldsymbol{Y}_a$ via MH or Gibbs

After a suitable burn-in period this algorithm will generate samples from the (joint) posterior $f(\boldsymbol{Y}, \boldsymbol{\theta} | \boldsymbol{X})$. Comments on implementing steps 1 and 2 are described in the next two sections.

3.3.1 Conditional Sampling of Route Flow Vectors

Following the simplification provided in Equation 3.16, we need only sample the set $\{\boldsymbol{y}^t : t \geq d+1\}$. Furthermore, we have demonstrated that the route flow vectors $\boldsymbol{y}$ are independent conditional on the link flows. The advantage of this lies in the ability to define the day-to-day dependence in terms of the distribution of $\boldsymbol{y}^t$ given $\boldsymbol{x}^{t-1}, \ldots, \boldsymbol{x}^{t-d}$ rather than conditioning on $\boldsymbol{y}^{t-1}, \ldots, \boldsymbol{y}^{t-d}$, which allows us to proceed day by day when applying the MCMC sampling algorithm.

We are now in the position to give the general form of the MH sampler for sampling from the conditional $f(\boldsymbol{y}^t|\boldsymbol{\theta}, \boldsymbol{X})$ which can be expressed as

$$\begin{aligned} f(\boldsymbol{y}^t|\boldsymbol{\theta}, \boldsymbol{X}) &\propto f(\boldsymbol{y}^t|\boldsymbol{\theta}, \boldsymbol{x}^{t-1}, \ldots, \boldsymbol{x}^{t-d}) f(\boldsymbol{x}^t|\boldsymbol{y}^t, \boldsymbol{\theta}, \boldsymbol{x}^{t-1}, \ldots, \boldsymbol{x}^{t-d}) \\ &= f(\boldsymbol{y}^t|\boldsymbol{\theta}, \boldsymbol{x}^{t-1}, \ldots, \boldsymbol{x}^{t-d}) \mathbb{I}_{\mathcal{Y}(\boldsymbol{x}^t)}(\boldsymbol{y}^t) \end{aligned} \tag{3.18}$$

where $\mathbb{I}$ is the indicator function.

In what follows, we let q_t denote a proposal distribution for $\boldsymbol{y}^t$ with support equal to $\mathcal{Y}(\boldsymbol{x}^t)$. This means, we assume all candidate route flow vectors $\boldsymbol{y}^\dagger$ to be feasible. If the current path vector is $\boldsymbol{y}^t$, then the updating scheme operates as follows:

For $t = d+1, \ldots, T$:

1. Draw candidate $\boldsymbol{y}^\dagger$ from proposal distribution q_t with support $\mathcal{Y}(\boldsymbol{x}^t)$.
2. Accept $\boldsymbol{y}^\dagger$ with probability

$$\alpha = \min\left[1, \frac{f(\boldsymbol{y}^\dagger|\,\boldsymbol{\theta}, \boldsymbol{X}) q_t(\boldsymbol{y}^t)}{f(\boldsymbol{y}^t|\,\boldsymbol{\theta}, \boldsymbol{X}) q_t(\boldsymbol{y}^\dagger)}\right].$$

3. If the candidate is accepted then set $\boldsymbol{y}^t = \boldsymbol{y}^\dagger$; otherwise $\boldsymbol{y}^t$ remains unchanged.

After iteratively sampling the route flow vectors over all days, we obtain a concatenated $\boldsymbol{Y} = (\boldsymbol{y}^{d+1}, ..., \boldsymbol{y}^T)$ route flow matrix which we then use to condition on when sampling $\boldsymbol{\theta}$ in step 2 of the overall MCMC sampler.

The difficulty now lies in efficiently deriving a means of sampling candidates $\boldsymbol{y}^\dagger$ from a proposal distribution q_t with support $\mathcal{Y}(\boldsymbol{x}^t)$ without enumerating this set. The development of methods for sampling feasible path flows has proved to be a major obstacle for inference for traffic models on general networks (Hazelton, 2001b).

Two methods put forward, the first as proposed by Tebaldi and West (1998) and the second the alternate method developed in chapter 2 for tree networks, are both applied here. Comparisons made in simulation studies are presented in the following section 3.4.

Conditional Sampling of the behavioural parameters We note that the conditional sampling at step 2 will generally be relatively straightforward. The conditional density is defined by

$$f(\boldsymbol{\theta}|\boldsymbol{Y}, \boldsymbol{X}) = f(\boldsymbol{\theta}|\boldsymbol{Y}) \tag{3.19}$$

because $\boldsymbol{\theta}$ is conditionally independent of $\boldsymbol{X}$ given feasible trip vectors $\boldsymbol{Y}$.

We can either use a MH sampler to sample the elements of $\boldsymbol{\theta}$ sequentially or, in the event of the model $f(\boldsymbol{\theta}|\boldsymbol{Y})$ and the prior $f(\boldsymbol{\theta})$ being conjugate we can use a Gibbs sampler.

Specifically, in our numerical examples, we use standard conjugate gamma priors for the average route flow rate vector, $f(\boldsymbol{\lambda}) = \mathsf{Gam}(\check{\boldsymbol{a}}, \check{\boldsymbol{b}})$. The prior denotes a joint density of independent gamma random variables with $\check{a}_j$ and $\check{b}_j$ as parameters for the jth one. As a result, the conditional posterior $f(\boldsymbol{\lambda}|\boldsymbol{Y})$ will follow a $\mathsf{Gam}(\check{\boldsymbol{a}}+\boldsymbol{y}_{\boldsymbol{.}}, \check{\boldsymbol{b}}+T)$ distribution with $\boldsymbol{y}_{\boldsymbol{.}} = \sum_{t=1}^{T} \boldsymbol{y}^t$, where the vector notation should be interpreted elementwise under the assumption of independence between the elements. Overall, this allows us to sample $\boldsymbol{\lambda}$ directly for the conditional posterior using the Gibbs method.

3.4 Numerical studies

We begin by examining two applications of the MCMC sampler we developed in chapter 2. For purposes of comparing the route sampling methods, we restrict ourselves in our numerical studies to tree network systems where our method can be applied.

In the first example, we consider a simple four node section of highway where the target for inference are parameters controlling the mean travel demand. Our sole purpose in this example is to illustrate the operation of the modified MCMC algorithm for the day-to-day case (using the route sampling method based on Hazelton (2010b)'s work which we developed in the previous chapter).

In the second example, we will look at a section of the road network in Auckland, New Zealand, centred on the major arterial state highway 16. We illustrate the performance of the two samplers assuming that the day-to-day flows can be modelled using a logit route choice model, a commonly used traffic model in transport science.

Here, the target for inference is a logit parameter, controlling travellers sensitivity to travel costs. For this, we compare the performance of the Tebaldi and West (1998) and our extended version of the Hazelton (2010b) route sampling methods when used within the overall MCMC algorithm.

3.4.1 Artificial example

We consider a hypothetical tree network traffic system as depicted in Figure 3.1, with four nodes and three links. We imagine that the first two nodes are origins only, and the final two nodes destinations only.

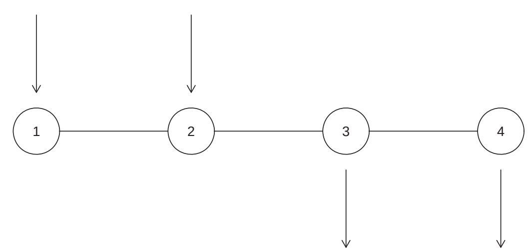


Figure 3.1: Artificial linear network with four nodes.

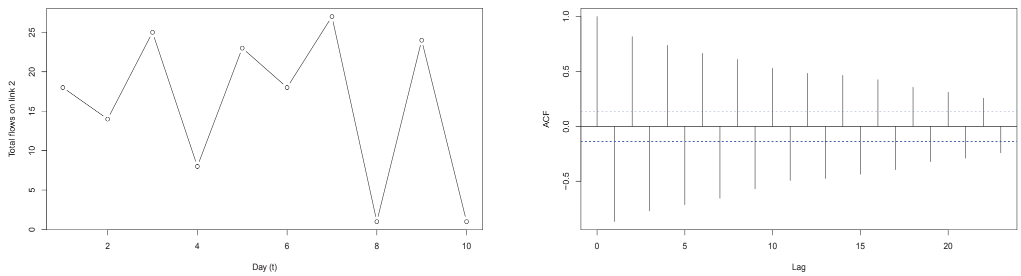
In order to investigate how correlation patterns between days can assist in the simulation process, we consider a scenario where one of the links has only recently been introduced. In the initial period afterwards, we may expect larger than usual fluctuations in demand levels on a day-to-day basis.

We assume that the parameters in the link cost function of the general form in Equation 3.6 are known, with equal length, $a_l^* = 1$, and the same capacity, $b_l^* = 20$, on all three links, l=1,2,3. We use the following truncated linear demand function to model the average flow for each of the four OD pairs, $o = 1, ..., 4$:

$$\mu_o = \mathsf{E}[y_o^t|\boldsymbol{\theta}] = \max(\beta_o - \gamma_o u_o^t, 0) \tag{3.20}$$

where u_o^t is the utility of OD pair o on day t as defined by Equation 3.5, where $\delta = 1$ because $d = 1$. In defining y_o^t and u_o^t as the traffic flow and utility for an OD pair, we are using the fact that $w_o = y_o$ in this traffic system, that is, there is a one-to-one correspondence between OD pairs and routes for networks with unique routing such as linear or tree networks. The true values of the travel demand parameters, $\boldsymbol{\theta} = (\beta_{13}, \gamma_{13}, \beta_{14}, \gamma_{14}, \beta_{23}, \gamma_{23}, \beta_{24}, \gamma_{24})^\mathsf{T}$, labelled here with a double index in the obvious manner, are (50,18,50,18,30,18,30,18), respectively.

We generate day-to-day routes flows from the model described above, and then compute the link counts as the observed data. Under the circumstances envisaged we can expect the traffic demand to oscillate in the manner depicted in Figure 3.2(a). The effect of high demand serving as a deterrent for travel on successive days introduces a negative correlation between the successive link counts, a fact which is confirmed in the autocorrelation plot of the simulated link counts in Figure 3.2(b).



(a) Function of varying demand. (b) Autocorrelation of successive link counts.

Figure 3.2: Simulated data from over a period of 10 days.

We note here that we actually compute day-to-day counts for a very large number of days, around 2000 generally, but for example in Figure 3.2, only the last 10 days of simulated daily counts are taken into account. The computed data is otherwise too heavily influenced by the link counts given as starting values. Instead the evolution over a large number of days allows the one-stage Markov process to be the driving factor in the creation of the link counts $\boldsymbol{x}$.

3.4.1.1 Reconstruction of route flows $\boldsymbol{y}$.

For illustrative purposes we initially set the parameter vector $\boldsymbol{\theta}$ to be known, and investigate the ability of the algorithm to reconstruct the distribution of the most likely route flows, $\boldsymbol{y}^t$ (which are the same as the OD flows here), via MCMC estimation. We apply the overall MCMC algorithm implemented using the Hazelton (2010b) route flow sampler to reconstruct the OD flows. The algorithm was run for R=100,000 iterations, with the first B=20,000 discarded as the burn-in period.

We visualised our results by plotting histograms, seen in Figures 3.3, 3.4 and 3.5, showing the resulting estimates of the day-to-day conditional distribution for each latent route flow, for days 2 to 10. The red bar represents the realised route flow on that particular day.

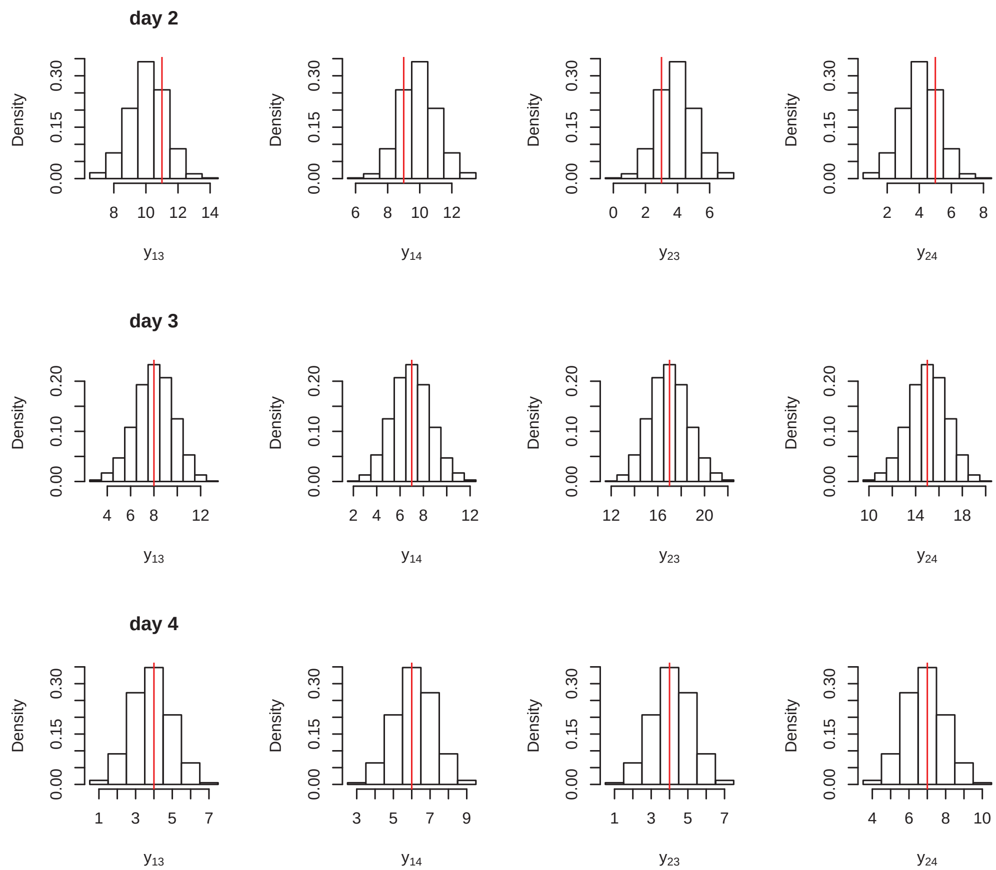


Figure 3.3: Histograms of route flows in four-node network, for days 2 to 4.

As displayed in Figure 3.3, the estimation process seems to be doing reasonably well for the first three days, with the distributions all centred around the true route flows. For day 2 we see the interdependence of the parameters, with an underestimation of the route flow from 1 to 3 leading to an overestimation of the flows from 1 to 4.

Remarkably, in Figure 3.4 the estimation on day 5 appears to be quite bad. However, this does not necessarily mean that the MCMC sampler is performing poorly. A situation like this is to be expected given that the route flows are random variables and as in this case, the actual route flow is an unusually high one, as indicated by the red bar. One of the characteristics of random variables is that they occasionally take on atypical values, something which can reflect real life scenarios. An example may be an accident on route y_{14} leading users to favour route y_{13} for a change.

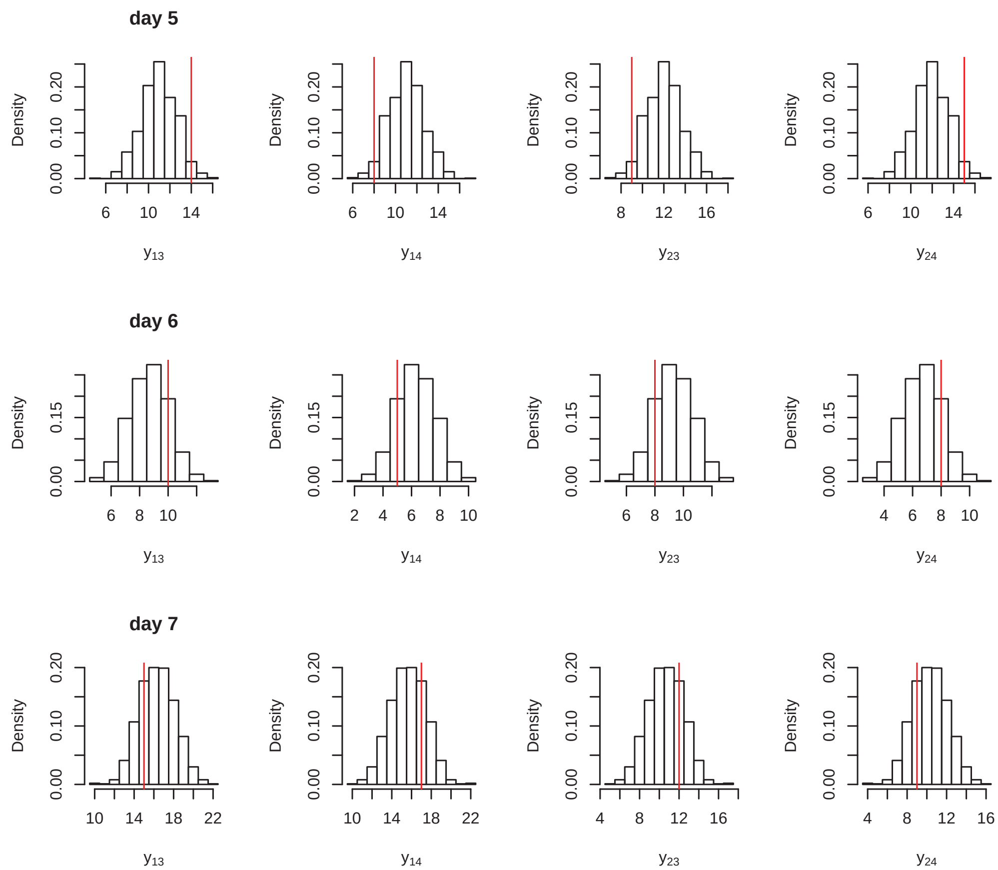


Figure 3.4: Histograms of route flows in four-node network, for days 5 to 7.

The interesting aspect is that the histogram remains centred around the most likely route flow as determined by the data on average across days and is not strongly affected by the abnormal observation. This conveniently demonstrates how this developed sampler is not intended for wholly reproducing the observed link counts, but is capable of making predictions in the long term.

In Figure 3.5 we see that the histograms for days 8 and 10 have an unusual shape. The simulated link flows on these days was very low, with only 5 and 6 travellers leaving node 1 on days 8 and 10 respectively, thus resulting in only two route flows being possible.

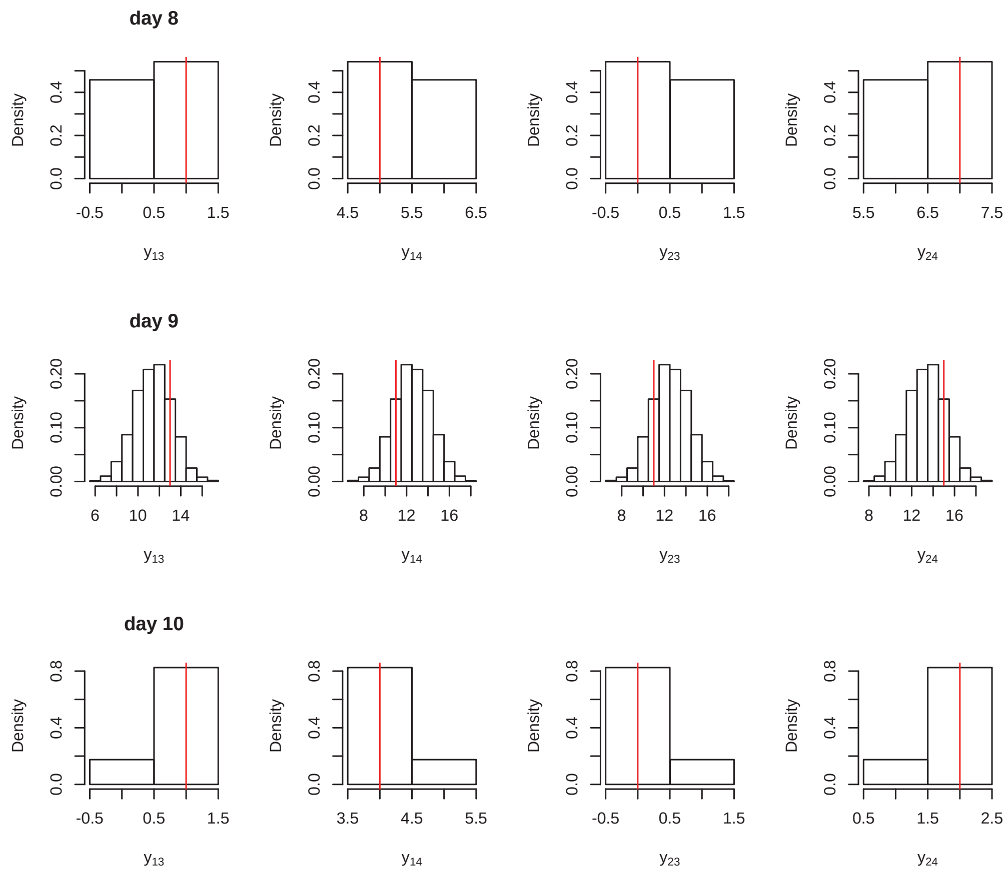


Figure 3.5: Histograms of route flows in four-node network, for days 8 to 10.

3.4.1.2 Estimation of demand parameters β and γ.

We now turn to the full problem in which the travel demand parameters β and γ are unknown. We note that the prior we use in the MCMC sampler is the same as the proposal distribution, in which case the acceptance probability is equal to a likelihood ratio. An informative prior was used, a Gamma distribution whose parameter values were modelled on the values used to generate the count data.

The histograms of the resulting estimates of the day-to-day conditional distribution for each path flow were essentially the same as in the simulation where the parameter vector was assumed to be known. Posterior densities, estimated using kernel density estimation (Wand and Jones, 1995), for the demand parameters (here indexed by OD node pair) are displayed in Figures 3.6 and 3.7 as unbroken lines.

Prior densities are shown for comparison using dashed lines. The vertical red lines indicate the parameter values at which the data were generated.

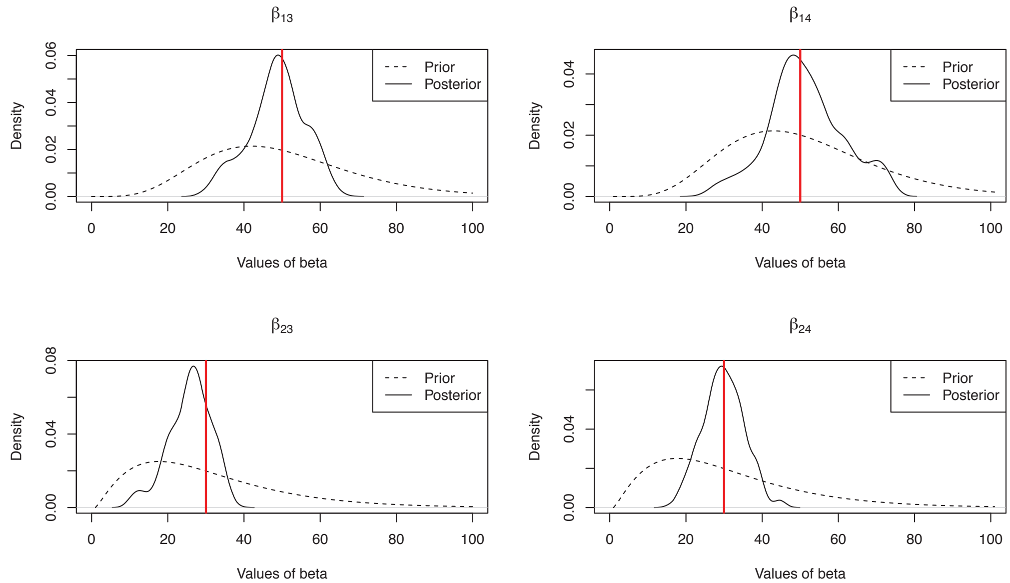


Figure 3.6: Posterior and prior density of demand parameter β.

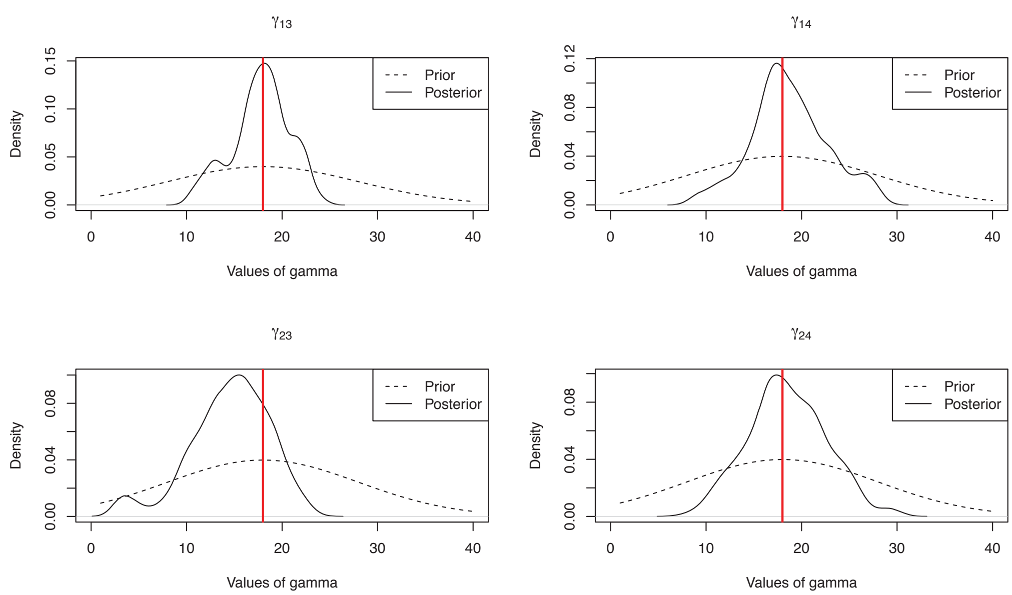


Figure 3.7: Posterior and prior density of demand parameter γ.

The modes of the posterior densities in both Figures 3.6 and 3.7 match up reasonably well with the true underlying parameter values that generated the data. The results are especially reassuring in the case of the β estimation, as the posteriors are centred on the true mode although the prior density modes deviate considerably from these values.

It can be seen in Figure 3.7 that the prior has a stronger influence in the estimation of the demand parameter γ. In particular in the plot for γ_{23} the posterior mode is some way away from the true value. Nonetheless, the modes of the posterior densities of the remaining γ parameters coincide with the true parameter values to a large degree.

3.4.1.3 Estimation of sensitivity ψ.

We now employ the traffic demand model described at the end of section 3.2. As our artificial linear network, seen in Figure 3.1, is a traffic system with unique routing we introduce a dummy route corresponding to a decision not to travel for each OD pair. The probabilities associated with these alternatives are defined by a logit random utility model as described by Equations 3.4, 3.5 and 3.7. As a consequence of there only being one real route for each OD pair and each traveller making a decision between just two alternatives, either travel by the available route or not travel at all, Equation 3.4 is now of the simpler form

$$\varrho_r^t = \frac{e^{\psi u_r^t}}{e^{\psi u_r^t} + e^{\psi u_{r_0}^t}} \tag{3.21}$$

where $u_{r_0}^t$ is the utility for not travelling on route r. We may think of $u_{r_0}^t$ as a lateness penalty that accounts for time wasted on waiting for a better opportunity to travel. We set this fixed utility to be 2 for all routes.

In this section, we suppose all other parameters are now known apart from the logit parameter in Equation 3.4 and compare the performance of the two routing samplers in terms of the estimation of ψ for this toy example.

The parameter of interest delivers a measure of the sensitivity of travellers to system changes. A high value of ψ would point to high levels of volatility in the route choice (or in this case, the decision to travel or not), whilst a value of ψ close to 0 would occur if users are not easily deterred in their travel decisions. A value of 0 entails that users don't use previous experiences in any way in their decision process. We use $\psi = 1$ for generating the link count data.

In the implementation of the MCMC sampler, using both route flows samplers, we base the prior on a $\mathsf{Gam}(0.05, 0.1)$ distribution. We note that this prior has mean $\mathrm{E}(\psi)$=0.5, and a standard deviation of $\sqrt{5}$ which is relatively high. This is representative of a fairly common situation in practice where prior knowledge is unreliable.

In Figure 3.8 we display the estimated posterior densities for ψ based on MCMC runs of 100,000 iterations with the first 20,000 as burn-in. The plots are classified by the route sampler and sample size T, which is determined by the number of days worth of data. In each case the prior density is also displayed (as a dashed line) and the true value of the parameter of interest via a vertical red line.

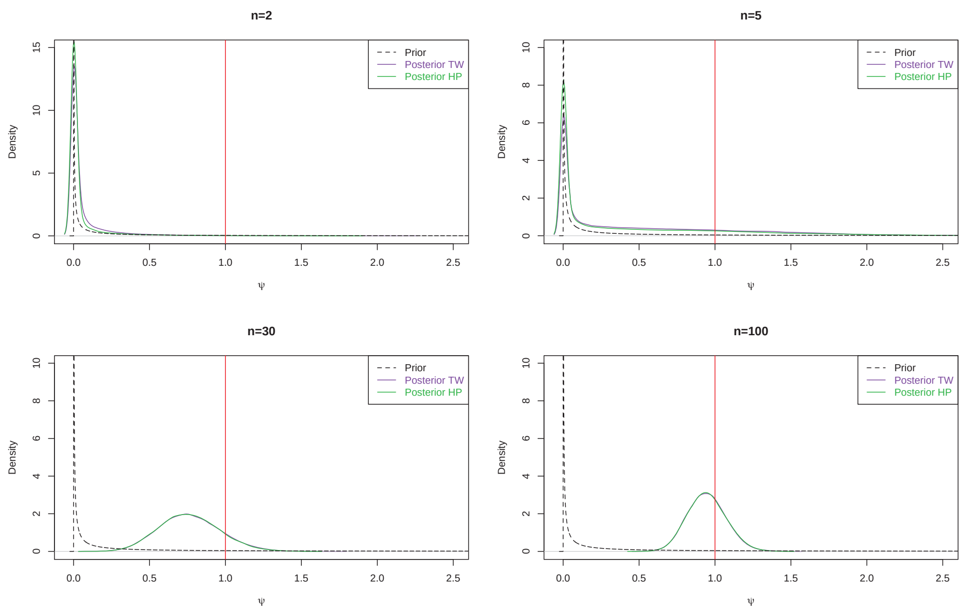


Figure 3.8: Posterior and prior density of demand parameter ψ for both methods in case of simple four-node network

We can see that both the methods perform equally well in estimating ψ for this simple example. The strong influence of the prior is obvious, with a large amount of data (at least 30 consecutive days) needed to achieve estimates close to the true value of 1 rather than the mean of the prior which is centred on 0.5.

3.4.2 State Highway 16

In this section we revisit the state highway 16 in Auckland, New Zealand network we first saw in section 2.4.3. The schematic layout can be seen in Figure 3.9.

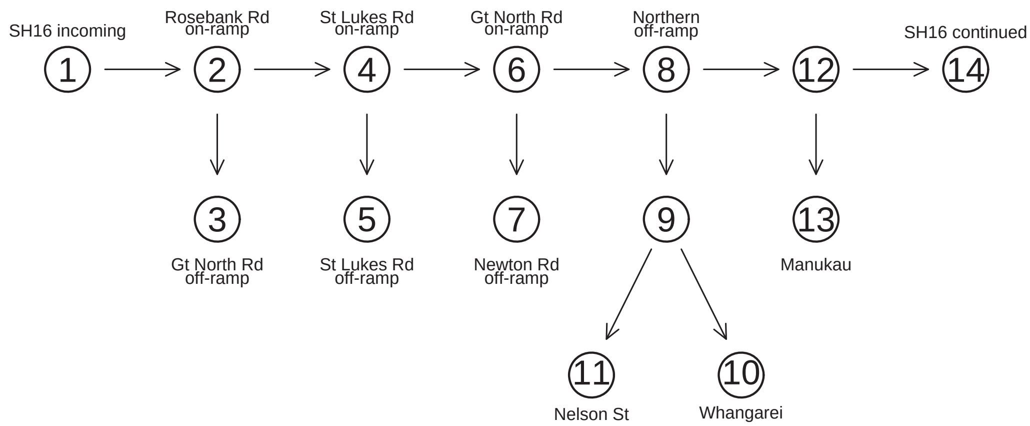


Figure 3.9: Section of State Highway 16, Auckland

As we may recall from the previous chapter, this traffic system is a tree network, another type of network with unique routing, ergo when implementing the logit route choice traffic demand model we need to incorporate dummy variables representing the option of non-travel. As described before we can then calculate the route choice probabilities using Equations 3.21, 3.5 and 3.7.

The counts were generated by independent Poisson distributions with mean values equal to traffic volumes as reported for this section of the highway by the New Zealand Transport Agency for the year 2010. The target of inference is again the logit parameter ψ with all other travel demand parameters assumed to be known. Once more we use a true value of $\psi = 1$ to simulate the link counts.

We carry out 100,000 MCMC iterations, with the first 20,000 discarded as burn-in samples. In Figure 3.10 we plot the estimated posterior densities for ψ for each of the four levels of the number of days worth of data assumed to be available. In each case the vague $\mathsf{Gam}(0.05, 0.1)$ prior density is also displayed (as a dashed line) and the true value of $\psi = 1$ via a vertical red line.

The results are very similar for both methods, with the dispersion of the posteriors reducing significantly as the numbers of days increases. This is to be expected as a result of the increase in information available from the data.

While the output from the simulations in Figure 3.10 show little to distinguish between the methods, compelling differences become apparent when we compare characteristics such as the acceptance rate. An analysis of the performance of the algorithms shows a far superior acceptance rate for the route sampling method developed on the basis of Hazelton (2010b)'s work in comparison to its alternative.

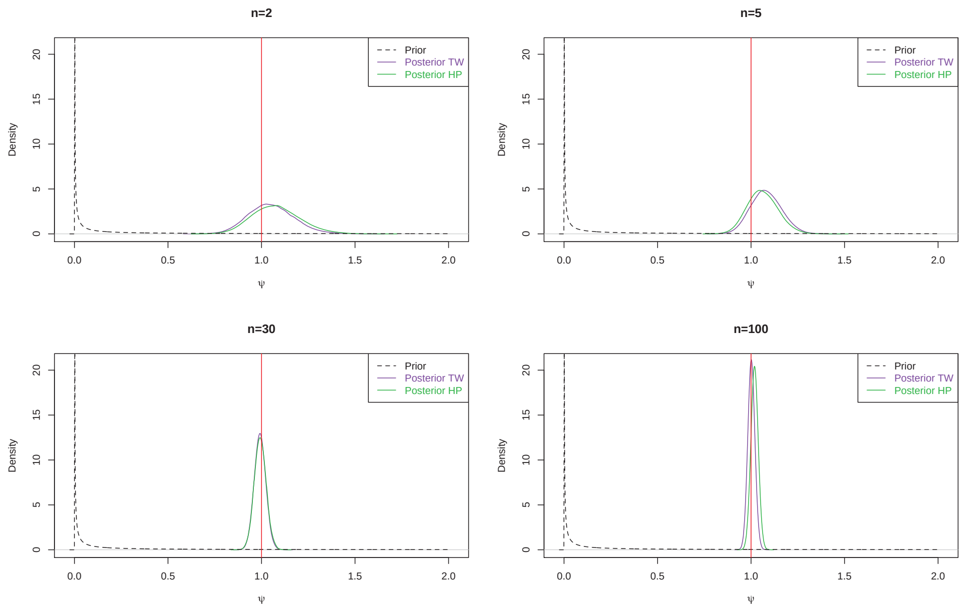


Figure 3.10: Posterior and prior density of demand parameter ψ for both methods in case of SH16 network.

Specifically, our sampler had a 31.4% update rate for elements of the route flow vectors, whereas the figure for Tebaldi and West (1998)'s sampler was 1.2%. We note that the higher acceptance rate is not obtained at the expense of slow mixing of the chain. On the contrary, the mixing is very quick, because we use a blocked independence sampler Gamerman and Lopes (2006). Another important benefit of sampling candidate route flows by constructing them as outcomes from a first-order Markov process is that the MCMC algorithm ran about 25% faster than when using Tebaldi and West (1998)'s method.

3.5 Conclusion

Models that describe the day-to-day dynamics of a traffic network are typically contain many parameters. Some of the most interesting and important parameters in such models relate directly to the manner in which travellers use past experiences to determine future travel choices. Reliable estimation of such behavioural parameters is essential, but also very challenging.

In principle, information about these parameters is available through the subtle changes in traffic flow patterns observed from day-to-day. For example correlation structures, where certain links share similar demand levels, can provide useful details.

Bayesian statistical models are able to incorporate characteristics such as second-order properties naturally into the estimation process and this ability can be taken advantage of to an even greater extent when link counts from successive days are available, as we can now observe dependencies between days as well as between links in the network.

In this chapter we adapt the MCMC method for sampling route flows that was introduced in chapter 2, to day-to-day traffic models, where the set of link counts are observed on consecutive days. We show that the route flow vectors $\boldsymbol{y}$ are conditionally independent given the observed link counts $\boldsymbol{x}$ and parameter vector $\boldsymbol{\theta}$, which means that we can separately sample the route flow candidates for each day. This greatly increases the acceptance rate, which in turn reduces the computational demand of the MCMC algorithm considerably.

We illustrate how well the MCMC algorithm based on the route sampling method we developed in the previous section for tree networks (as an extension of Hazelton (2010b)'s work on linear networks) operates for a toy example. Attempts to accurately reconstruct the route flows deliver promising results, we find that the modes of the posteriors often match the true route flow values. This is relevant with respect to the estimation of other parameters in the model, as knowledge of the latent variable $\boldsymbol{y}$ is essential for determining the values of $\boldsymbol{\theta}$.

This is confirmed when we then attempt to use the route flow information obtained from the MCMC sampler to estimate the parameters of the truncated linear traffic demand model in Equation 3.20. The estimated posterior densities of the demand parameters show a reasonable degree of accuracy considering they were based on the estimated route flows. This is reassuring as implementation of day-to-day dynamic models rely heavily on good estimates for these parameter values.

As a next step, we consider all parameters to be unknown apart from the logit model parameter ψ and concentrate on comparing this method with the MCMC algorithm based on an alternative route sampling procedure developed by Tebaldi and West (1998). We find that both methods perform similarly well, which is to be expected given the large number of MCMC iterations used.

The requirement for sufficient data in order to counter the influence of the prior is evident, however, this effect was particularly strong as the system considered is very small and hence little information can be extracted from second-order properties such as correlation structures.

A significant finding from our numerical experiments was the extent to which our extension of Hazelton (2010b)'s route flow sampler out-performed Tebaldi and West (1998)'s algorithm, especially with respect to the acceptance rate. A higher acceptance rate means a faster convergence rate of the MCMC sampler, ergo less iterations will be needed to get the same level of accuracy in the resulting estimates.

However, we note that there are two important caveats. Firstly, the examples are both tree networks, a type of network system for which the algorithm was specially designed. At present, when dealing with more general network topologies, there is no alternative to Tebaldi and West (1998)'s method. The second drawback is that even though our method is a significant improvement, the MCMC approach still requires a high computational cost in running the analysis.

Nonetheless, we have made considerable progress in this chapter in the development of models of the day-to-day evolution of road traffic networks. We have developed an inferential methodology for day-to-day data that is very general, and can certainly be applied to more heavily studied problems such as OD matrix estimation.

Chapter 4

Incorporating Partial Routing Information

There is a fine line between serendipity and stalking.

(David Coleman)

4.1 Introduction

We have discussed the estimation of OD matrices with data from vehicles observed on links in the network of interest, for example with the use of inductive loop detectors which collect data as cars pass over them. The primary challenge in this line of statistical inference are the identifiability problems that arise because of the lack of information in this kind of data. In this chapter we investigate another approach to the OD matrix estimation problem, namely, to supplement the link counts with data from another source.

In previous work, Watling (1994) developed a maximum likelihood estimator of the average route flow rates based on the usage of registration plate scanning, a method which takes advantage of the emergence of roadways with installed electronic toll systems.

Problematically, the model estimates were shown to be biased due to the problem of incomplete trips. That is, not all vehicles whose plates were registered will complete the journey within the observational period, an occurrence which introduces bias to all methods that rely on tracking vehicles at selected locations in the network only.

Castillo et al. (2008a) points out that the presence of automated license plate scanners on almost all links in the network would deliver data that would contribute to very accurate estimates. However, due to budget constraints this is currently not possible and the number of links with cameras positioned on them is limited (Castillo et al., 2010).

The matter at hand is that, resource limitations aside, if we had perfect routing information and thus were able to monitor the movements for all individual vehicles from the moment they enter the network until they exit, estimation of the OD-matrix would become trivial. At present such information is not available, however, phones equipped with a Global Navigation Satellite System (GLONASS) or, more commonly in New Zealand, a Global Positioning System (GPS), as well as other types of automatic location technology are becoming increasingly popular. This development is giving rise to the potential of providing a more sophisticated coverage of transportation networks than link counts using current sensor technology or license plate recognition does.

In particular, measurements from GPS devices are very economical to collect and can therefore be re-sampled more easily. Furthermore data compilation from GPS-equipped vehicles has been shown to be similarly resistant to error as current dual magnetic loop detectors, Bar-Gera (2007). This should improve OD estimation considerably, given that as Bell (1991) points out, the quality of the estimation depends on the quality of the prior information.

Utilising vehicle flows from tracked vehicles as a new source of traffic data creates a statistical challenge for OD matrix estimation, because only a small proportion of all vehicles surveyed will currently be using these new technologies (Herrera et al., 2010). And we can realistically assume that only a percentage of the vehicles known to be equipped with GPS will be travelling on any given day, introducing a binomial variation into the estimates of average flows. The number of tracked vehicles in total can be expected to be only a small subset of all data available in the near future.

One complication that arises from only a small number of vehicles being tracked via GPS is that the probability of being tracked is a nuisance parameter that also needs to be estimated. In the most extreme case possible a separate rate of GPS penetration could be associated with each route, yielding extra information given by the GPS data redundant due to overparametrisation.

In the long term, we may find ourselves facing the opposite problem, where we gain access to enough data from tracked vehicles that the usefulness of the information from the traditional link count data becomes questionable.

We investigate the possibility that the information given by link count data could become redundant if enough routing information is available or, as suspected by Westerman et al. (1996), the most accurate estimation can only be achieved through the use of both sources of information.

When making inference, we may consider incorporating the additional information by combining the counts from the tracked vehicles with the link counts via a suitable prior. This would be similar to Maher (1983)'s Bayesian approach in which route flow information gathered in a previous year's roadside survey is used as a prior. The difficulty with this approach is that the data may be collected contemporaneously, in which case, they will not be independent.

The focus of this chapter will be to address the issue of combining these two sources of information using a formal likelihood based approach to inference for static OD matrices. To this end we develop a statistical model to describe the joint distribution of link and routing data.

We also develop two variants. The first is the standard normal approximation model, where the variance-covariance matrix is a diagonal matrix with the mean vector as elements, a critical characteristic for the purposes of OD matrix estimation (Castro et al., 2004). The second adaptation is a simplification of the first where there is no longer a relationship between mean and variance.

We examine the likelihoods of these two models, as well as the corresponding models based on link count data only. The likelihood analysis also illustrates a variety of standard and non-standard properties which leads to a more general theoretical analysis of the model likelihoods. Initially, the inference is based on likelihoods where the probability of vehicle tracking (i.e. the penetration rate) is assumed to be known. In practice this will not be the case, although we may have some exogenous estimates.

Later on, we discuss the estimation of this nuisance parameter and outline the difficulties we encounter when allowing the probabilities to vary from route to route. As a starting point, we impose a structure on the penetration rate of tracked vehicles. In the case that we have a rate of vehicle tracking that is homogeneous across the network, we show that a method of moments estimator for the probability of vehicle tracking performs well. In addition, we investigate the consequences of mis-specifying the model for this probability (for example, assuming that the rate is constant when in truth it varies between routes).

4.2 Likelihood-based inference

As before in the traditional modelling framework, we have the link counts $\boldsymbol{x}$ and the route flows, $\boldsymbol{y}$, which, although ultimately of interest, are usually not directly observed. However, we now imagine that a fraction of vehicles deliver routing information directly, $\boldsymbol{y}_{gps} = (y_{gps,1}, \ldots, y_{gps,N})^{\mathsf{T}}$, where N is the number of routes in the network. We also assume that each vehicle on route r has probability p_r of being tracked, with a corresponding vector of probabilities $\boldsymbol{p} = (p_1, \ldots, p_N)^{\mathsf{T}}$.

Conditional on the overall number of route flows, the tracked route flows are binomially distributed with $\boldsymbol{y}_{gps} \,|\, \boldsymbol{y} \sim \mathsf{Binom}(\boldsymbol{y}, \boldsymbol{p})$.

Due to the fact that we consider the total route flows, including both tracked and non-tracked vehicles, as Poisson distributed with mean rate vector $\boldsymbol{\lambda}$, we consider the unconditional distribution of the tracked route flows as a thinned Poisson process. That is, each time we observe a vehicle on route r it can either be classified as being tracked with probability p_r or categorised as non-tracked with probability $1 - p_r$. We define a diagonal matrix $\boldsymbol{P} = diag(\boldsymbol{p})$ with the tracking probabilities as elements and a diagonal matrix with the probabilities of not being tracked as $\boldsymbol{Q} = \boldsymbol{I} - \boldsymbol{P}$ with $\boldsymbol{I}$ being the appropriately dimensioned identity matrix. The matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ are diagonal matrices because the routes are independent of each other. Under these settings, the tracked vehicle flows are randomly $\mathsf{Pois}(\boldsymbol{P\lambda})$ distributed and the non-tracked vehicles follow a $\mathsf{Pois}(\boldsymbol{Q\lambda})$ distribution in accordance to standard Poisson thinning arguments.

If the route flows of all vehicles are large enough we can approximate their distribution by the normal distribution as follows:

$$\boldsymbol{y} \sim \mathsf{Norm}(\boldsymbol{\lambda}, \Sigma), \tag{4.1}$$

in which case the unconditional distribution of the tracked route flows is

$$\boldsymbol{y}_{gps} \sim \mathsf{Norm}(\boldsymbol{P\lambda}, \boldsymbol{P}\Lambda). \tag{4.2}$$

According to Equation 4.1 we can apply a normal approximation to the link counts as well, that is

$$\boldsymbol{x} \sim \mathsf{Norm}(\boldsymbol{A\lambda}, \boldsymbol{A}\Sigma\boldsymbol{A}^{\mathsf{T}}), \tag{4.3}$$

because the link flows are related to the route flows as detailed in Equation 1.1.

The rationale for assuming we are dealing with high demand, where the normal approximation holds, is that otherwise in the low demand case, where Poisson and similar models are necessary, there will be too small a chance of observing GPS equipped vehicles for the kind of analysis we examine here to be possible. In addition, using a normal approximation offers greater computational efficiency, which improves the ability to compute estimates for all kinds of networks, including large and more complicated ones.

We can develop a likelihood individually for each of the two sources of data, the first representing the link count information, while the second relates how likely certain average flows are given the routing information.

These two likelihoods can then be multiplied with each other, where we can consider the information from the tracked vehicles as an update of the link count data likelihood. This factorisation of the likelihood is only valid, however, if the two elements are statistically independent of each other.

This may not be the case if in the data collection process the link counts include both the tracked and the non-tracked vehicles. Since we assume here that they are observed contemporaneously, we cannot consider the link counts $\boldsymbol{x}$ and monitored route counts $\boldsymbol{y}_{gps}$ to be independent.

We recommend separating the two sources of information by decomposing the link counts into $\boldsymbol{x} = \boldsymbol{x}_{gps} + \boldsymbol{x}_{not}$ where $\boldsymbol{x}_{gps} = \boldsymbol{A}\boldsymbol{y}_{gps}$ is the contribution to the link counts from tracked vehicles, and $\boldsymbol{x}_{not}$ is the contribution from those vehicles that are not tracked. The vectors $\boldsymbol{y}_{gps}$ and $\boldsymbol{x}_{not}$ are independent under standard assumptions.

We know the distribution of the tracked route flows from Equation 4.2, while the corresponding result for $\boldsymbol{x}_{not}$ is

$$\boldsymbol{x}_{not} \sim \mathsf{Norm}(\boldsymbol{AQ\lambda}, \boldsymbol{AQ}\Lambda\boldsymbol{A}^{\mathsf{T}}). \tag{4.4}$$

Thus, the updated likelihood has the form

$$\mathsf{L}(\boldsymbol{\lambda}, \boldsymbol{p}) = f(\boldsymbol{x}_{not} \,|\, \boldsymbol{\lambda}, \boldsymbol{p}) \cdot f(\boldsymbol{y}_{gps} \,|\, \boldsymbol{\lambda}, \boldsymbol{p}) \tag{4.5}$$

where f generically denotes probability mass functions. Equation 4.5 emphasises the fact that the vector $\boldsymbol{p}$ of tracking probabilities is a nuisance parameter that will generally need estimating in tandem with $\boldsymbol{\lambda}$. We assume for now that the probability of being tracked is known and will return to its inference at a later point.

Both the normal models defined in Equations 4.2 and 4.4 have a functional relationship between the mean and the variance, that is, the variance is a function of the mean. While the dependence of the covariance matrices on $\boldsymbol{\lambda}$ can provide important information (Hazelton, 2003) it also leads to a more complex log-likelihood. A simplification of the likelihood is to assume that the covariance matrix of $\boldsymbol{y}$ is fixed; i.e. $\mathsf{Var}(\boldsymbol{y}) = \Sigma = \mathsf{diag}(\boldsymbol{\sigma})$ not dependent on $\boldsymbol{\lambda}$.

We then get the corresponding simplified models defined by

$$\boldsymbol{y}_{gps} \sim \mathsf{Norm}(\boldsymbol{P\lambda}, \boldsymbol{P}\Sigma) \tag{4.6}$$

and

$$\boldsymbol{x}_{not} \sim \mathsf{Norm}(\boldsymbol{AQ\lambda}, \boldsymbol{AQ}\Sigma\boldsymbol{A}^{\mathsf{T}}). \tag{4.7}$$

We examine the likelihood theory for both the original 4.2/4.4 and simplified 4.6/4.7 models in the next two sections. One reason for looking at both scenarios is that the computational needs for obtaining estimates for the simple model are considerably less than in the case of the original model. Thus, it may be more practical, especially with very large networks, to use the simplified version.

A particular line of interest in our likelihood analysis is obtaining large sample approximations to the properties of estimators of the mean route flows. The asymptotic process that we have in mind is an increase in the length of the observation period, $\hbar$, so that the elements of $\boldsymbol{\lambda}$, and hence the likely vehicle counts, become larger.

In other words, we set $\boldsymbol{\lambda} \equiv \boldsymbol{\lambda}(\hbar) = \hbar\boldsymbol{\lambda}_0$ where $\boldsymbol{\lambda}_0$ is the vector of mean route flows per unit time (e.g. an hour), and then consider the asymptotic process $\hbar \to \infty$. We assume throughout that $\hbar$ is known so that we can switch between estimation of $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_0$ by application of this scaling factor.

The distinction between these parameters is important when we discuss the asymptotic likelihood theory, since then we will require a fixed (non-asymptotic) parameter $\boldsymbol{\lambda}_0$ rather than the 'moving target' $\boldsymbol{\lambda}$. For this reason Λ_0 is defined to be $\mathsf{diag}(\boldsymbol{\lambda}_0)$, as well as $\Sigma = \hbar\Sigma_0$ for the simplified model, so that the scale of these dispersion matrices match that of the data.

4.3 Analysis of likelihoods

We first visualise the likelihood functions under the different models/approximations for a simple linear network, seen in Figure 4.1.

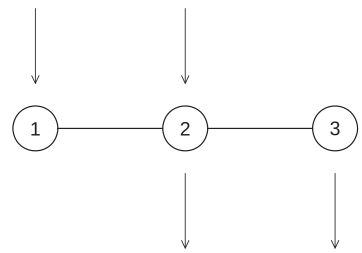


Figure 4.1: Linear network with only three OD pairs.

There are three OD pairs, where OD pairs 1,2 and 3 correspond to trips from node 1 to 2, 1 to 3, and 2 to 3, respectively. As this is a network with unique routing, we have the same number of routes. In the following plots, we assume the parameter vector $\boldsymbol{\lambda}$ of interest are the average rates of traffic flow on each route, with only three elements, λ_1, λ_2 and λ_3. The link-path incidence matrix is given by

$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}.$$

We observe data $\boldsymbol{x} = (10, 10)^{\mathsf{T}}$ and $\boldsymbol{y}_{gps} = (1, 1, 1)^{\mathsf{T}}$, so that $\boldsymbol{x}_{not} = (8, 8)^{\mathsf{T}}$. The proportion of tracked vehicles is set to be a constant 10% across all the network, $\boldsymbol{p} = (0.1, 0.1, 0.1)^{\mathsf{T}}$. The choice of these numeric values make no qualitative difference to the following results.

The small number of parameters allows us to examine plots of the likelihood, which in turn help us to perceive changes due to modifications in the various sources of travel information. We cannot display plots of the full likelihood since it is a function of a three-dimensional vector, however, we can find the likelihood which represents the values of λ_1 and λ_2 for which the likelihood is maximised with respect to a fixed value of λ_3:

$$\mathsf{L}(\lambda_1, \lambda_2) = \max_{\lambda_3} \mathsf{L}(\boldsymbol{\lambda}).$$

This likelihood is called the *profile likelihood* for parameters λ_1 and λ_2.

For each model, apart from getting a first visual impression of the impact of changes in the likelihood functions for changes in the level of tracking information available, we develop a mathematical description to give insight on the theoretical foundation on which these changes are based. We adopt the same vector calculus notation as used in Wand (2002).

The ML estimator of a parameter vector $\boldsymbol{\theta}$ is defined by $\hat{\boldsymbol{\theta}} = \arg\max \mathsf{L}(\boldsymbol{\theta})$. Assuming that $\boldsymbol{\theta}$ does not lie at a boundary of the parameter space, we work out its estimate by differentiating the likelihood, where the resulting vector of derivatives is called the score vector $\boldsymbol{D}$, and solving the first order conditions. Under certain regularity conditions this estimator is asymptotically optimal, in that it has minimum variance amongst asymptotically unbiased estimators. Moreover the distribution of the ML estimator tends, as the sample size increases, towards a Gaussian distribution centred on the true parameter vector $\boldsymbol{\theta}$,

$$\hat{\boldsymbol{\theta}} \sim \mathsf{Norm}(\boldsymbol{\theta}, \mathcal{I}^{-1}),$$

where $\mathcal{I} \equiv \mathcal{I}(\boldsymbol{\theta}) = \mathsf{E}[\mathcal{J}(\boldsymbol{\theta})]$ is the expected Fisher information matrix and $\mathcal{J}$ is the observed information matrix. The observed Fisher information matrix is defined as minus the expectation of the Hessian matrix, the matrix of second derivatives of the likelihood. The inverse of $\mathcal{J} = -\boldsymbol{H}_{\boldsymbol{\theta}}\mathsf{L}(\boldsymbol{\theta})$ relates to the variability associated with the ML estimates. Overall this asymptotic distribution provides a natural approximation for finite samples, so that one can use

$$\hat{\boldsymbol{\theta}} - \boldsymbol{\theta} \mathrel{\dot{\sim}} \mathsf{Norm}(0, \mathcal{I}^{-1})$$

to construct standard errors, confidence intervals and statistical tests, evaluating the information matrix at $\hat{\boldsymbol{\theta}}$ where necessary.

We assume for now that the aforementioned regularity conditions hold and determine the ML estimator, $\hat{\boldsymbol{\lambda}}$, for each model by solving the normal equations $\boldsymbol{D}_{\lambda}\mathsf{L}(\boldsymbol{\lambda}) = 0$ for the parameter $\boldsymbol{\lambda}$. We also attempt to determine the corresponding expected Fisher information matrices.

4.3.1 Simplified link count model

The log-likelihood for the simple version of the traditional model, which relies on link flow rates alone, with constant route flow covariance matrix $\Sigma = \mathsf{diag}((10, 10, 10)^{\mathsf{T}})$ is

$$\mathsf{L}(\boldsymbol{\lambda}) = \underbrace{-\tfrac{1}{2}\log(|\boldsymbol{A}\Sigma\boldsymbol{A}^{\mathsf{T}}|)}_{\text{constant}} - \underbrace{\tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{A}\boldsymbol{\lambda})^{\mathsf{T}}(\boldsymbol{A}\Sigma\boldsymbol{A}^{\mathsf{T}})^{-1}(\boldsymbol{x} - \boldsymbol{A}\boldsymbol{\lambda})}_{\text{becomes 0 when x=Ay}}. \tag{4.8}$$

This is a quadratic form in $\boldsymbol{\lambda}$. The on-diagonal elements are constant, as the covariance is structurally independent of the values of $\boldsymbol{\lambda}$.

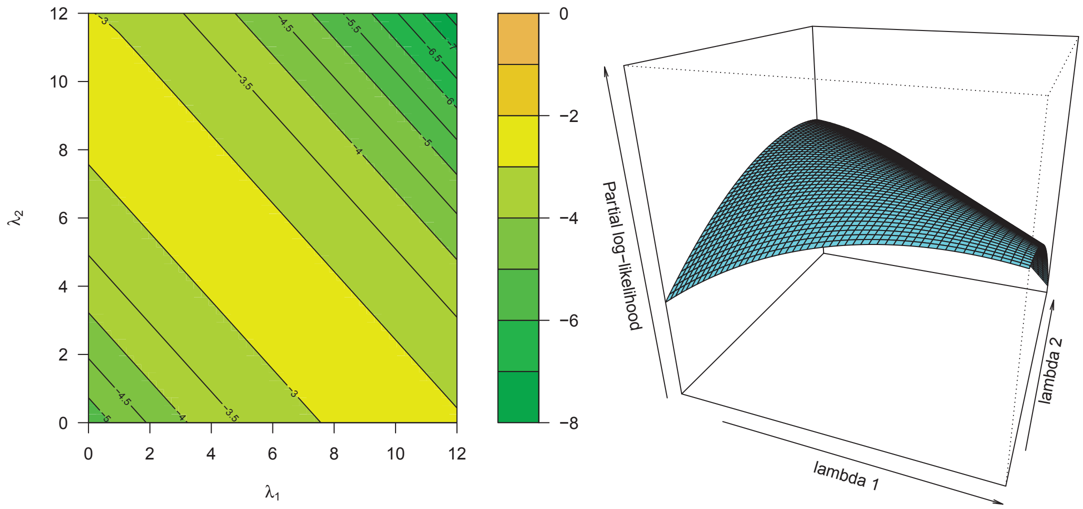


Figure 4.2: Contour and perspective plots of the profile log-likelihood for simplified model using link count data only.

The identifiability problems for $\boldsymbol{\lambda}$ become clear in Figure 4.2. The (profile) log-likelihood is very flat at its peak, with a ridge running along the diagonal. This makes maximum likelihood estimation problematic as there is no global maximum.

The calculation of the derivative vector, that is, the gradient of the likelihood function $\mathsf{L}(\boldsymbol{\lambda})$ is straightforward in this case:

$$\boldsymbol{D}_{\lambda}\mathsf{L}(\boldsymbol{\lambda}) = -\tfrac{1}{2}\cdot 2\cdot(\boldsymbol{x}-\boldsymbol{A}\boldsymbol{\lambda})^{\mathsf{T}}(\boldsymbol{A}\Sigma\boldsymbol{A}^{\mathsf{T}})^{-1}\cdot(-\boldsymbol{A}) = (\boldsymbol{x}-\boldsymbol{A}\boldsymbol{\lambda})^{\mathsf{T}}(\boldsymbol{A}\Sigma\boldsymbol{A}^{\mathsf{T}})^{-1}\boldsymbol{A}.$$

Setting the score vector $\boldsymbol{D}_{\lambda}\mathsf{L}(\boldsymbol{\lambda})$ equal to zero and solving for $\boldsymbol{\lambda}$ gives us $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{\lambda}$. It will generally not have a unique solution because this linear system is underdetermined. This confirms what we can see in Figure 4.2, namely that we get a whole set of possible maximum values, namely any $\boldsymbol{\lambda}$ for which $\boldsymbol{\lambda} = \boldsymbol{x}$. In addition, when we calculate the Hessian matrix,

$$\boldsymbol{H}_{\lambda}\mathsf{L}(\boldsymbol{\lambda}) = -\boldsymbol{A}^{\mathsf{T}}(\boldsymbol{A}\Sigma\boldsymbol{A}^{\mathsf{T}})^{-1}\boldsymbol{A},$$

we can evaluate the expected Fisher information matrix

$$\mathcal{I} = -\mathsf{E}(\boldsymbol{H}_{\lambda}\mathsf{L}(\boldsymbol{\lambda})) = \boldsymbol{A}^{\mathsf{T}}(\boldsymbol{A}\Sigma\boldsymbol{A}^{\mathsf{T}})^{-1}\boldsymbol{A}$$

which is always singular. This can be algebraically shown for our simple example.

For our network with routing matrix $\boldsymbol{A} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ and general $\Sigma = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix}$,

we find that the determinant of the matrix

$$\boldsymbol{A}^{\mathsf{T}}(\boldsymbol{A}\Sigma\boldsymbol{A}^{\mathsf{T}})^{-1}\boldsymbol{A} = \frac{1}{ab+bc+ac}\begin{pmatrix} b+c & -c & b \\ -c & a+c & a \\ b & a & a+b \end{pmatrix}$$

is equal to zero for arbitrary values of a, b and c. In other words, the information matrix is not invertible, ergo we are dealing with infinite standard errors. This means that the calculated values of the parameters are unreliable, which is to be expected as the ML estimates are not even uniquely defined in this case.

4.3.2 Normal link count model

The log-likelihood of the model based on link count data only with variance-covariance matrix Λ is given by

$$\mathsf{L}(\boldsymbol{\lambda}) = \underbrace{-\tfrac{1}{2}\log(|\boldsymbol{A}\Lambda\boldsymbol{A}^{\mathsf{T}}|)}_{\text{dependent on } \boldsymbol{\lambda}} - \tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{A}\boldsymbol{\lambda})^{\mathsf{T}}(\boldsymbol{A}\Lambda\boldsymbol{A}^{\mathsf{T}})^{-1}(\boldsymbol{x} - \boldsymbol{A}\boldsymbol{\lambda}) \tag{4.9}$$

where the variance is now a function of $\boldsymbol{\lambda}$, $\Lambda = \mathsf{diag}(\boldsymbol{\lambda})$. That is, the first part of the log-likelihood is no longer constant as it contains $\boldsymbol{\lambda}$ and, because it thus also contributes to the derivative, it needs to be maximised too.

We observe from the plots of the profile log-likelihood $\mathsf{L}(\lambda_1, \lambda_2)$ for our example in Figure 4.3 that taking account of the link between mean and variance has introduced curvature along the flat ridges that we saw previously in Figure 4.2. This shows that even before the inclusion of routing information, the likelihood estimation is using correlation inherent in the link count data as extra support. Correlation arises when motorists using one link all then go on to use another link, something that would be picked up in the link counts because of the matching corresponding link usage rates.

The profile likelihood in Figure 4.3 also shows that accounting for the functional relationship between mean and variance has resulted in some extreme behaviour at the boundaries of parameter space. We will later comment further on the circumstances that led to these kind of unusual shaped likelihood functions.

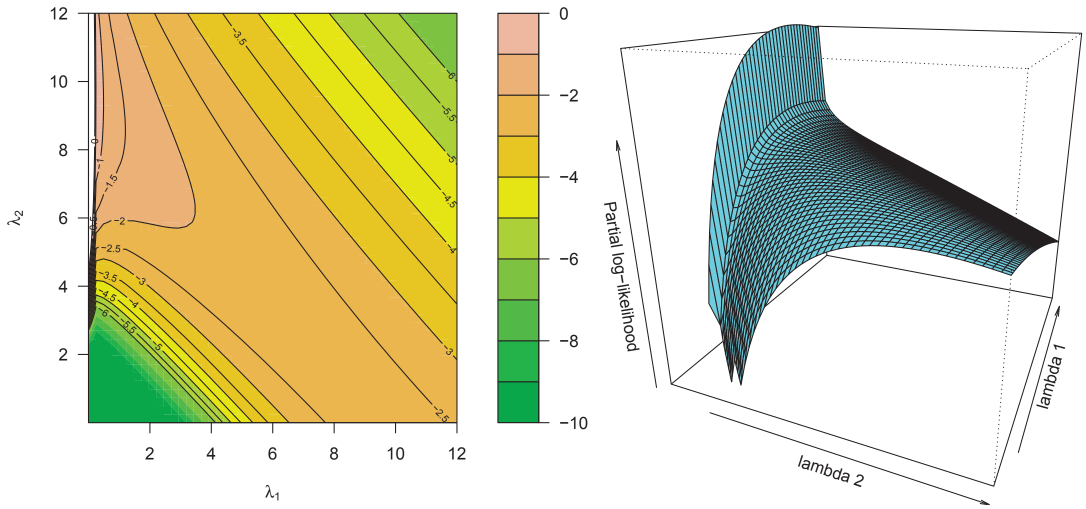


Figure 4.3: Contour and perspective plots of the profile log-likelihood for the normal model using link count data only.

When attempting to write down the score function $\boldsymbol{D}_\lambda \mathsf{L}(\boldsymbol{\lambda})$ for this model we find that the dependence of the covariance-variance matrix on the parameter $\boldsymbol{\lambda}$ complicates finding an explicit expression for the score vector considerably. An uncomplicated form of the score function is not available. However, evaluating the score function is nonessential here, because the MLE lies at the boundary in this case, and hence does not solve the normal equations. In addition, there is no need to evaluate the Hessian matrix $\boldsymbol{H}_\lambda \mathsf{L}(\boldsymbol{\lambda})$ either, because we cannot use the standard asymptotic approximations to provide measures of the precision of the MLE.

4.3.3 Simplified updated model

Next, we consider the simplified model using both link counts and routing information from the tracked vehicles. The log-likelihood is then

$$\begin{aligned} \mathsf{L}(\boldsymbol{\lambda}) \quad = \quad & -\tfrac{1}{2}\log(|\boldsymbol{A}\boldsymbol{Q}\Sigma\boldsymbol{A}^\mathsf{T}|) - \tfrac{1}{2}(\boldsymbol{x}_{not} - \boldsymbol{A}\boldsymbol{Q}\boldsymbol{\lambda})^\mathsf{T}(\boldsymbol{A}\boldsymbol{Q}\Sigma\boldsymbol{A}^\mathsf{T})^{-1}(\boldsymbol{x}_{not} - \boldsymbol{A}\boldsymbol{Q}\boldsymbol{\lambda}) \\ & -\tfrac{1}{2}\log(|\boldsymbol{P}\Sigma|) - \tfrac{1}{2}(\boldsymbol{y}_{gps} - \boldsymbol{P}\boldsymbol{\lambda})^\mathsf{T}(\boldsymbol{P}\Sigma)^{-1}(\boldsymbol{y}_{gps} - \boldsymbol{P}\boldsymbol{\lambda}). \qquad (4.10) \end{aligned}$$

This is another quadratic form in $\boldsymbol{\lambda}$, but the third and fourth terms (derived from the contribution of the routing data) mean that it will not be rank deficient, assuming that all the elements of $\boldsymbol{p}$ are non-zero.

As seen in Figure 4.4, even a small amount of routing data has generated enough curvature along the ridges at the top of the surface to display a unique maximum.

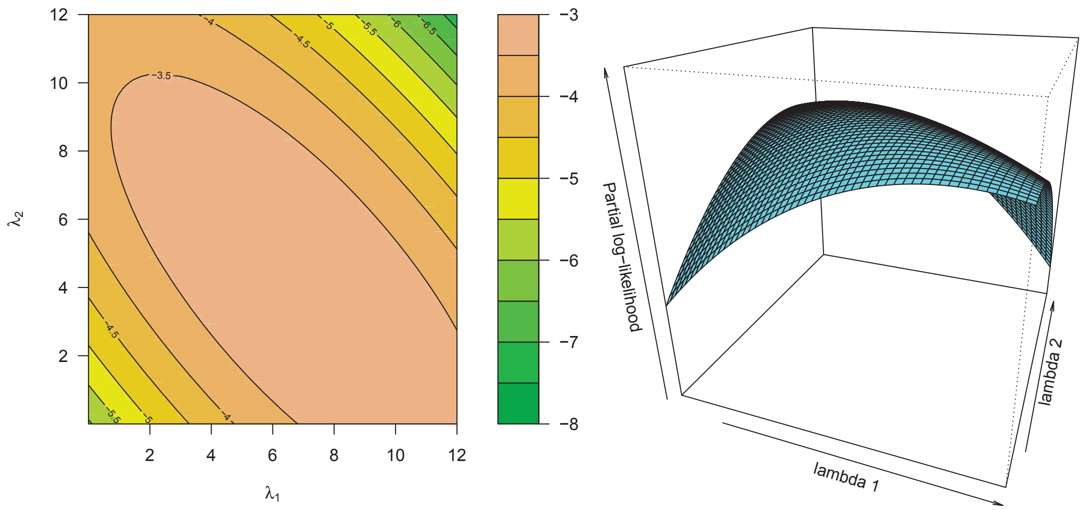


Figure 4.4: Contour and perspective plots of the profile log-likelihood for the simplified model using link counts and available routing data.

For the log-likelihood of the simplified model, the score vector is given by

$$\boldsymbol{D}_\lambda \mathsf{L}(\lambda) = (\boldsymbol{x}_{not} - \boldsymbol{AQ\lambda})^\mathsf{T}(\boldsymbol{AQ}\Sigma\boldsymbol{A}^\mathsf{T})^{-1}\boldsymbol{AQ} + (\boldsymbol{y}_{gps} - \boldsymbol{P\lambda})^\mathsf{T}\Sigma^{-1} \tag{4.11}$$

where we have assumed that the (known) probability vector $\boldsymbol{p}$ has no zero elements. Setting this equal to the zero vector and solving for $\boldsymbol{\lambda}$ gives the result

$$\hat{\boldsymbol{\lambda}} = \Big((\boldsymbol{AQ})^\mathsf{T}(\boldsymbol{AQ}\Sigma\boldsymbol{A}^\mathsf{T})^{-1}\boldsymbol{AQ} + \Sigma^{-1}\boldsymbol{P}\Big)^{-1}\Big((\boldsymbol{AQ})^\mathsf{T}(\boldsymbol{AQ}\Sigma\boldsymbol{A}^\mathsf{T})^{-1}\boldsymbol{x}_{not} + \Sigma^{-1}\boldsymbol{y}_{gps}\Big). \tag{4.12}$$

The general form of the maximum likelihood estimate vector $\hat{\boldsymbol{\lambda}}$ in Equation 4.12 matches that of the generalised least squares estimator developed by Cascetta (1984), with the tracked vehicle route counts playing the role of the target OD matrix in that earlier work.

We can also think of $\hat{\boldsymbol{\lambda}}$ as a type of ridge estimator (Hoerl and Kennard, 2000). The idea of ridge regression is rooted in multiple linear regression, which is analogous to the maximum likelihood estimation since we assume normality here. A problem often encountered in multiple linear regression is that the determinant of the variance-covariance matrix has a value of nearly 0.

This issue is described as near-collinearity and causes the model parameter estimates to become unstable due to the large variances and renders interpretations of them unreliable. As a matter of fact, as seen in section 4.3.1 exact collinearity occurs in the simplified link count mode, because at least one of the route flows is a linear combination of the other route flows. This relationship leads to the covariance matrix not being a full rank matrix anymore and its determinant is 0. In this case inverting the variance-covariance matrix is futile because the inverse does not exist. This is the reason why the information matrix was singular.

A solution to this kind of problem is to adopt ridge regression. The main disadvantage is that the ridge regression parameters become biased in contrast to the standard parameter estimates. Nevertheless, the variances of ridge regression estimators are decreased to the point that the MSE becomes smaller than that of the parameter estimates of the least squares/maximum likelihood model. In other words, the biased ridge regression estimator can outperform the unbiased maximum likelihood estimator provided its variance is small enough.

The aforementioned regularity conditions require that the likelihood has a unique and finite global maximum, and that this occurs at an interior point of the parameter space Ω. If we assume that $\boldsymbol{p}$ is fixed and known for the present, then the parameter space is $\Omega = \{\boldsymbol{\lambda} : \lambda_r \geq 0, r = 1, \ldots, N\}$, that is, the route flow rates are non-negative. However, a problematic feature of $\hat{\boldsymbol{\lambda}}$ is that it can take on negative values. For example, in the case that we observe $\boldsymbol{x}_{not} = (10, 8)^{\mathsf{T}}$ and $\boldsymbol{y}_{gps} = (2, 1, 3)^{\mathsf{T}}$ in our toy network, we would obtain $\hat{\boldsymbol{\lambda}} = (17.7, -4.7, 17.7)^{\mathsf{T}}$, based on known $p_r = 0.1$ and fixed variance $\sigma_r = 10$ for $r = 1, 2, 3$. The maximum likelihood estimator $\hat{\boldsymbol{\lambda}}$ on the natural (non-negative) parameter space Ω will lie on a boundary in such cases. It is for example $\hat{\boldsymbol{\lambda}} = (14.2, 0, 14.2)^{\mathsf{T}}$ for the data just described.

The algorithm described by Bell (1991) for computing generalised least squares estimates can be applied in such situations, where the objective function $\mathsf{L}(\boldsymbol{\lambda}, \boldsymbol{p})$ being minimised is

$$\tfrac{1}{2}(\boldsymbol{\lambda} - \boldsymbol{y}_{gps})^{\mathsf{T}} \boldsymbol{P} \Sigma^{-1} (\boldsymbol{\lambda} - \boldsymbol{y}_{gps}) + \tfrac{1}{2}(\boldsymbol{x}_{non} - \boldsymbol{A}\boldsymbol{\lambda})^{\mathsf{T}} (\boldsymbol{A}\boldsymbol{Q}\Sigma\boldsymbol{A}^{\mathsf{T}})^{-1} (\boldsymbol{x}_{non} - \boldsymbol{A}\boldsymbol{\lambda}),$$

where the elements of $\boldsymbol{\lambda}$ are constrained to be greater than zero, as negative traffic flows are impossible. This problem may be solved by forming the Lagrangian equation

$$\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\xi}) = \mathsf{L}(\boldsymbol{\lambda}, \boldsymbol{p}) - \boldsymbol{\xi}^{\mathsf{T}} \boldsymbol{\lambda}, \tag{4.13}$$

where $\boldsymbol{\xi}$ is a vector of Lagrangian multipliers.

The likelihood $\mathsf{L}(\boldsymbol{\lambda}, \boldsymbol{p})$ is a convex function whereas the constraint of nonnegativity is concave. Consequently, the necessary and sufficient condition of any solution is given by

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \mathbf{0} \tag{4.14}$$

as well as the complementary slackness conditions $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}} \leq \mathbf{0}$, $\boldsymbol{\xi} \geq \mathbf{0}$ and $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}} \boldsymbol{\xi} = 0$, where $\mathbf{0}$ is a vector of 0's.

After setting the first derivative of the Lagrangian Equation 4.13 equal to zero, as according to Equation 4.14, we can solve for $\boldsymbol{\lambda}$ to get:

$$\check{\boldsymbol{\lambda}} = \boldsymbol{D}^{-1} \cdot \left(\boldsymbol{P} \Sigma^{-1} \boldsymbol{y}_{yps} + \boldsymbol{A}^{\mathsf{T}} (\boldsymbol{A}\boldsymbol{Q}\Sigma\boldsymbol{A}^{\mathsf{T}})^{-1} \boldsymbol{x}_{non} + \boldsymbol{\xi} \right) \tag{4.15}$$

where we define $\boldsymbol{D} = \boldsymbol{P}\Sigma^{-1} + \boldsymbol{A}^{\mathsf{T}} (\boldsymbol{A}\boldsymbol{Q}\Sigma\boldsymbol{A}^{\mathsf{T}})^{-1} \boldsymbol{A}$. We can easily see the similarity to the expression in Equation 4.12:

$$\hat{\boldsymbol{\lambda}} = \Big((\boldsymbol{A}\boldsymbol{Q})^{\mathsf{T}} (\boldsymbol{A}\boldsymbol{Q}\Sigma\boldsymbol{A}^{\mathsf{T}})^{-1} \boldsymbol{A}\boldsymbol{Q} + \Sigma^{-1}\boldsymbol{P} \Big)^{-1} \Big((\boldsymbol{A}\boldsymbol{Q})^{\mathsf{T}} (\boldsymbol{A}\boldsymbol{Q}\Sigma\boldsymbol{A}^{\mathsf{T}})^{-1} \boldsymbol{x}_{not} + \Sigma^{-1} \boldsymbol{y}_{gps} \Big).$$

The problem is to find $\boldsymbol{\xi}$ such that condition 4.14 as well as the slackness conditions are fulfilled. The solution is to minimise the Lagrangian Equation 4.13, with respect to $\boldsymbol{\lambda}$ and maximise with respect to $\boldsymbol{\xi}$ (subject to $\boldsymbol{\xi} \geq \mathbf{0}$). In addition, we can use the estimate of $\boldsymbol{\lambda}$ in Equation 4.15 to derive that $\frac{\delta \boldsymbol{\lambda}}{\delta \xi} = \boldsymbol{D}^{-1}$, that is, we define $d_r = \frac{\partial \lambda_r}{\partial \xi_r} > 0$ as the elements on the diagonal of $\boldsymbol{D}^{-1}$. Hence the following algorithm solves the constrained GLS problem:

Step 1 (*Initialisation*)

Set $\check{\boldsymbol{\lambda}} = \mathbf{0}$ (unconstrained estimation)

Step 2 (*Iteration*)

For any $\check{\lambda}_r < 0$ calculate $\check{\lambda}_r$ from 4.15, then set

$$\xi_r = \begin{cases} \boldsymbol{\xi} - \frac{\check{\lambda}_r}{d_r}, & \text{if } \lambda_r < 0 \\ \max\left(0, \boldsymbol{\xi} - \frac{\check{\lambda}_r}{d_r}\right), & \text{otherwise,} \end{cases}$$

until convergence.

Nonetheless, we note that the results obtained by optimising over Ω can still be of questionable value as a consequence of relaxing the link between mean vector and covariance matrix in the simplified model. We may obtain zero estimates for mean route flows when the observed tracked counts for that route are non-zero.

In actual fact, we get a boundary estimate for the mean route flows, regardless of tracked vehicles being observed, for the data just introduced: $y_{gps,2} = 1$ but $\check{\lambda}_2 = 0$. The occurrence of $\check{\boldsymbol{\lambda}}$ on the boundary of Ω is a finite sample phenomenon if $p_r > 0$ and $\lambda_r > 0$ for all r. For sufficiently large sample sizes, which we obtain by allowing the observation window $\hbar$ to be sufficiently large, the MLE will be at an interior point, because the GLS estimator is asymptotically consistent. In other words, $\check{\boldsymbol{\lambda}}$ will converge arbitrarily close towards $\boldsymbol{\lambda}_0$ as the sample size increases. Moreover, this is true even though our simplified model is misspecified. That is, even if the data is generated by a Poisson process (or the full normal approximation thereof) we will still find that Bell (1991)'s GLS estimates based on the simplified model are consistent, though not asymptotically efficient.

It follows that when $\check{\boldsymbol{\lambda}}$ is interior to Ω, and the sample size is at least moderately large, then a normal approximation to the sampling distribution of $\check{\boldsymbol{\lambda}}$ is reasonable. More specifically, the distribution of $\check{\boldsymbol{\lambda}}_0$ will be approximately $\mathsf{Norm}(\boldsymbol{\lambda}_0, \mathcal{I}_{\boldsymbol{\lambda}_0}^{-1})$.

Here $\mathcal{I}_{\boldsymbol{\lambda}_0}$ is the expected information matrix for $\boldsymbol{\lambda}_0$ in the simplified model using both link counts and routing information, given by

$$\mathcal{I}_{\boldsymbol{\lambda}_0} = -\mathsf{E}[\boldsymbol{H}_{\boldsymbol{\lambda}_0}\mathsf{L}(\boldsymbol{\lambda}_0)] = \hbar\left(\boldsymbol{p}^\mathsf{T}\Sigma_0^{-1}\mathbf{1} + \boldsymbol{q}^\mathsf{T}\boldsymbol{A}^\mathsf{T}(\boldsymbol{A}\boldsymbol{Q}\Sigma_0\boldsymbol{A}^\mathsf{T})^{-1}\boldsymbol{A}\boldsymbol{q}\right). \tag{4.16}$$

4.3.4 Normal updated model

Finally we examine the normal model based on both link counts and the available routing data. The log-likelihood in this case is

$$\begin{aligned}\mathsf{L}(\boldsymbol{\lambda}) = \; & -\tfrac{1}{2}\log(|\boldsymbol{A}\boldsymbol{Q}\Lambda\boldsymbol{A}^\mathsf{T}|) - \tfrac{1}{2}(\boldsymbol{x}_{not} - \boldsymbol{A}\boldsymbol{Q}\boldsymbol{\lambda})^\mathsf{T}(\boldsymbol{A}\boldsymbol{Q}\Lambda\boldsymbol{A}^\mathsf{T})^{-1}(\boldsymbol{x}_{not} - \boldsymbol{A}\boldsymbol{Q}\boldsymbol{\lambda}) \\ & -\tfrac{1}{2}\log(|\boldsymbol{P}\Lambda|) - \tfrac{1}{2}(\boldsymbol{y}_{gps} - \boldsymbol{P}\boldsymbol{\lambda})^\mathsf{T}(\boldsymbol{P}\Lambda)^{-1}(\boldsymbol{y}_{gps} - \boldsymbol{P}\boldsymbol{\lambda}).\end{aligned} \tag{4.17}$$

The profile log-likelihood for our example is shown in Figure 4.5. Once again we see a clear unique maximum. However, the overall appearance of the profile likelihood is markedly different to the previous one in Figure 4.4, largely due to the extreme behaviour of the function at the boundaries. The differences are a consequence of the variance-covariance matrices now being diagonal matrices of the corresponding mean vectors of x_{non} and y_{gps}, as this is the only dissimilarity between Equations 4.12 and 4.17.

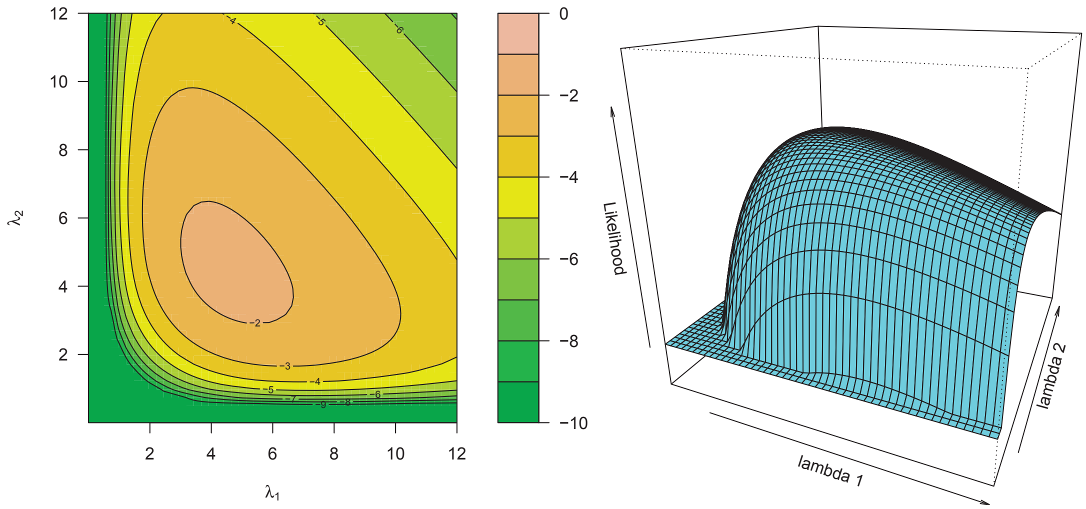


Figure 4.5: Contour and perspective plots of the profile log-likelihood for the normal model using link counts and available routing data.

The dependency of the dispersion matrix Λ on $\boldsymbol{\lambda}$ makes calculating the score vector in order to find the ML estimator $\hat{\boldsymbol{\lambda}}$ difficult, as discussed previously in section 4.3.2. Instead of attempting to find an explicit expression, we calculate $\hat{\boldsymbol{\lambda}}$ by numerical optimisation. Although we cannot determine the estimated average route flows directly, we can provide the circumstances under which this estimate can lie on a boundary, as outlined in the theorem below.

Theorem:
Let $\mathcal{Y}(\boldsymbol{x})_r = \{\boldsymbol{y}_{not} : \boldsymbol{x}_{not} = \boldsymbol{A}\boldsymbol{y}_{not} \text{ and } y_{not,r} = 0\}$ be the set of feasible route flow vectors for the non-tracked vehicles with a zero at the rth entry. Then for the normal model log-likelihood in Equation 4.17, $\hat{\boldsymbol{\lambda}}$ will lie on the $\lambda_r = 0$ boundary of Ω if and only if $y_{gps,r} = 0$ and $\mathcal{Y}(\boldsymbol{x})_r \neq \emptyset$.

Proof:
If $y_{gps,r} = 0$ and $\mathcal{Y}(\boldsymbol{x})_r \neq \emptyset$ then the coset $\mathcal{Y}(\boldsymbol{x})'_r = \boldsymbol{y}_{gps} + \mathcal{Y}(\boldsymbol{x})_r$ is not empty, and contains only vectors with zero rth element. For any finite vector $\boldsymbol{\lambda}' \in \mathcal{Y}(\boldsymbol{x})'_r$ the likelihood is (positively) infinite. To see this, note that the second and fourth terms of the log-likelihood from Equation 4.17 are zero for such parameter vectors; the first term is bounded below for finite $\boldsymbol{\lambda}'$; and the third term is $O(\log(1/\lambda_r))$ which is unbounded above as $\lambda_r \to 0$. The log-likelihood is finite at any interior point of Ω, and hence must be maximised on the $\lambda_r = 0$ boundary.

On the other hand, suppose that $y_{gps,r} \neq 0$ or $\mathcal{Y}(\boldsymbol{x})_r = \emptyset$. In either case $\mathsf{L}(\boldsymbol{\lambda}) = O(\log(1/\lambda_r) - \lambda_r)$ for $\boldsymbol{\lambda}$ close to the $\boldsymbol{\lambda}_r = 0$ boundary, and hence the log-likelihood is negatively infinite at that edge of parameter space. It follows that the maximum likelihood estimator must lie away from this boundary, completing the proof. □

Corollary:
If for all $r = 1, \ldots, N$ we have $y_{gps,r} \neq 0$ or $\mathcal{Y}(\boldsymbol{x})_i = \emptyset$, then the maximum likelihood estimator $\hat{\boldsymbol{\lambda}}$ is interior to Ω.

What the theorem along with the corollary tell us is that if we are certain that there is traffic flow on a given route then the maximum likelihood estimator will reflect this by having a non-zero estimate of the mean flow for that route. As an illustration, if we return to our example where we observe the data $\boldsymbol{x}_{not} = (10, 8)^{\mathsf{T}}$ and $\boldsymbol{y}_{gps} = (2, 1, 3)^{\mathsf{T}}$, we find that for this model we obtain $\hat{\boldsymbol{\lambda}} = (10.0, 3.0, 10.8)^{\mathsf{T}}$. Thus, the estimate from the normal model is much more plausible than the boundary value obtained using the simplified model.

The information matrix $\mathcal{I}_{\boldsymbol{\lambda}_0}$ is more complex here than in Equation 4.16 because of the contributions from the covariance matrices. Specifically, for the normal model based on both link counts and routing data, the rr'th element of $\mathcal{I}_{\boldsymbol{\lambda}_0}$ is given by

$$\begin{aligned} \mathcal{I}_{rr'} \quad = \quad & \hbar\Bigg(\Delta_{rr'} p_r \lambda_{0,r}^{-1} + \tfrac{1}{2}\Delta_{rr'}\lambda_{0,r}^{-2} + (\boldsymbol{AQe}_r)^{\mathsf{T}}(\boldsymbol{AQ}\Lambda_0\boldsymbol{A}^{\mathsf{T}})^{-1}\boldsymbol{AQe}_{r'} \\ & + \tfrac{1}{2}\mathsf{tr}\left[(\boldsymbol{AQ}\Lambda_0\boldsymbol{A}^{\mathsf{T}})^{-1}\boldsymbol{AQE}_r\boldsymbol{A}^{\mathsf{T}}(\boldsymbol{AQ}\Lambda_0\boldsymbol{A}^{\mathsf{T}})^{-1}\boldsymbol{AQE}_{r'}\boldsymbol{A}^{\mathsf{T}}\right]\Bigg) \end{aligned} \tag{4.18}$$

where $\boldsymbol{e}_r$ is the rth elementary vector, $\boldsymbol{E}_r = \mathsf{diag}(\boldsymbol{e}_r)$, and $\Delta_{rr'}$ is the Kronecker delta.

4.4 Inference for p

Until now we have focused on the case where the probabilities for vehicle tracking, $\boldsymbol{p}$, are known. When attempting to estimate $\boldsymbol{p}$ along with $\boldsymbol{\lambda}$ by likelihood methods, we find that we run into identifiability problems.

This follows from the fact that the dimension of the parameter vector $\boldsymbol{\theta} = (\boldsymbol{\lambda}, \boldsymbol{p})$ is $2N$, if we make no constraints on the structure of the probability vector, while we have $N + M$ data points from $\boldsymbol{y}_{gps}$ and $\boldsymbol{x}_{not}$. Given that the number of monitored links, M, will typically be less than the number of routes, N, we cannot expect to obtain reliable estimates under either the normal model or its simplified version.

In particular, any element of the set $\{(\boldsymbol{\lambda}, \boldsymbol{p}) : A\boldsymbol{\lambda} = \boldsymbol{x} \text{ and } P\boldsymbol{\lambda} = \boldsymbol{y}_{gps}\}$ will produce estimates of $\mathsf{E}(\boldsymbol{x})$ and $\mathsf{E}(\boldsymbol{y}_{gps})$ that match the corresponding observed data vectors exactly. Consequently, the only way to distinguish between elements of this set of solutions will be to inspect the given dispersion matrices closely. This can be misleading in the case of the simplified model, because in this case the variance-covariance matrix Σ is not derived from the observed data and we may have little information *a priori* about this matrix otherwise.

We found an elegant solution if we impose some structure on $\boldsymbol{p}$. The most extreme (but potentially useful) case is where the penetration rate of vehicles equipped with tracking technology is homogeneous across the network, in which case we have $\boldsymbol{p} = p\mathbf{1}$ for a scalar p. In particular, we show that a simple method of moments estimator can provide a suitable alternative to a ML estimator. We denote the methods of moments estimator as $\tilde{p}$ and it is derived as follows.

We have $\mathbf{1}^\mathsf{T}\mathsf{E}(\boldsymbol{x}_{not}) = (1-p)\mathbf{1}^\mathsf{T}\boldsymbol{A\lambda}$ and $\mathbf{1}^\mathsf{T}\boldsymbol{A}\mathsf{E}(\boldsymbol{y}_{gps}) = p\mathbf{1}^\mathsf{T}\boldsymbol{A\lambda}$. From this we conclude the equality

$$\frac{\mathbf{1}^\mathsf{T}\boldsymbol{A}\mathsf{E}(\boldsymbol{y}_{gps})}{\mathbf{1}^\mathsf{T}\mathsf{E}(\boldsymbol{x}_{not})} = \frac{p}{1-p}, \tag{4.19}$$

whereby replacing the population means by sample values and then solving for p we obtain the estimator:

$$\tilde{p} = \frac{\mathbf{1}^\mathsf{T}\boldsymbol{A}\boldsymbol{y}_{gps}}{\mathbf{1}^\mathsf{T}(\boldsymbol{A}\boldsymbol{y}_{gps} + \boldsymbol{x}_{not})}. \tag{4.20}$$

In the next section, we demonstrate the application of this estimator in a numerical example. We also examine the consequences for estimation of $\boldsymbol{\lambda}$ when incorrectly making the assumption that the penetration is constant.

4.5 Simulation results

In this section we now address some of the questions that arose during our theoretical analysis of the likelihood functions as well as practicalities that we would like to consider such as the possibility of assuming $\boldsymbol{p}$ is constant, when in fact it differs depending on the routes involved. The likelihood methodology is applicable to any kind of network and more computationally efficient that the MCMC methods previously discussed, thus we illustrate their use on larger and more general systems. We again focus on the OD estimation problem, where the average route flows, $\boldsymbol{\lambda}$, as well as the nuisance parameter $\boldsymbol{p}$ are now of interest.

4.5.1 Experiments with Known Constant Tracking Probabilities

We performed a series of numerical tests using a small eight node network. The first two nodes are origins of travel and nodes 7 and 8 are destinations. Origin and destination nodes are highlighted as darker circles in the depiction of the network in Figure 4.6.

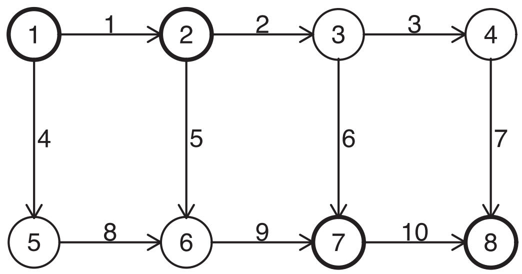


Figure 4.6: Test network.

We order the total of 12 routes lexicographically in blocks by OD pair, and then lexicographically by link sequence within each block. We assume that vehicle counts were measured on all links except for 3, 4, 7 and 8, which is sufficient to produce a 6×12 link-path routing matrix $\boldsymbol{A}$ with linearly independent rows.

We begin by comparing the performance of the full and simplified maximum likelihood estimators under varying levels of demand, and availability of routing data. The mean route flows are $\boldsymbol{\lambda} = (21, 6, 3, 18, 12, 6, 4, 34, 6, 24, 3, 3)^{\mathsf{T}}$ for the lowest level of demand. We also consider medium and high demand scenarios where $\boldsymbol{\lambda}$ is multiplied by factors of 5 and 25 respectively. In these simulation experiments we consider the tracking probabilities $\boldsymbol{p}$ to be a known constant p_0 across all routes. We examine values of p_0 of $0.05, 0.1, 0.2, \ldots, 0.8, 1.0$.

For each combination of demand level and p_0 we generate 100 sets of link counts and tracked routing data using the Poisson model described in section 4.2. We then compute estimates of $\boldsymbol{\lambda}$ for each data set, and then calculate the scaled root mean squared error, defined as

$$\mathsf{SRMSE} = \frac{1}{||\boldsymbol{\lambda}||_1} \sqrt{\frac{1}{100} \sum_{n=1}^{100} \sum_{r=1}^{12} (\hat{\lambda}_r^{(n)} - \lambda_r)^2}, \tag{4.21}$$

over each block of 100 datasets, where $\hat{\lambda}_r^{(n)}$ is the estimate of the average flow rate for the rth route and the nth data set.

This error criterion standardises the squared errors by the total demand, which provides us with a measure that can be used in a straightforward manner for making comparisons in estimation accuracy across different levels of demand.

As a point of reference we also present results obtained using just the routing data at hand and investigate how the accuracy of the estimates contrasts with the results achieved where we use the available link count data as well. We implement our simulations under the assumption that $\boldsymbol{p}$ is known, so that when using the tracking data alone we use $\tilde{\boldsymbol{\lambda}} = \boldsymbol{y}_{gps}/\boldsymbol{p}$, where division is applied elementwise, to estimate the mean route flows. In this experiment we assume the rate of penetration to be constant across all routes, ergo $\tilde{\boldsymbol{\lambda}}$ is given by $\boldsymbol{y}_{gps}/p_0$ here. We can see the results for all methods in Table 4.1.

Method	Tracking probability, p_0									
	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	1
	Low demand									
NOR	0.253	0.198	0.152	0.127	0.113	0.097	0.093	0.087	0.086	0.083
SIM	0.195	0.155	0.123	0.115	0.106	0.097	0.089	0.085	0.084	0.082
GPS	0.346	0.246	0.171	0.144	0.125	0.114	0.099	0.092	0.087	0.082
	Medium demand									
NOR	0.127	0.085	0.061	0.051	0.046	0.044	0.041	0.037	0.036	0.036
SIM	0.108	0.081	0.063	0.052	0.047	0.044	0.042	0.038	0.037	0.036
GPS	0.166	0.116	0.083	0.064	0.055	0.051	0.047	0.041	0.039	0.036
	High demand									
NOR	0.049	0.036	0.026	0.023	0.020	0.019	0.018	0.018	0.017	0.016
SIM	0.048	0.037	0.028	0.024	0.021	0.020	0.019	0.018	0.017	0.016
GPS	0.069	0.051	0.036	0.029	0.025	0.023	0.021	0.019	0.018	0.016

Table 4.1: Scaled root mean squared errors of maximum likelihood estimators using NOR (normal model), SIM (simplified model), and GPS (tracked vehicle data only). The results are computed for varying levels of demand and global tracking probability.

One striking result we notice from studying this table is the fact that the simplified model returns a lower error rate than the normal model when the demand is low. The rationale for this phenomena is rooted in the failure of the normal approximation to the Poisson distribution for very low mean flows.

For small tracking probabilities ($p_0 \leq 0.3$) and low demand ($\lambda_r = 3$ for some r), there are some routes where the mean number of tracked vehicles ($\lambda_{gps,r} = p_0\lambda_r$) is less than one. In such cases, the probability mass functions whose variance depend on the mean rates will become unbounded for $\lambda_{gps,r} \to 0$ (as we are essentially trying to divide by 0). The mass function of the simplified model on the other hand remains finite because the variance is a fixed value which is not influenced by the values of true flow rates.

A closer analysis of the results confirm that for data sets where the normal model returns interior (i.e. non-boundary) estimates, it enjoys (slightly) smaller errors than the results using the simplified model.

It is only in those cases in which the normal method produces boundary estimates (coupled with infinite log-likelihood values) that the results are particularly imprecise.

In both cases, where we have small p_0 in combination with low demand, we find that we often generate zero counts $y_{gps,r}$, which leads to both the normal and the simplified model producing estimates that lie on the boundary of the parameter space. The normal model however, will never produce a zero estimate of an element of $\boldsymbol{\lambda}$ if the corresponding route flow is zero as we saw in our discussion of the theoretical properties of the normal likelihood. The simplified model, in contrast, while it continues to produce fairly reliable estimates due to the relative robustness of GLS estimation, can sometimes turn out negative estimates of average route flows, which are non-sensical.

For moderate demand the differences between the normal and simplified model estimators are miniscule. There is a suggestion that the latter is preferable for the smallest tracking probabilities. This is again explicable in terms of problems with the normal approximation to the Poisson at low demands, and the consequent likelihood of boundary estimates, an effect we still occasionally see, even with the moderate demand, if the tracked vehicle flows rates $\boldsymbol{\lambda}_{gps,r}$ are sufficiently small. Actually, there is still a slight hint of the expected theoretical (asymptotic) advantage of the normal method over the simplified one when the demand is high, but the differences in the results are very small.

We note that in all cases the benchmarking results based on the use of the routing data alone are significantly less accurate than the models which incorporate the link counts as well. Especially in the case of lower penetration rates, $p_0 \leq 0.2$, which are more likely to be encountered in practice in the near future, we often see a 50% loss in estimation efficiency when the link data is ignored.

4.5.2 Experiments with Known but Varying Tracking Probabilities

In reality it is possible and even likely that the route tracking probabilities will vary from OD pair to OD pair, and route to route. The problem is that if we allowed a different probability value for every route to be specified, we would introduce a very large number of nuisance parameters into the model. These would need to be pre-specified because unique estimation of an unstructured probability vector $\boldsymbol{p}$ from the observed data will not be feasible, as we discussed previously.

It will generally be more appealing to specify a parsimonious parameterisation for $\boldsymbol{p}$. For example, we might take it as given that the routing probabilities differ only between OD pairs (but are constant between routes for any given OD pair), or that they vary according to origin.

In this experiment we examine such a scenario for the network from Figure 4.6, where we suppose that vehicles that start out from node 2 are three times as likely as vehicles originating at node 1 to be equipped with tracking devices. Conceptually we can imagine this in a real-life setting as a situation in which node 1 is located in an underprivileged area whilst travellers from node 2, which lies in a more well-off part of town, are more likely to be able to afford the necessary technology.

The results in terms of scaled root mean squared errors are displayed in Table 4.2.

Method	Marginal tracking probability, p_0						
	0.05	0.1	0.2	0.3	0.4	0.5	0.6
	Low demand						
NOR	0.263	0.209	0.146	0.123	0.115	0.103	0.096
SIM	0.195	0.159	0.123	0.112	0.107	0.097	0.090
GPS	0.393	0.278	0.196	0.164	0.145	0.130	0.117
	Medium demand						
NOR	0.138	0.092	0.067	0.054	0.048	0.044	0.040
SIM	0.116	0.087	0.066	0.055	0.049	0.046	0.041
GPS	0.185	0.133	0.093	0.075	0.066	0.057	0.052
	High demand						
NOR	0.055	0.041	0.028	0.024	0.021	0.019	0.018
SIM	0.054	0.041	0.029	0.025	0.022	0.020	0.018
GPS	0.082	0.058	0.042	0.033	0.028	0.025	0.023

Table 4.2: Scaled root mean squared errors of maximum likelihood estimators using NOR (normal model), SIM (simplified model), and GPS (tracked vehicle data only). The results are computed for varying levels of demand and marginal tracking probability p_0. The actual tracking probabilities are $\frac{1}{2}p_0$ for vehicles originating at node 1, and $\frac{3}{2}p_0$ for those originating at node 2.

These probabilities are calibrated so that the marginal probability p_0 of vehicle tracking across the whole network matches the fixed tracking probabilities from the previous experiment.

Aside from the variation in $\boldsymbol{p}$, all other aspects of the experimental design match those of the previous experiment. As expected, the additional uncertainty introduced by the variability in $\boldsymbol{p}$ has led to the errors being 5–10% larger overall than for the first experiment. The estimator $\tilde{\boldsymbol{\lambda}}$ based on routing data only (GPS) shows the greatest degradation in performance.

We have a trade-off in terms of the sources of information, we gain from having more routing information for travellers departing from node 2, but also lose more in terms of estimation accuracy because we acquire less information on travellers originating from node 1. Indeed, if we took the difference in tracking probabilities to its extreme and imagine tracking no vehicles from node 1 and instead have tracking probabilities for those leaving node 2 as $2p_0$, then the results would be much worse.

4.5.3 Experiments with Misspecified Tracking Probabilities

We now take our experiments one step further. In the previous section we assume that tracking probabilities vary depending on where trips originate, but we assume that the different rates $\boldsymbol{p}$ were fairly constant. In other words, the probability of being tracked if entering the network system at node 1 is the same regardless of the route chosen when travelling through the network. We now consider the situation where we accept that there may be variation in these probabilities, but that it is fairly small.

As the estimator developed in section 4.4 applies for systems where the probability of being tracked is homogenous across routes, we might consider ignoring the variability in $\boldsymbol{p}$ and proceed using a single global value p_0 for the tracking probabilities across all routes regardless of the fact that this is not actually the case. Here, we examine the consequences of doing so.

Using again the network from Figure 4.6 we consider four global levels of tracking probability, namely 0.1, 0.2, 0.5 and 0.8. These are the mean values of the tracking probability when averaging across all routes. We then introduce three levels of variability in the elements of $\boldsymbol{p}$. The coefficient of variation for the elements of $\boldsymbol{p}$ are 10%, 50% and 100% for the low, medium and high variability cases, respectively. Naturally we are only able to obtain the higher levels of variation when the mean probability is low. For example, with a mean probability of 0.8 it is impossible to get a standard deviation of 0.8 (i.e. 100% coefficient of variation) because the probabilities are constrained to lie on the unit interval.

As before, we implement these models as if $\boldsymbol{p}$ is known and constant across all routes, where the value of this constant is set to the mean probability in this case. The scaled root mean squared errors are displayed in Table 4.3.

In the case where there is low variation in $\boldsymbol{p}$ the results are little worse than those in Table 4.1, where the model is correctly specified. For example, we get a SRMSE of 0.09 as compared to 0.085 when the mean is 0.1 and we are using the normal model, for the misspecified and correct models respectively.

Mean of $\boldsymbol{p}$	0.1			0.2			0.5		0.8
Variation in $\boldsymbol{p}$	Low	Medium	High	Low	Medium	High	Low	Medium	Low
	Medium demand								
NOR	0.090	0.126	0.214	0.065	0.112	0.199	0.049	0.108	0.045
SIM	0.085	0.130	0.199	0.065	0.125	0.193	0.051	0.114	0.046
GPS	0.120	0.202	0.370	0.089	0.190	0.339	0.063	0.171	0.054
	High demand								
NOR	0.040	0.098	0.191	0.033	0.095	0.196	0.028	0.099	0.027
SIM	0.045	0.116	0.191	0.037	0.114	0.193	0.032	0.107	0.030
GPS	0.062	0.183	0.357	0.051	0.178	0.343	0.042	0.164	0.039

Table 4.3: Scaled root mean squared errors of maximum likelihood estimators using NOR (normal model), SIM (simplified model), and GPS (tracked vehicle data only). The results are computed for varying combinations of level of demand, mean tracking probability, and level of variability in tracking probabilities between routes.

However, as the variance of $\boldsymbol{p}$ increases, there is quite a swift degradation in the accuracy of the estimates for any given value for the mean of $\boldsymbol{p}$. This is to be expected as higher variability in $\boldsymbol{p}$ means that the model misspecification becomes more severe. For the medium demand model the results for normal and simplified models are still fairly similar, but for the high demand case there is a suggestion that the normal model has a slightly better performance overall.

We anticipate this behaviour given that the more unreliable the routing information becomes, the more the link count data will gain in relative importance. Consequently, using the normal model has the advantage that it has the potential to make better use of this latter data source through its incorporation of the functional relationship between the mean and the variance.

Once again, we find that ignoring the additional information provided by the link counts altogether can be shown to be inefficient. As it happens, the drawback of using the routing data only appears even more clearly when the penetration rate is misspecified. When the variation in $\boldsymbol{p}$ is at its greatest, with the mean of $\boldsymbol{p}$ equal to 0.5 and medium variation, the errors using the routing data alone are almost double those for the normal model, 0.108 versus 0.171 and 0.099 versus 0.164, for medium and high demand, respectively.

4.5.4 Application to real network

We finish our presentation of numerical results with an illustrative example using a section of the road network from the English city of Leicester. An abstracted version of this network is depicted in Figure 4.7.

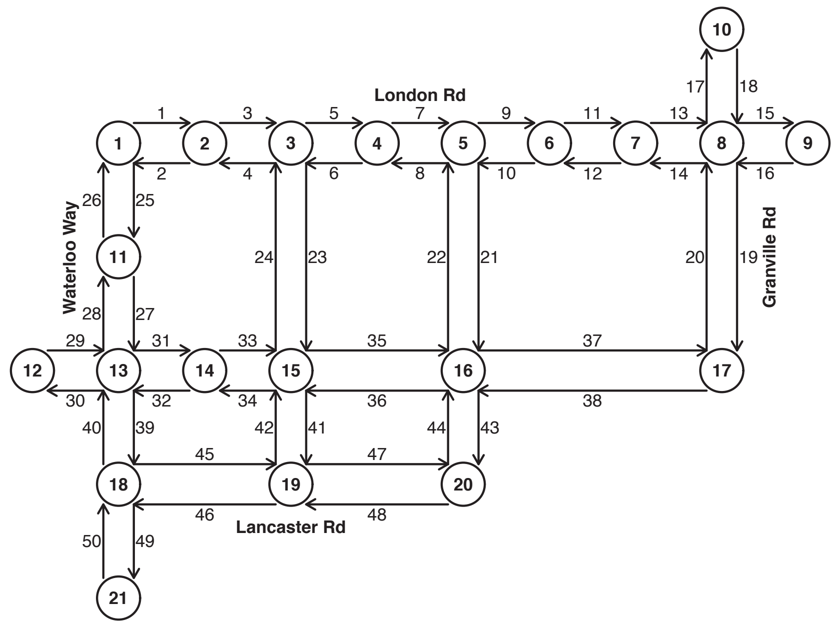


Figure 4.7: Part of the road network in the English city of Leicester.

We obtain mean route flows per unit time, $\boldsymbol{\lambda}_0$, based on weekday estimates obtained in Hazelton (2008). We assume an observation window of length $\hbar = 4$ units, and simulate link counts and routing data with a constant tracking probability of $p_0 = 0.05$ across all routes.

Our first step was to estimate the tracking probability which we now pretend to be unknown using the method of moments estimator from Equation 4.20. This gave $\tilde{p} = 0.0536$ which is very close to the true value of 0.05.

The next step was to apply both the normal and the simplified likelihood methods to the data. For this network (with more than 25 nodes) we found that the normal model did not converge if the starting values were too far away from the true values. Our solution was to first apply the simplified model using a modified version of Bell (1991)'s constrained GLS algorithm, as described in section 4.3.3, and then use the resulting estimates for initialisation when employing the normal likelihood method.

Using the two methods we produce both maximum likelihood estimates $\hat{\boldsymbol{\lambda}}_0$ and associated standard errors (SEs) for $\hat{\boldsymbol{\lambda}}_0$. For the simplified model these are based on variance-covariance matrix $\Sigma_0 = \bar{\lambda}_0 \boldsymbol{I}$ where $\bar{\lambda}_0$ is the mean flow average per unit time over all routes.

We calculate Wald-style 95% confidence intervals, $\hat{\lambda}_{0r} \pm 1.96\, SE(\hat{\lambda}_{0r})$, for the mean flow per unit time for each route r. The confidence intervals are displayed in Figure 4.8 for the simplified and normal model respectively.

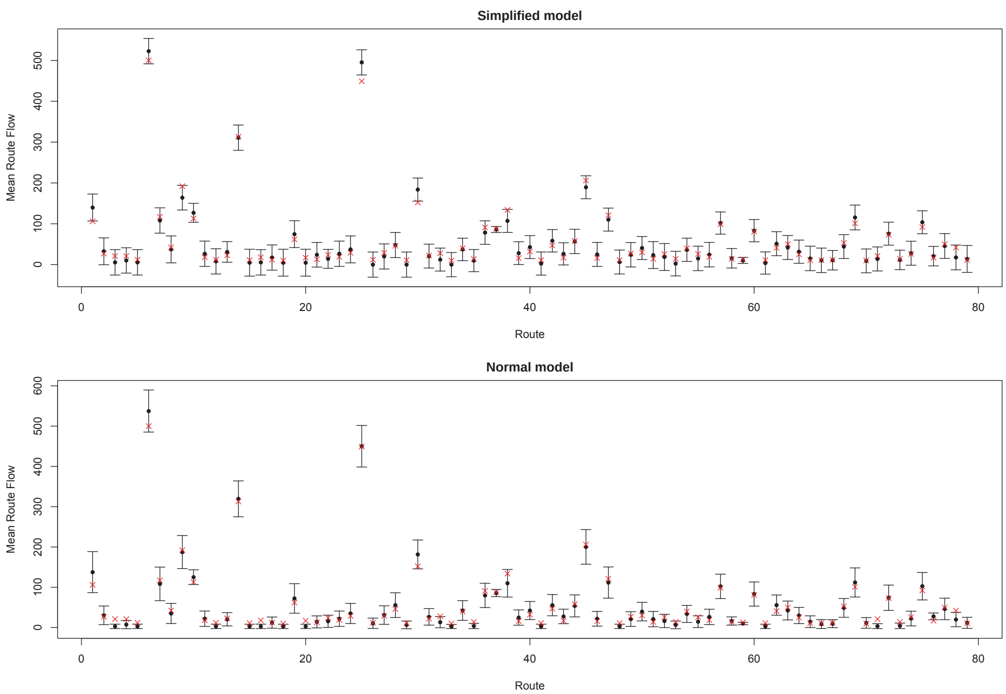


Figure 4.8: Maximum likelihood estimates (dots) and associated 95% confidence intervals for mean route flows in the Leicester network for both models. Red crosses mark the true parameter values.

We notice that the confidence interval widths for the normal model adapt in correspondence to the magnitude of λ_0, while for the simplified model the width of the confidence intervals are the same for all routes.

As a result, we can see that when the average route flow is very small, the confidence intervals of the simplified model appear too wide and more importantly often cross the boundary at zero. This is caused by the fact that the standard errors depend on the variance-covariance matrix for the route flows. For the normal model the variances match the mean flows, mirroring the Poisson process that generates the data, but for the simplified model the variances were set to a common fixed value, $\bar{\lambda}_0$.

The misspecification of the error variance has a limited effect on the quality of the point estimates of $\boldsymbol{\lambda}_0$ themselves, however it renders the standard errors as well as other estimated measures of precision unreliable.

We essentially need to assume the variance-covariance matrix of Σ_0 is either known, or at least have very good prior information about it, in order to produce confidence intervals and test statistics based on standard likelihood theory that will be formally valid. Such problems associated with conducting tests and interval estimation are often overlooked in the literature, but are of considerable practical importance. Transport planning decisions may rest upon clear evidence of changed flow patterns, but it is difficult to draw firm conclusions without reliable test statistics.

4.6 Conclusion

In the introduction to this chapter we discuss the possibility of automatic vehicle location technology becoming a new low-priced source of data available for traffic modelling. However, we also mention some challenges of statistical inference when including information available from GPS-equipped vehicles into the current likelihood models which incorporate link counts only.

There are two modus operandi that may intuitively come to mind. The first would be to follow Maher (1983)'s Bayesian approach and directly include the additional information provided by the tracked vehicles as a Bayesian prior. The second is to use the routing information to design a target matrix in analogy to the work seen in Bell (1991). The difficulty with both these methods is the possibility of us double counting the contributions of the tracked vehicles. A dependency between the link and route counts is very likely given the odds that the data will be collected at the same time, for example, we can expect vehicles equipped with GPS to be driving over the inductor loops on the roads as well. A decomposition of the link counts allows us to separate the two sources of information and develop a likelihood for each source individually which can then be multiplied with each other. This forms a factorised likelihood, where the GPS information can be thought of as an update of the link count likelihood.

We examine a variety of likelihood functions in section 4.3, where some interesting properties become apparent. We also compare the performance of traditional link count likelihoods to the updated likelihood using routing information.

For the simplified model based on link count data only, we show that modelling route flows without any form of exogenous information is an ill-posed problem. The log-likelihood has no unique maximum, so $\hat{\boldsymbol{\lambda}}$ does not exist. We show that the updated likelihood for the simplified model is similar in its general form to the objective functions employed in the GLS estimation of OD matrices developed by Bell (1991), with the routing data in essence playing the role of the target matrix.

The presence of even a small amount of routing data seems to already give rise to more useful estimates of the OD matrix, with the likelihood becoming less flat at its peak, as can be seen in Figure 4.4. This becomes even more evident in the case of the updated likelihood for the normal model, with mean and variance linked according to the underlying Poisson process. As we can see from Figure 4.5 the likelihood now shows an obvious maximum likelihood value.

We then turn our focus to the impact of having a functional relationship between the mean and the variance in the likelihood model. As we have mentioned before, models which link the mean to the variance can provide better estimation results (Cao et al., 2000; Hazelton, 2003), but also have more complicated likelihood functions. A simplified version, where the variance is fixed independently of the mean, requires significantly lower computational efforts in the sampling process. As expected, we demonstrate that a functional relationship between the mean and the variance improves the estimation, however, this can lead to irregular behaviour on the boundary. This becomes especially apparent when we apply the normal model based on link count data only, see Figure 4.3. This phenomena is caused by boundary estimates, that is, values of the mean vector $\boldsymbol{\lambda}$ being equal or close to zero. Due to being functionally linked to the mean, we can expect that as the estimates of $\boldsymbol{\lambda}$ tend towards 0, the variance becomes infinite.

In our simulation study, where we compare the different likelihood models, we find that for a small network with eight nodes the normal and the simplified model both perform almost identically in terms of the accuracy of the point estimates. However, the full normal model has the important advantage of always providing sensible estimates, that is, it will always return non-negative estimates for the mean flow on routes where at least one vehicle is observed, as well as provide valid standard errors in regular cases where the estimates do not lie on the boundary of the parameter space. We therefore recommend the use of the normal model in practice. That being said, we do recognise that using the simplified model for large networks brings a considerable computational advantage because of the availability of a simple algorithm for computing estimates (even when they lie on the boundary).

Another important realisation of the simulation study was the importance of the information provided by the link count data. A model based on information from tracked vehicles consistently delivered point estimates with lower accuracy in comparison to either of the other two models, even for high levels of penetration through GPS equipped vehicles. Ergo, link count data should not be discarded unless nearly 100% routing data is available.

Knowledge about the probability of vehicles using a given route being tracked is critical. If this is constant throughout the entire network then it can be estimated with ease, using a methods of moments estimator proposed in section 4.4. In reality however, we suspect there will often be a marked variation between the probability of vehicle tracking from route to route. One scenario we imagine for example, is that one might expect vehicles beginning their journeys in affluent suburbs to be more likely to be fitted with the necessary technology than those originating from poorer areas.

In our numerical studies, we show that in the case of varying $\boldsymbol{p}$, we can still expect good estimates of the OD matrix even when using a single network-wide estimate for this parameter. This still holds when low levels of variation in the values of $\boldsymbol{p}$ are present. However, as this variation grows we may see a significant bias in parameter estimates obtained from the misspecified model. In such cases it will be necessary to develop models for the variation in the tracking probabilities, perhaps obtained using exogenous data.

Like any models, ours are a simplification of reality and there are numerous extensions and refinements possible. As an important example we have ignored the issue of measurement error in the link counts (and possibly in the routing data also). An error rate of 5% is not uncommon for traffic counters, see for example (Hazelton, 2001a). Fortunately this can be incorporated into the normal modelling framework in a straightforward manner.

We can model the number of detected vehicles that correspond to a single car as $1+\varepsilon$, where ε is a random error term with mean 0 and variance τ^2. As we assume the errors to have an expectation of zero, the inaccuracies in the data will effect only the variance-covariance structure of the link counts.

Thus, if the measurement errors on the links are additive we can model the error prone observed link counts $\boldsymbol{x}_{non}$ as $\mathsf{Norm}(\boldsymbol{AQ\lambda}, \boldsymbol{AQ}(\tau^2 I + \Lambda)\boldsymbol{A}^{\mathsf{T}})$ distributed random variables. Inference can then proceed using the methods presented earlier in this chapter.

However, it is worth noting that in the presence of an additive measurement error for the link counts (with fixed $\tau^2 > 0$) but errorless tracking data, the log-likelihood on the $\lambda_r = 0$ boundary will be positively infinite for any route r for which $y_{gps,r} = 0$, regardless of the $\boldsymbol{x}_{non}$. In other words, we can obtain a zero estimate for the flow rate on any route which was not used by any GPS-equipped vehicles, regardless of how many vehicles were suggested to have been using this route according to the link count data.

This does not make much common sense, and if this becomes a problem, we recommend accounting for measurement error through a multiplicative effect on the covariance matrix. The distribution of the likelihood in this case then becomes $\mathsf{Norm}(\boldsymbol{AQ\lambda}, (1+\tau^2)\boldsymbol{AQ}\Lambda\boldsymbol{A}^\mathsf{T})$, whose boundary properties naturally mirror those of the normal model without measurement error.

Another extension of interest would be to modify the updated likelihood model so it accounts for the varying travel behaviour vehicles equipped with GPS may exhibit. Commuters follow very different behavioural patterns than professional users such as taxi drivers or truckers. So far, we assume that all tracked vehicles are representative of the travelling population in general, but severe bias may be introduced if this is not the case.

Finally, another venue for future research would be to consider data from a sequence of observation periods, that is, the data are measured on a succession of days. In principle, as discussed in the previous chapter, the data from a sequence of observational periods can provide additional insight into the OD estimates though the correlation structure for link counts.

Chapter 5

A New Class of Day-to-Day Traffic Assignment Model

I just invent. Then I wait until man comes around to needing what I've invented.
(Richard Buckminster Fuller)

5.1 Introduction

One aspect of great significance in transport studies is obtaining reliable estimates of model parameters. These kind of problems of inference have been the focus in the preceding chapters. Another very important aspect of transportation research is the the development of traffic assignment models, which will be the focus of this chapter.

Traffic assignment models translate travel demand into trips made between each origin and destination in a network by modelling the interaction between traffic congestion and drivers' route choice decisions. It is vital for accurate road condition predictions to obtain the correct model. The model needs to describe as many underlying features and characteristics of the observed traffic flows as possible.

When obtaining data from successive points in time over a period of days, we may encounter a variety of different phenomena in the observed flows over time. In principle one can postulate explanations as to what variations in the data might represent. Regular systematic patterns such as changes in route choices during the week may be a consequence of attempts to avoid roads prone to congestion on weekdays. Irregular fluctuations in the average traffic volumes may be users responding to underlying conditions such as weather patterns, with certain routes being chosen on rainy days for safety reasons. We may also observe changes in the variation of the flows over time.

This may reflect the manner in which travellers learn about the transport network. If a new road is introduced, we may find that as users trial the new street, there is a certain amount of added volatility in the immediate aftermath of the road opening.

Heterogeneity in the travelling population may also lead to heterogeneity in the observed flows, for example some users may own gadgets that provide additional traffic information and hence allow them to react faster to congestion while other users without this input may be more likely to be driven by habit. This would result in higher variability in the flows and if we now suppose that professional drivers install systems to a greater extent, then we would potentially see changes in the second-order property of the flows on weekdays as compared to weekends.

A better understanding of such variation in flow patterns has increasing policy-relevance in the context of network reliability assessment and the design of intelligent transport systems. However, we note that these kind of specific hypotheses relating to the *causes* are not necessary, or even helpful, given that many properties of a system will differ according to a host of factors, some of which will be *unmeasured*. Instead, the aim in this chapter is to develop a traffic assignment model capable of recreating the day-to-day variability exhibited in real-life traffic flows, with the kind of empirical evidence typically available for traffic networks as our starting point. That is, we aim to propose an empirically-inspired class of traffic assignment models.

As discussed previously in chapter 3, traffic assignment models have historically been based on notions of deterministic or stochastic equilibrium, potentially time-sliced within the day as in dynamic traffic assignment approaches. Such models are not based on empirical considerations, but rather on idealised behaviour where users are fully informed and travelling in a system with no variability. The two most widely used static models of this kind, the Wardrop equilibrium model (Wardrop, 1952) and stochastic user-equilibrium (Daganzo and Sheffi, 1977) are restricted to finding a steady state of the mean flow patterns; they cannot capture the variance in flows as well. As Cascetta (1989) notes, however, by neglecting variation, equilibrium models cannot even be expected to deliver unbiased estimators of *mean* traffic conditions, due to the non-linearity of the traffic assignment process.

Traffic models that also account for second-order properties are an appropriate alternative to consider. In chapter 3 we discuss the estimation of the parameters for Cascetta-type Markov day-to-day dynamic models as described in Cantarella and Cascetta (1995), where a Markov chain is used to model current traffic flows as a function of previous traffic conditions.

Day-to-day dynamic models such as these differ from equilibrium models in that they analyse the evolution over days of travel choices using a learning model based on the effect of previous traffic conditions on the travellers, that is, day-to-day changes in the system are modelled dependent on time. The framework of these models is very flexible as it allows for an effortless integration of different behavioural patterns and types of learning processes (Watling, 1999). In other words, these types of models can concentrate to a great extent on drivers' responses to external factors such as congestion or other changes in the system.

However, as we shall discuss in this chapter, even these types of models are limited in terms of the types of realistic changes in both the mean and the variance over time they may reproduce. They still suffer from some of the limitations of equilibrium analyses, in that, while they permit variation, they are still constrained by the concept of stationarity, where the mean, variance and covariance all remain constant over time. Clearly, there are a variety of ways in which this assumption may be violated; for representing realistic phenomena, day-to-day dynamic models are still restricted in terms of capturing all types of non-stationary characteristics possible.

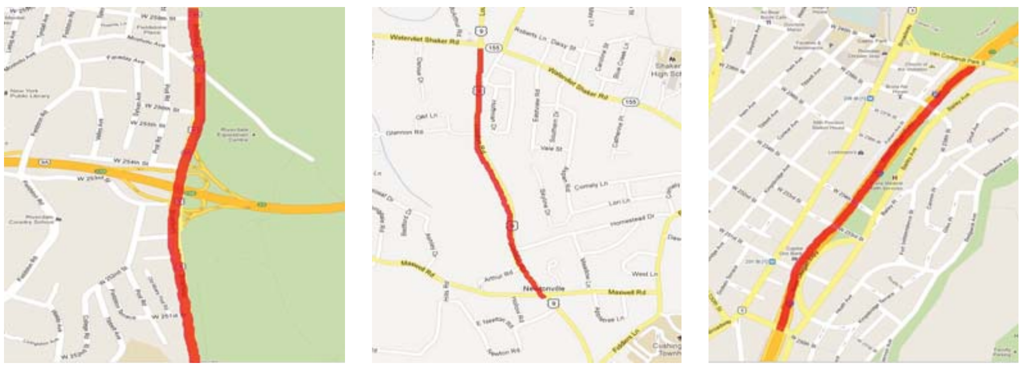
In this chapter, our aim is to combine ideas from transportation network analysis (Cascetta-type Markov models) with developments in statistics in the representation of 'doubly stochastic' processes. In the latter, certain parameters of the model are set to be random variables, and by incorporating such concepts with transportation system analysis, we create a hybrid, stochastic day-to-day dynamic traffic assignment model.

5.2 Case studies

We begin by illustrating some of the phenomena we observe in real-life systems via an example of traffic flows from 2009 measured on consecutive days on some road links in New York state, see Figure 5.1. The data can be sourced from the New York State Department of Transportation website, `www.nysdot.gov/divisions/engineering/technical-services/highway-data-services/hdsb`.

The first set of counts were northbound traffic observed on Broadway road in Bronx, coded as link 10012, in the time between January the 2nd and May the 25th 2009. The second sequence of observations were vehicles travelling north on Loudon road in Albany, coded as link 110137, over a span of seven months from the end of May until the end of December. The third collection of vehicle flows, gathered from February the 2nd until April the 27th, come from cars heading north on the Major Deegan Expressway in Bronx; the road section is coded as 10027.

All flows were measured at 10 minute intervals for an hour a day during their respective observation periods, which was 4-5 pm for the roads located in Bronx and 10-11pm for the road in Albany. No weekend data was available for links 110137 and 10027.



(a) Link 10012, Bronx. (b) Link 110137, Albany. (c) Link 10027, Bronx.

Figure 5.1: Locations of road sections in New York. Reproduced courtesy of 2012 Google Map data.

Figures 5.2, 5.3 and 5.4, are the corresponding time series (and associated autocorrelation plots) for the observed traffic flows. They demonstrate a variety of patterns of temporal dependence.

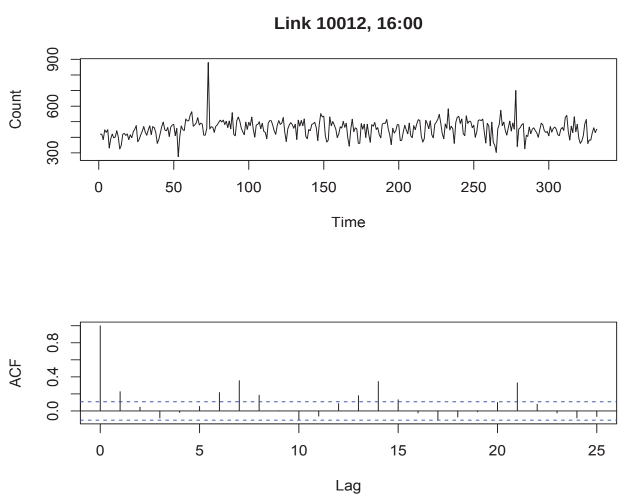


Figure 5.2: Time series and associated autocorrelation plot for traffic counts northbound on Broadway (route 9) in the Bronx, NY.

In Figure 5.2 the day-to-day variation in the flows seems almost entirely haphazard, with a couple of extreme events approximately at time points 50, 75 and 280. In effect we might be observing a sequence of a white noise series, that is, observations of entirely independent and identically distributed random variables.

However, we do note that the autocorrelation plot shows peaks at a 7-day interval, indicating that there is a weekly cycle, where traffic flow is influenced to some extent by whether the observation stems from during the week as opposed to the weekend. Nevertheless, the autocorrelations are weak enough for it to still be reasonable to model this series as white noise.

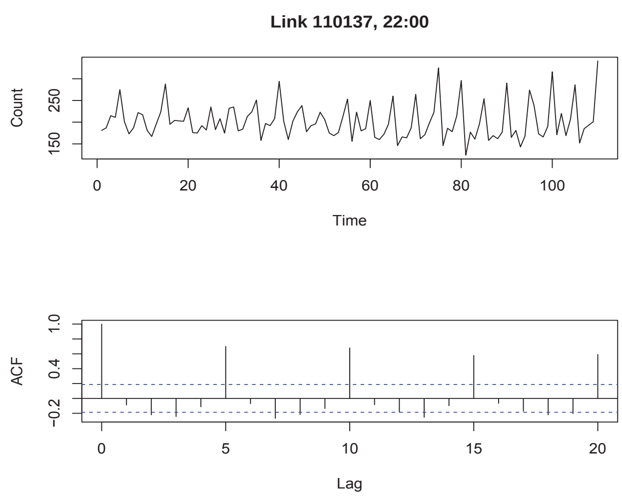


Figure 5.3: Time series and associated autocorrelation plot for traffic counts northbound on Loudon Road (route 9) in Albany, NY.

Let us now consider a different kind of characteristic pattern. In Figure 5.3 we see a clear deviation from a white noise series. There is clear periodicity in the time series plot, leading to strong autocorrelation at lags of multiples of five. This indicates a marked day-of-the-week effect. While the mean flow remains more or less constant, there is at least a hint of increased variability towards the end of the observational period.

Thirdly and finally, in Figure 5.4 we see some still different characteristic features. There is clear evidence of heteroscedasticity, with the variation being significantly larger in the second half of the period than the first. In addition, the autocorrelation plot shows no significant lags, thus no periodicity as seen in Figure 5.3 is evident.

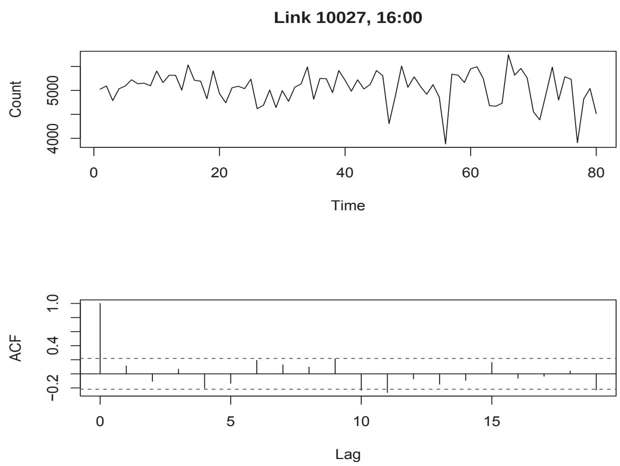


Figure 5.4: Time series and associated autocorrelation plot for traffic counts northbound on Major Deegan Expressway (route 87) in the Bronx, NY.

A model that included a covariate for the weekday effect would not suffice in this case. The implications of this are that we are dealing with a variation in the magnitude of the flows over time that is not just random error. On the contrary, the clustered behaviour of the link flows point to an underlying effect that is causing the temporal variability.

It is these kind of non-stationary traffic flows, in particular the temporal changes in variability as seen in Figure 5.4, that current models, as far as we are aware of, do not reproduce. In the next section we provide a reminder on the details of Markovian day-to-day traffic assignment models and then present our proposed extension of these types of models.

5.3 Day-to-day Markov traffic assignment models

We consider a transport network with M directed links and L OD pairs. Let $\boldsymbol{A}$ be the routing matrix of 0/1 elements, where a_{lr}= 1 if link l is part of route r. The data are link counts measured for short periods of time regularly over several successive days.

At each time point we observe $\boldsymbol{\mu}$, the column L-vector of mean OD demands and $\boldsymbol{y}$ the column N-vector of route flows, both ordered lexicographically. We can introduce a single dummy route for each OD pair denoting the option of not travelling, thus allowing transport systems with fluctuating demand from day-to-day within our modelling framework.

We seek to model the evolution of traffic flows over the network at discrete time intervals t=$1, 2, \ldots, T$. Let $\boldsymbol{x}$ be the M-vector of link flows, where the pattern of travel on the individual sections of road is related to the route flows by $\boldsymbol{x}^t = \boldsymbol{A}\boldsymbol{y}^t$ for time point t. We measure the cost, which we can imagine as time needed, to travel along the links according to the equation $\boldsymbol{c}^t = \boldsymbol{c}(\boldsymbol{x}^t)$. In accordance, we can then define the cost of traveling along the routes as

$$\boldsymbol{k}^t = \boldsymbol{A}^\mathsf{T}\boldsymbol{c}^t. \tag{5.1}$$

In modelling the day-to-day dynamics of the system, we assume that the route choices made by travellers on day t depend on previously experienced travel costs. To this end, we define a learning process in terms of the utility of a route conditional on previous utilities as well as travel costs encountered during d past time periods, where d is a finite constant. There is little practical loss in making such assumptions, but the theoretical properties of our model become more tractable because of the resulting Markov structure.

We write the vector $\boldsymbol{u}^t$ for the vector of mean route utilities on day t, and assume that

$$\boldsymbol{u}^t = h(\boldsymbol{u}^{t-1}, \boldsymbol{x}^{t-1}, ..., \boldsymbol{u}^{t-d}, \boldsymbol{x}^{t-d}|\boldsymbol{\zeta}) \tag{5.2}$$

for some function h, where $\boldsymbol{\zeta}$ is a parameter vector. This definition allows for a straightforward updating process of utilities on a day-to-day basis.

We consider travellers' route choices as random variables whose distribution is defined conditionally on the utility of each route. The evolution of the route flows is then modelled by

$$\boldsymbol{y}^t \sim f(\boldsymbol{u}^{t-1}, \boldsymbol{y}^{t-1}, ..., \boldsymbol{u}^{t-d}, \boldsymbol{y}^{t-d}|\boldsymbol{\theta}) \tag{5.3}$$

where f denotes some conditional probability distribution, with travel demand parameter vector $\boldsymbol{\theta}$. Note that if we define the state vector $\boldsymbol{s}^t$ by

$$\boldsymbol{s} = (\boldsymbol{y}^t, \boldsymbol{u}^t, \boldsymbol{y}^{t-1}, \boldsymbol{u}^{t-1}, ..., \boldsymbol{y}^{t-d+1}, \boldsymbol{u}^{t-d+1}),$$

then the series of successive states $\{\boldsymbol{s}^t; t = 1, 2, ...\}$ is a d-step Markov process.

The class of models defined by Equations 5.2 and 5.3 is quite broad, and incorporates the seminal models of Cascetta (1989). The Markov process defined in this way will be ergodic under rather general conditions. For example, ergodicity is assured if we assume that each traveller assigns a non-zero probability to all routes servicing the relevant OD pair. Ergodicity means that a process has a unique stationary distribution, and in addition the marginal probability distribution of $\boldsymbol{s}^t$ will converge in time to this stationary distribution, regardless of the initial state of the system. This property also implies that we may generate unbiased estimates of features of the stationary distribution from a *single* pseudo-random realisation of the process (Cantarella and Cascetta, 1995). Davis and Nihan (1993) report similar findings for an analysis of such models in the asymptotic case, where demand becomes large in tandem with the network capacity.

Of course, even when the marginal distribution of the system is stationary, the traffic flow patterns will display a complex pattern of temporal dependence. Nonetheless, stationarity dictates that the spatio-temporal correlation structure is constant. Since the link flows on any given day are a deterministic function of the state vector $\boldsymbol{s}^t$, it follows that these models will be unable to reproduce the kind of temporal change in variance that was observed in our third numerical example in the introduction 5.1.

Illustrative example A natural starting point for examining time-inhomogoneous models are the subclass of Markovian models considered in Hazelton and Watling (2004) and previously described in chapter 3. We denote by $c(\boldsymbol{x}^t)$ the M-vector of travelling costs for each of the links in the network.

We suppose that the cost functions $c(.)$ for each link are quadratic, parametrised as $\boldsymbol{c}(\boldsymbol{x}^t) = a^* + (\frac{\boldsymbol{x}^t}{b^*})^2$ so that b^* is proportional to link capacity. The traveller learning process is based on linear filters of past costs with exponentially decreasing weights. That is, we assume in our route choice model that the measured disutilities for day t (based upon the states of the transport system up to and including day $t-1$) is given by

$$\boldsymbol{u}^t = -\sum_{s=1}^{d} \delta^s \boldsymbol{k}^{t-s} \tag{5.4}$$

where $\sum_{s=1}^{d} \delta^s = 1$ for $0 < \delta < 1$ and $\boldsymbol{k}^{t-s}$ is the N-vector of route costs as defined in Equation 5.1. The parameter δ measures the degree to which traveller's previous experiences weigh in travel decisions. A high value means that all past experiences are fairly evenly taken into account, while a value of δ close to zero implies that traffic conditions in the past are hardly if at all considered.

On any given day each traveller selects the route with greatest personal utility. We implement the route-choice mechanism via a standard random utility model, the logit model. We define ϱ_{or}^t to be the probability of a traveller choosing route r when travelling between OD pair o, conditional on the N-vector of measured utilities as

$$\varrho_{or}^t = \frac{e^{\psi u_r^t}}{\sum\limits_{r^* \sim r} e^{\psi u_{r^*}^t}} \tag{5.5}$$

where $r^* \sim r$ if and only if routes r^* and r serve the same oth OD pair and ψ is a measure of the degree to which travellers react to current conditions after changes in the system. We assume that travellers operate independently conditional on the utilities, so that the number of trips on each of the routes follow a Multinomial distribution. Specifically, if $\boldsymbol{y}_{\{o\}}$ is the vector of routes that serve the oth OD pair, then $\boldsymbol{y}_{\{o\}}$ follows a $\mathsf{Mn}(\mu_o, \varrho_{or})$ distribution conditional on the utilities.

5.4 An Extended Class of Models

So far as we are aware, all day-to-day dynamic models that have appeared in the literature to date are members of the models defined by Equations 5.2 and 5.3. There is considerable flexibility within these models for representing the processes by which travellers learn from past travel experiences, and how these experiences influence future route choice.

Nevertheless, despite the apparent scope offered by these models, they are incapable of reproducing the kind of non-stationary behaviour seen in Figures 5.3 and 5.4. This issue with modelling non-stationary processes is well-covered in the time series and spatial-point processes literature. In those areas, a variety of models have been developed to address this matter, for example Generalised Autoregressive Conditional Heteroscedasticity (GARCH) models in financial time series modelling (Bollerslev, 1986; Thavaneswaran et al., 2006), or Cox processes in spatial statistics (Brix and Diggle, 2001; Møller and Waagepetersen, 2007). The essential idea behind these methods is to exchange some of the fixed parameters in the models with random variables that are allowed to change over time (as is the case for GARCH models) and/or space (as for Cox process models). Such models are often termed 'doubly stochastic' to reflect the randomness in the parameter vectors and the random behaviour of the process conditional on those parameter vectors.

Motivated by such models, we propose generalising Cascetta-type traffic models through replacement of some elements of the parameter vectors $\boldsymbol{\zeta}$ and $\boldsymbol{\theta}$ from Equations 5.2 and 5.3 respectively by random processes. This provides a means of replicating the kind of non-stationarity experienced in some of our observed data examples, even when information on the underlying cause is not known.

There is an unlimited number of ways in which the parameters $\boldsymbol{\zeta}$ and $\boldsymbol{\theta}$ might vary. In order to impose some structure, we suggest that these vectors be modelled as Gaussian random processes with correlation structures in time and space. This leads to a doubly stochastic model, analogous to a Cox process, which can allow for patterns of spatio-temporal correlations in both the utilities $\boldsymbol{u}$ and traffic flow vectors $\boldsymbol{y}$.

Illustrative example continued. We present a specific modification of the Cascetta-type model described in the illustrative example previously. We redefine the parameter ψ, that is, the measure of sensitivity, in equation 5.5 as

$$\log(\psi^t) = \omega^t + \kappa \tag{5.6}$$

at time point t where κ is a constant. The values of ψ are kept on a log scale in order to ensure non-negativity because negative levels of sensitivity are not technically possible. We can then describe the parameter $\boldsymbol{\omega}$ as a first order autoregressive process

$$\omega^t = \phi\omega^{t-1} + \varepsilon^t \quad \text{where } \varepsilon^t \sim \mathsf{Norm}(0, \tau^2) \tag{5.7}$$

The marginal distribution of ω is then given by

$$\omega^t \sim \mathsf{Norm}\Big(0, \frac{\tau^2}{1-\phi^2}\Big). \tag{5.8}$$

On the natural scale this implies that the expectation of the level of sensitivity is given by

$$\mathsf{E}(\psi) = e^{\kappa} e^{\frac{1}{2}var(\omega)}. \tag{5.9}$$

The full model of the evolution of route flows is now defined by Equations (5.2), (5.3), (5.4) and (5.5) (which define the system conditionally given any value of ψ) and equations (5.6) - (5.8).

In this illustrative example, and indeed more generally, the random process for the parameters is itself stationary. It follows immediately that the marginal process for the state vector $\boldsymbol{s}^t$ is also stationary.

The stationarity of the marginal process might at first seem to contradict our desire to replicate the kind of non-stationary behaviour observed in our real data examples. However, it is critical to recognize that our doubly stochastic models are non-stationary *conditional* on any (non-trivial) realization of the random process on the parameters. This means that our models can reproduce stochastic trends and changes in volatility in a manner which is impossible when the parameter vectors are fixed. We illustrate this through synthetic numerical examples in the next section.

5.5 Empirical study of Modelling Non-stationarity

We investigate properties of traffic flows for the illustrative Markov assignment model that was introduced in section 5.3 and extended in section 5.4. We make direct comparisons between: (a) traffic flows simulated under the classic Cascetta-type Markov model with a fixed level of sensitivity ψ and (b) the characteristics of traffic flows that result from a modified version of the same model, where the parameter ψ is instead a random variable. For several variations of the parameters in the model we examine the extent to which the models can reproduce the general characteristics seen in the real traffic flows.

5.5.1 Temporal variability

We begin our analysis with a simple two-zone network with only one interzonal movement, in which the zones are connected by two parallel non-overlapping routes.

We run simulations for a range of combinations of three factors, first, the level of inertia of travellers, second, the degree to which travellers learn from their experiences over time and thirdly the number of days taken into consideration in the evolution of the system. In general, we found the number of days taken into consideration was not very influential in terms of the resulting simulated link flows, other than the smoothing effect of averaging over a larger number of days. That is, if experiences up to 20 days in the past were taken into account, then we found that noticeable changes over time were only produced for relatively high levels of sensitivity. For this reason, we only show results where a two-day memory span, $d = 2$ is assumed.

Some particularly interesting cases are considered here, where the contrast between the more stationary-looking flows for the fixed ψ case and the more non-stationary appearance of the flows for the random ψ case is very obvious.

In Figure 5.5 we see simulated link flows on one of the two links, when the sensitivity to costs is relatively high ($\psi = 1$) and the users take into account experiences from the past two days with very little consideration of less recent events (δ=0.01).

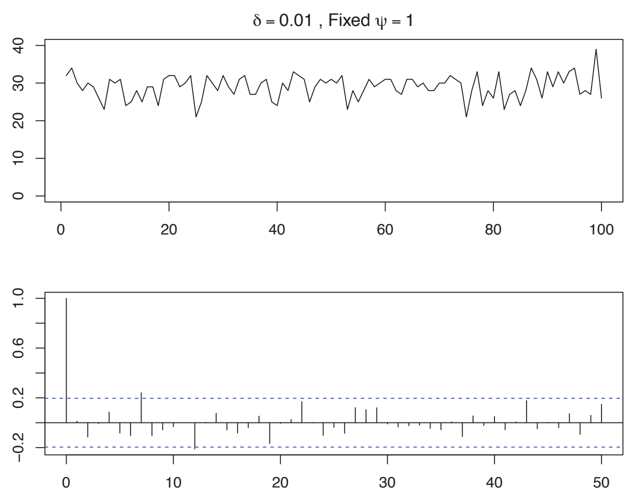


Figure 5.5: Simulated time series and associated autocorrelation plot for traffic counts on two-node network with memory length d=2, a low learning weight $\delta = 0.01$ and fixed sensitivity level $\psi = 1$.

The flows show no signs of changes over time, the mean remains around 30 and the variance appears constant. We have only one link of interest, which enables us to examine the corresponding autocorrelation plot, as seen in the second panel of Figure 5.5, as a way of assessing whether or not the series is indeed stationary in terms of the variance. There appear to be no significant autocorrelations.

In Figure 5.6 we see flows from the modified day-to-day model where we now allow the parameter ψ to vary over time, according to Equation 5.6. Here the parameters ϕ and τ governing the random process $\{\psi^t\}$ have been selected so that the mean value of ψ (as given in Equation 5.9) matches the fixed value $\psi = 1$ considered above. We see the overall level of the mean flows drop over time as well as heteroscedasticity.

In the second panel of Figure 5.6 we can see a plot of the realised sequence of $\{\psi^t\}$ over time. For this particular instance of simulated ψ values, we see an increase in the average levels of simulated values, with a peak around time point 20. The higher than average values are followed by a rapid descent towards very low levels of sensitivity.

In relation to the simulated link flows, we see that the oscillations appear to be higher in magnitude for elevated simulated values of ψ.

As the values of sensitivity drop, the mean of the corresponding link flows drops and the variability becomes less pronounced.

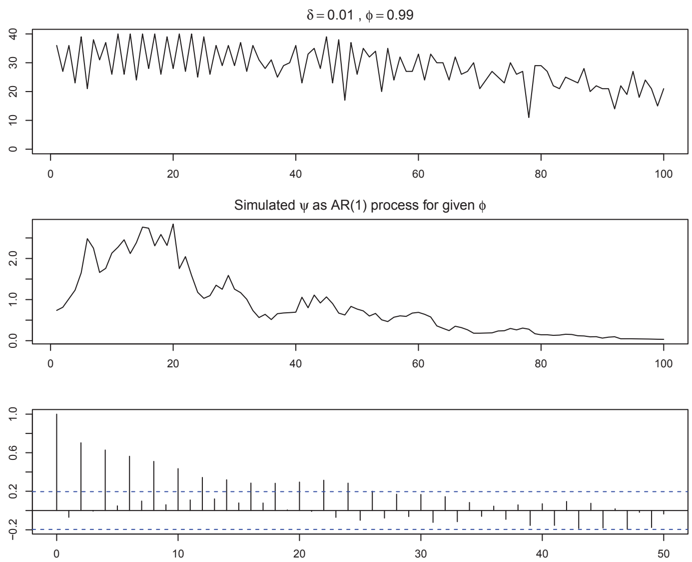


Figure 5.6: Simulated time series and associated autocorrelation plot for traffic counts on two-node network with memory length d=2, a low learning weight $\delta = 0.01$ and random sensitivity levels where $\mathsf{E}(\boldsymbol{\psi}) = 1$.

In the third and final panel, we see that the autocorrelations are decreasing exponentially, albeit slowly.

We now consider the case that the users take into account experiences from the past two days, where the value of δ is now 0.8, that is, all previous experiences carry a more noticeable weight in the route choice process, while keeping the level of sensitivity to costs experienced high ($\psi = 1$). The oscillations in the flows are considerably more pronounced than what we witnessed in 5.5. This is most likely a consequence of the stronger consideration of previous experiences in combination with the high propensity to make changes. The change in δ, as a measure of the extent to which previous experiences are taken into account, also appears to effect the overall mean, which is now lower (around 20 as opposed to 30 as before).

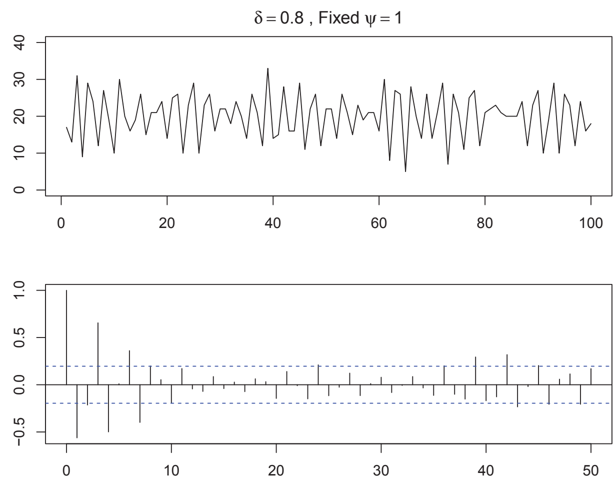


Figure 5.7: Simulated time series and associated autocorrelation plot for traffic counts on two-node network with memory length d=2, a high learning weight $\delta = 0.8$ and fixed sensitivity level $\psi = 1$.

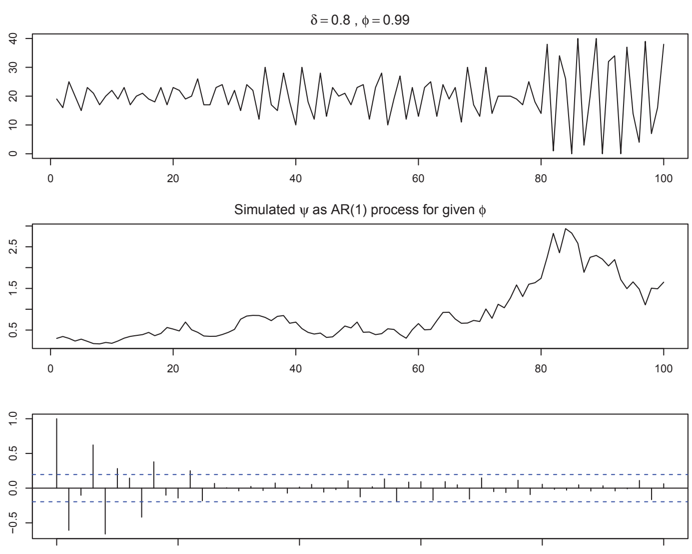


Figure 5.8: Simulated time series and associated autocorrelation plot for traffic counts on two-node network with memory length d=2, a high learning weight $\delta = 0.8$ and random sensitivity levels where $\mathsf{E}(\boldsymbol{\psi}) = 1$.

As before, we now introduce a random process for ψ, calibrated (using ϕ and τ) to have mean $\mathsf{E}(\psi) = 1$. The most striking feature in the resulting simulated link flows depicted in Figure 5.8 is the heteroscedasticity we introduce by letting the parameter ψ be a random process. We have essentially replicated the flows seen in Figure 5.4, that is, the flows in Figure 5.8 imitate the change in the variability of the oscillations without a change in the overall mean.

Both these artificial examples demonstrate well to what extent the clustering seen in the original data can be reproduced by adding an additional level of variability via the introduction of random variables.

5.5.2 Temporal-spatial correlation structure

So far, we have focused on demonstrating the kind of changes in simulated traffic flow patterns we see over time when allowing parameters in our model to be random rather than fixed. We now aim to show that these kind of temporal changes occur in larger networks as well, but more importantly, spatial correlation structures can also be shown to change when we apply our extended models. For our study we return to the network based on a section of the road system in the English city of Leicester, shown in Figure 5.9, as it is a suitably sized system for demonstrating correlation structures.

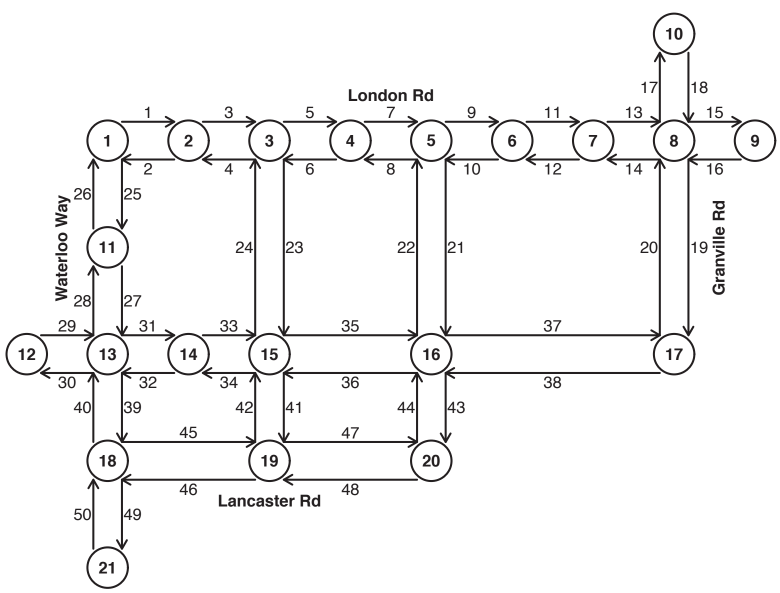


Figure 5.9: Larger sample network.

As in section 5.5.1, our comparisons between fixed and random processes for ψ are calibrated so that the mean value of the latter matches the former.

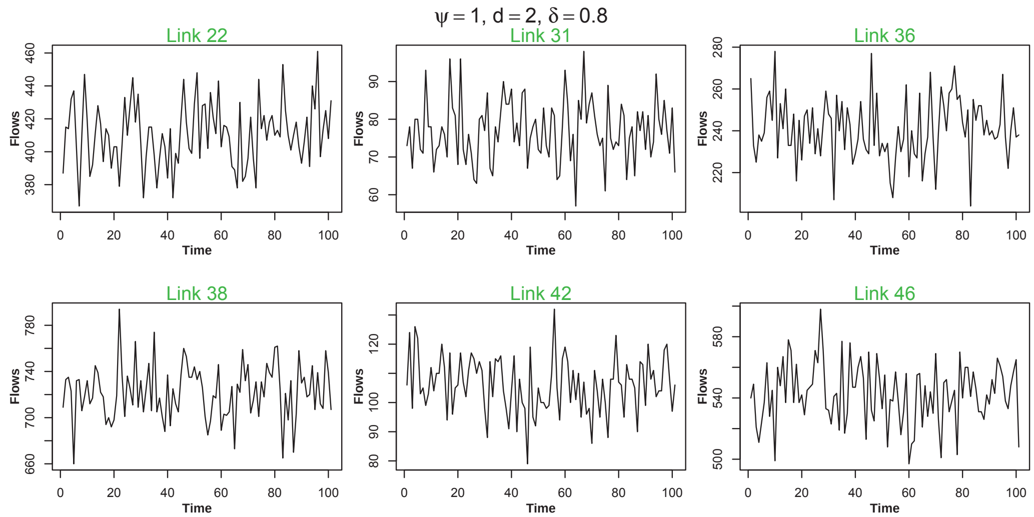
For reasons of brevity we present the flows on only a select number of the road sections, specifically the links 22, 31, 36, 38, 42 and 46. When contrasting the link flows we see in Figure 5.10 we can again see the familiar feature of stationary flows in the fixed ψ case, and temporal changes in the overall mean and pattern of the oscillations in the flows when ψ is a random variable. This same phenomena is something we observe for a variety of different values in key parameters, for example, adjusting the length of memory, d, to longer periods of time delivers similar results. This robustness demonstrates that this effect is a genuine one, in that the extended models, in a similar manner to their counterpart Cox processes, can allow for a change over time in the link flow patterns, in contrast to the standard models.

We now examine the potential spatio-temporal correlation patterns that may arise on a set of interconnected links in the Leicester example.

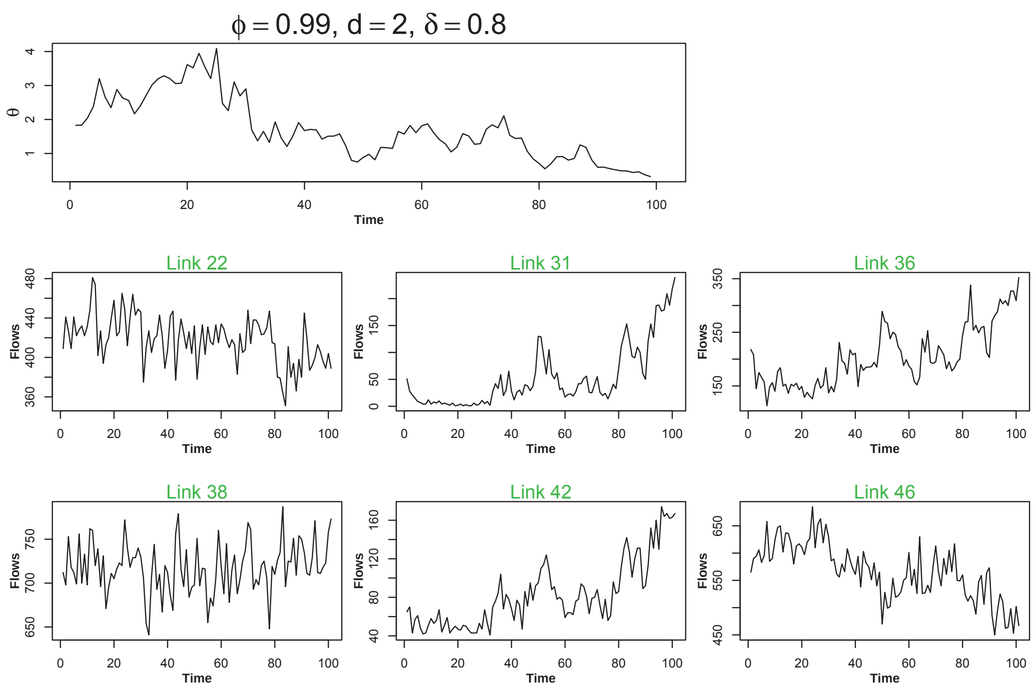
One reason this may be of interest is rooted in the following: From a transport planning point of view if a section of road appears to be having problems with the volumes of demand, a sensible solution may be to widen the road in response. This alteration of the traffic system typically has a more broader effect, a transport planner may well expect smoother flows on the road section in question to lead to better flow patterns on other parts of the network in the vicinity. An implicit assumption being made in the latter stages of that argument is stationarity, whereby the spatial relationships between link flows in the network are expected to remain constant over time. However, if correlation patterns do happen to vary temporally then travellers might find themselves in the situation that the effects of the imagined road-widening scheme are more difficult to predict, for example, there may be even greater congestion on some days rather than a consistently improved traffic flow.

As a way of investigating this aspect we create a plot analogous to a moving average plot, with correlations instead of averages. That is, in Figure 5.11 the first value that is plotted is the correlation between the link flows from days 1 to 100 and the link flows from days 2 to 101. The next value is the correlation between the link flows for time period 2 to 101 and the time period 3 to 102, and so and so forth.

In the case that there is no particular correlation structure, that is, there is no temporal-spatial dependence, we would expect the sequence of points to form a line.



(a) Fixed ψ case.



(b) Random ψ case.

Figure 5.10: Simulated Time series for traffic counts on Leicester network.

In Figure 5.11, we can see that for fixed ψ none of the link pairs exhibit any strong correlation structures that change over time, as the theory predicts. However, when we allow ψ to vary, some of the link flow pairs do display a noticeable change in the correlation structure over time.

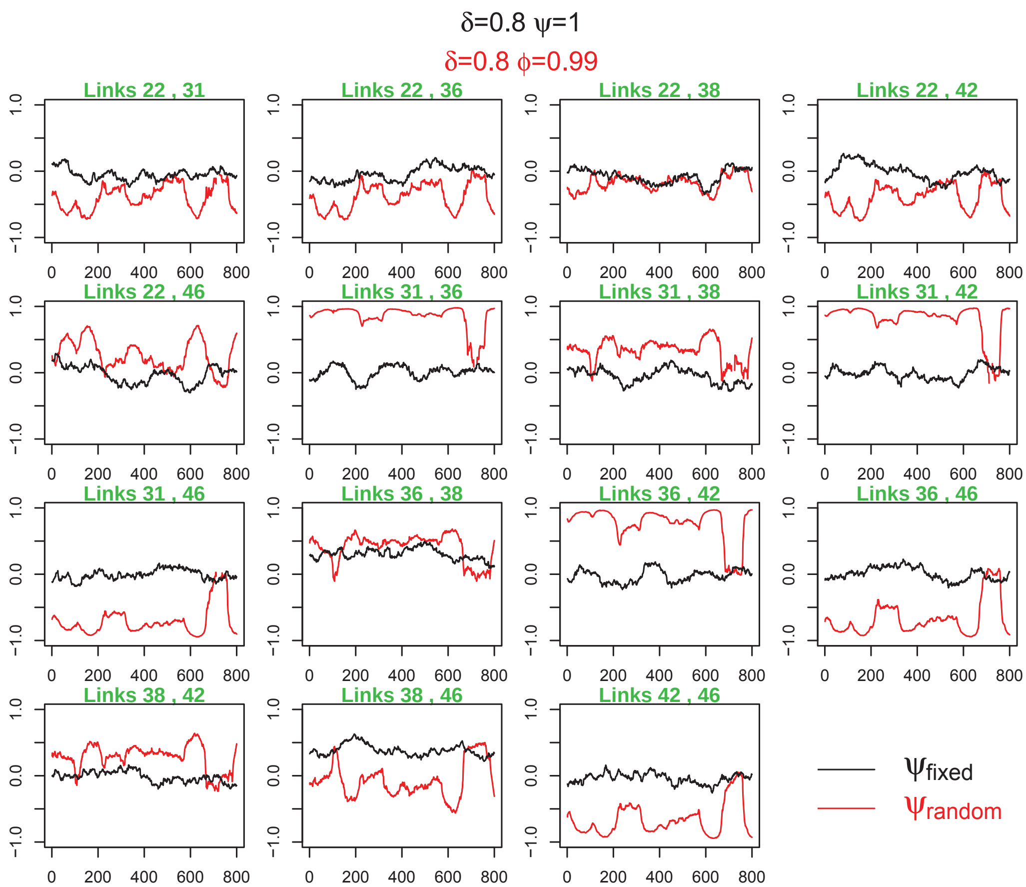


Figure 5.11: Spatial correlations in Leicester network for both fixed and random ψ.

5.6 Conclusion

Non-stationarity can be observed in day-to-day traffic flow data from real world transport systems. However, existing day-to-day traffic assignment models such as those proposed by Cascetta (1989) are by nature stationary, and so cannot faithfully represent such systems.

In this chapter, we describe how such Markovian day-to-day dynamic traffic assignment models can be extended in a straightforward manner by replacing a subset of the fixed parameters in the Markov model with random processes. The resulting models are analogous to Cox process models. They are conditionally non-stationary given any realisation of the parameter processes and hence can reproduce features such as heteroscedasticity in traffic flows.

In particular, our numerical examples were able to mimic some of the features of link flow variation observed in Figures 5.3 and 5.4.

Our illustrative model and numerical examples focus on cases where model parameters were allowed to vary randomly over time, but were constant across space. In part this is a response to the nature of our motivating example. Moreover, focusing on models with just time variation in the parameters simplified our explanation of doubly stochastic models. One possible extension to our proposed model may be to consider adding in spatial random variation in the model parameters, and in such a case determine how the value of a parameter in space will relate to the route choice model. For example, suppose that we wish the logit parameter ψ from our illustrative example to vary across the network. Which specific value applies when computing probabilities according to Equation 5.5? One simple solution would be to define the value of the random process only at the network nodes, and then allocate the value from the appropriate origin node to each traveller. In that case a Gaussian spatio-temporal model for the model parameters would be

$$\begin{pmatrix} \boldsymbol{\zeta} \\ \boldsymbol{\theta} \end{pmatrix} \sim \mathsf{Norm}\left(\mathbf{0}, \Upsilon\right) \tag{5.10}$$

where $\boldsymbol{\zeta}$ and $\boldsymbol{\theta}$ now denote the values of the process at all required times and nodes. The covariance matrix Υ in Equation 5.10 in principle allows for huge flexibility in the form of spatio-temporal correlations. In practice it will be necessary to impose some highly parsimonious parameterisation.

Finally, we note that our doubly stochastic models can be regarded as describing the effects of *unobserved* covariates. However, cyclical variation due to day-of-the-week effects may be concerned with factors that are *observable*. Such properties of the system can in principle be handled by incorporating fixed covariates into the model, although we are not aware of any work in this direction in the literature for dynamic day-to-day models. Thus an interesting future direction would be to consider extended models incorporating not only the random (unobserved) factors that we have considered, but also some fixed (observed) covariates where that information is available.

Chapter 6

Conclusions

A sip of wine, a cigarette, And then it's time to go.
I tidied up the kitchenette; I tuned the old banjo.
I'm wanted at the traffic-jam. They're saving me a seat.

(Leonard Cohen)

The overall aim of this study was to develop better tools for statistical inference and modelling for transport network models.

We begin by commenting on the two approaches used to make inference in underdetermined systems: the frequentist and the Bayesian inferential paradigms. In general we consider a Bayesian approach to be well-suited to problems involving missing data, because the use of a prior regularises an otherwise ill-posed problem. We work with the frequentist paradigm only in the case where we can assume high flows on all the links in the network system.

We then turn to our results in developing improved traffic assignment models. The calibration of existing models is a vital objective in the statistical analysis of transport network systems, however, the implemented models themselves deserve scrutiny in terms of how well they capture underlying features of the observed data.

6.1 Inference using Bayesian methods

We work with the Bayesian paradigm in chapters 2 and 3 as we allow for traffic systems with low demand, such as transit systems, where the use of a prior is vital in order to overcome the identifiability issues we encounter in this situation.

We discuss the need for a methodology for sampling latent route flows, as the unobserved route flows are crucial in making inference for a whole range of statistical problems. One particularly relevant line of work has been based on using a first-order Markov model of traveller behaviour to generate candidate route flows in a Metropolis-Hastings sampler in the case of transit network systems. Our innovation was to provide a modified version of this existing MCMC sampler that can be applied to tree networks. We consider tree networks to be traffic systems where up to two nodes can be reached by exactly one preceding node, and the nodes are ordered.

We compared the statistical properties of the derived Bayesian estimator with existing ones such as the GLS estimator. In particular, we demonstrate the extent to which GLS estimation is restricted to identifying the *mean* route flows, while MCMC estimation delivers more accurate estimates in cases where the ability to use information from second-order properties in the link flows can be exploited.

In cases where the data are collected over a sequence of days, as opposed to intermittent observations, we can model traffic flows using day-to-day traffic models. These kinds of models are more heavily parametrised but in return are able to represent a wider range of travel behaviour. We provide another generalisation of the above-mentioned MCMC sampler to apply to day-to-day traffic models. Simulation studies provide evidence that our technique is capable of reliable estimation of behavioural parameters. Furthermore, the route sampling component of the overall MCMC sampler is found to be vastly more computationally efficient than a route sampling technique developed by Tebaldi and West (1998) in terms of acceptance probability as well as iteration run times. The results were promising, however, our methodology applies only to linear and tree networks while its alternative can be applied to general networks.

6.2 Inference using frequentist methods

Another objective of our research was to examine the effect of additional data sources; that is, data beyond traffic counts observed on a subset of network links. The partial routing information emerging from the increased popularity of GPS tracking, and the development of motorways with installed electronic road toll systems, give rise to data sources which are a promising addition to traffic network models.

As a way of ensuring that the incoporation of tracking information is viable, we assume in chapter 4 that the traffic network systems for which we implement our methods experience high route flows. We assume high demand since we expect only a small percentage of vehicles to be equipped with tracking devices in the near future.

Thus, we can get a decent sample size of tracked vehicles only in the presence of large numbers of vehicles using the system. The assumption of high traffic demand render a normal approximation possible and we apply a frequentist method, namely ML estimation.

A visual investigation of the likelihood models revealed some unusual behaviour, in particular in the case of boundary estimates. We constructed a theorem that lays out the conditions under which we can expect to see these kind of non-standard features in the likelihoods.

An interesting observation was the extent to which our estimators based on the link count data as well as the routing information are akin to ridge regression estimators. The information from the tracked vehicles serve as an addition to the (diagonal) elements of the variance-covariance matrix. This then prevents the dispersion matrix from becoming singular and overall an ML estimator can be found. We also outlined the connection of our estimators with estimators derived from generalised least squares methods.

The introduction of partial routing information leads to a nuisance parameter, the probability of a vehicle in the system being tracked. We consider a methods of moments estimator for a very restricted case, another avenue for research would be finding a more general solution.

We found that the incorporation of the additional data improves estimation, even for very low penetration rates of vehicles equipped with tracking devices, but stress that the traditional link count data remains a valuable source of information in its own right.

6.3 Statistical modelling

At present, traffic assignment models are usually implemented on the implicit assumption that the traffic flows being modelled are stationary. In this thesis, we developed a new class of day-to-day traffic assignment models that are also able to exhibit non-stationary behaviour. To this end we combine existing ideas from transportation network analysis with developments in statistics in the representation of doubly stochastic processes. The result is a hybrid, stochastic, day-to-day dynamic assignment process, which is analogous to Cox process models. These kind of models are conditionally non-stationary given any realisation of the parameter processes.

We present numerical examples that demonstrate that this new class of doubly stochastic day-to-day traffic assignment models can indeed reproduce features such as the heteroscedasticity of traffic flows observed in real-life link flow data. Our illustrative model and numerical examples focused on cases where model parameters were random processes in time, but were constant across space. A further line of future work would be to consider an extension to our proposed model, where we may also consider adding in spatial random variation in model parameters.

6.4 Future avenues for research

The issues addressed in this thesis are quite fundamental to transport research. Firstly, we need traffic assignment models that can capture important features of observed traffic flows, and secondly, we need to be able to reliably calibrate the parameters of these models for practical use. We made some important contributions to both these areas, but acknowledge that there is still scope for future work.

The development of a more general MCMC latent route sampler that applies to networks other than linear and tree topologies remains an important challenge, as at present there is no alternative to a method we show to be highly inefficient in the case of realistic network systems.

Another recurring issue is the trade-off between methods that provide accurate estimates and the desire for swift calculations in the estimation process. The methods we introduce in this thesis present a vast improvement in terms of computational efficiency, while maintaining high standards of the estimation results. However, the computing demand is still high enough to render them unsuitable for very large networks. There is still much work that needs to be done in terms of establishing an estimation technique that fulfils both these criteria in a more satisfactory manner.

A further line of work that would be of considerable interest is a comprehensive testing of currently available day-to-day dynamic models on a wide range of real systems, in particular as a means of assessing the utility of the class of day-to-day models we developed in this thesis.

Notation

Roman Symbols

$\boldsymbol{A}$	routing matrix	5
B	number of first MCMC iterations removed as burn-in	29
$\boldsymbol{D}$	score vector	83
$\boldsymbol{H}$	Hessian matrix	84
$\boldsymbol{I}$	identity matrix	80
$\mathcal{I}$	expected Fisher information matrix	84
$\mathcal{J}$	observed Fisher information matrix	84
J	number of time periods multiplied by number of nodes	39
L	number of OD pairs	2
$\mathcal{M}$	matrix of probability transitions	25
M	number of links	4
N	number of routes	3
$\boldsymbol{P}$	matrix of variance-covariances between tracked route flows	80
$\boldsymbol{Q}$	matrix of variance-covariances between non-tracked route flows	80
R	number of MCMC iterations	24
$\boldsymbol{S}$	matrix of sample variance-covariances between link flows	14
T	number of observational periods	31
U	upper limit of uniform distribution	21
$\mathcal{Y}(\boldsymbol{x})$	set of feasible route flows	15
Z	random variable	24
$\check{a}, \check{b}$	parameters of Gamma distribution	31
a^*, b^*	BPR link cost function parameters	59
$\boldsymbol{b}$	vector of number of vehicles entering the system	35
$\boldsymbol{c}$	vector of link costs	58
d	order of Markov process	24
$\boldsymbol{e}$	elementary vector	93

$\hbar$	length of observation period	82
i^*, j^*	states	24
$\boldsymbol{k}$	vector of route costs	58
$\boldsymbol{p}$	vector of tracking probabilties	80
q	proposal density	26
$\boldsymbol{s}$	state vector	114
$\boldsymbol{u}$	vector of utilties	58
$\boldsymbol{v}$	vector of number of vehicles leaving the system	35
$\boldsymbol{w}$	vector of OD pair flows	2
$\boldsymbol{x}$	vector of link flows	4
$\boldsymbol{x}_{not}$	vector of non-tracked link flows	81
$\boldsymbol{y}$	vector of route flows	3
$\boldsymbol{y}_{gps}$	vector of tracked route flows	80
$\boldsymbol{z}$	vector of number of vehicles still in system when approaching a given node	37

Greek Symbols

α	acceptance probability	26
β, γ	parameters of truncated demand function	65
δ	exponential weighting of traveller memory	58
ϵ	root mean square error standardised by total demand	45
ε	error term	105
$\boldsymbol{\zeta}$	parameter vector	114
η	weighting parameter	12
$\boldsymbol{\theta}$	vector of travel demand parameters	4
ϑ	vector of route choice probabilities	58
κ	constant mean	117
$\boldsymbol{\lambda}$	vector of mean route flows	3
$\boldsymbol{\mu}$	vector of mean OD pair flows	3
$\boldsymbol{\xi}$	vector of Lagrangian multipliers	89
ϖ	state probability	25
ρ	probability of travelling between two given nodes	35
$\boldsymbol{\varrho}$	vector of route choice probabilities	4
τ	variance of error term	105
υ	scalar	21
ϕ	autoregressive model parameter	117
φ	probability of exiting the traffic system	36
ψ	logit model parameter	58

$\boldsymbol{\omega}$	autoregressive series of order 1	117
Δ	Kronecker delta	93
Λ	matrix of variance-covariances between route flows where variance=mean	13
Ω	parameter space	89
Φ_Λ	multivariate normal density with zero mean vector and covariance Λ	20
Σ	matrix of variance-covariances between route flows	11
Υ	spatio-temporal variance-covariance matrix	126

Superscripts

$\dagger$	MCMC candidate index	24
t	time index	8

Subscripts

a	MCMC iteration index	24
i	origin node index	2
j	destination node index	2
l	link index	4
n	data set index	95
o	OD pair index	2
r	route index	3

Mathematical Symbols

f	probability density function	13
E	expectation	3
$\mathcal{L}$	Lagrangian function	89
L	likelihood function	13
Var	variance	82

Acronyms

BLUE	best linear unbiased estimator	12
EM	expectation-maximisation	17
GLONASS	Global Navigation Satellite System	78
GLS	generalised least squares	11
GPS	Global Positioning System	78
MH	Metropolis-Hastings	25
ML	maximum likelihood	13
MSE	minimum square error	12
MVN	multivariate normal	11
OD	origin-destination	2
SE	standard error	101

SRMSE	scaled root mean squared error	95
SUE	Stochastic-User-Equilibrium	6
UE	User-Equilibrium	6

Appendix

The appendix contains all the computer code necessary for implementing the candidate route flow sampler we developed in this thesis that applies to both linear and tree networks. This sampler constructs a suitable proposal distribution by using a simple Markov model for traveller behaviour conditional on link counts. All code was written in R Development Core Team (2009).
The route flow sampler addresses the problem of the need to draw samples from the set of feasible route flows $\mathcal{Y}(\boldsymbol{x})$, a set we cannot fully specify in the overall MCMC sampler which is also presented here.

```
################################################################################
```

Creates incidence matrix of possible routes in the network as determined by its topology. The argument `stops` is the number of nodes in the network and `tom` is the pre-processing table in matrix form, an example of which is given for a small four-node network in Table 2.1.

```
is.od=function(stops,tom){
  help= matrix(F,nrow=stops,ncol=stops)
  for (i in 1:stops){
      if (tom$term[i]){
          j <- i
          while (j > 0){
             j <- tom$pred[j]
             if (j > 0) help[j,i] <- T
          }
      }
  }
}
################################################################################
```

Generates the count vectors $\boldsymbol{b}$ and $\boldsymbol{v}$ of the vehicles entering and exiting the system. The average flow rates used to simulate the data are stored in a matrix `MU` as we have a separate $\mathsf{Pois}(\mu)$ for each day and each route.

```
newdata=function(days,MU,stops){
 u=list()
 v=list()
 for(i in 1:days){
    y.tmp <- matrix(rpois(stops^2,c(MU)),ncol=stops,byrow=T)
    u[[i]]=rowSums(y.tmp)
    v[[i]]=colSums(y.tmp)
  }
  list(u=u,v=v)
}
################################################################################
```

Creates a list of matrices of passengers choices for each day of data collection. An example of a matrix of passenger choices for vehicle counts collected at single time point for a four-node tree network is presented in Table 2.2.

Each passenger choice matrix has as many columns as nodes in the network and two rows. The first row corresponds to the number of vehicles that stay in the network or turn right and the second row to vehicles that leave the system or turn left.

```
turn=function(v,u,day,stops,tom){
   p=list()
   for(i in 1:day){
       tmp= matrix(rep(0, stops*2), nrow=2, ncol=stops)
       for(j in stops:2){
          if(tom$term[j]==1){
             {tmp[    2    ,    j  ]=v[[i]][j]}
             {tmp[tom$left[j]+1,tom$pred[j]]=v[[i]][j]}
          }
          if(tom$term[j]==0){
             tmp[tom$left[j]+1,tom$pred[j]]=sum(tmp[,j])-u[[i]][j]
          }
       }
   p[[i]]=tmp
   }
 p
}
################################################################################
```

Generates link counts based on the matrix of passenger choices. This function is useful in the case where only the entry and exit data are made available for analysis.

```
links=function(p,days,stops,tom){
   x=matrix(0,nrow=days,ncol=stops-1)
   for(i in 1:days){
      x[i,1]=p[[i]][1,1]
      for(j in 2:(stops-1)){
      x[i,j]=p[[i]][tom$left[j+1]+1, tom$pred[j+1]]
      }
  }
  x
}
###############################################################################
```

Calculates hypergeometric proposal probabilities of sampled candidate route flows in accordance to Equation 2.28.

```
Prob=function(n,k,x)  sum(lchoose(n,x))- (lchoose(sum(n),k))
###############################################################################
```

The following function is the actual route flow sampler introduced and developed in section 2.3.3. It samples feasible route flows using a proposal distribution based on a first-order Markov chain and can be applied to linear as well as tree network systems.

The arguments are as follows: `day` is the number of observation periods, T. `stops` is the number of nodes in the network, $M+1$. Arguments `u` and `v` are lists of the entry and exit counts from each of the data collection time points and `p` the list of corresponding passenger choice matrices. `tom` is the matrix form of the pre-processing table for the network. The final argument `OD` is the number of possible routes in the network, N.

The output of this function is as follows. First, an array of candidate route flows. We store the route flows as matrices for programming reasons, and there is one candidate route flow for each observational time period. Second, we obtain a vector of the corresponding proposal probabilities.

```
fey= function(day,stops,u,v,p,tom,OD){
    Yt=array(rep(0,stops^2*day),dim=c(stops, stops, day))
    Qt=rep(0,day)
    for(i in 1:day){
       leaversL=list()  # aka LEAVE system
       leaversR=list()  # aka STAY in system
       leaversR[[1]]= rep(1,u[[i]][1])
       leaversL[[1]]= 0
       for(j in 2:stops) {
            tmp=c((1-tom$left[j])*leaversR[[tom$pred[j] ]],
                   tom$left[j] *leaversL[[tom$pred[j]]])
```

```
            if(tom$term[j]==1){
                leaversL[[j]]=tmp[which(tmp>0)]
                leaversR[[j]]= 0
           }
           if(tom$term[j]==0){
                tmp=c((1-tom$left[j])*leaversR[[ tom$pred[j] ]],
                       tom$left[j]*leaversL[[ tom$pred[j] ]])
                stillpresent=tmp[which(tmp>0)]
       arrivals= rep(j,u[[i]][j])
       stillpresent=c(stillpresent, arrivals)
       turnright= sample(1:length(stillpresent), size=p[[i]][1,j])
       leaversR[[j]]=stillpresent[turnright]
       if (sum(turnright) > 0) leaversL[[j]]=stillpresent[-turnright]
       if (!(sum(turnright) > 0)) leaversL[[j]]=stillpresent
  }
  if(tom$term[j]==1){Yt[,j,i]=tabulate(leaversL[[j]],nbins=stops)}
  prob=Prob(tabulate(stillpresent,nbins=OD),
            p[[i]][1,j],tabulate(leaversR[[j]],nbins=OD))
  Qt[i]=Qt[i]+prob
      }
   }
  return(list(Yt=Yt,Qt=Qt))
}
################################################################################
```

Calculates the likelihood values for the route flows.

```
LL= function(y,mu){
     dpois(y,mu,log=TRUE)
}
################################################################################
```

This function performs step 1 of the overall MCMC sampler, outlined in section 2.2.3. This is the MH sampler where we sample a route flow conditional on the current parameter value of $\boldsymbol{\theta}$, which is entered here as the argument `mut`.
For each observation period at a time, we draw a candidate route flow matrix (for programming reasons a matrix instead of a vector here) and decide based on the rule relating to the acceptance probability whether to update the route flow matrix or not.
The output of this function consists of an array containing the current route flows, a vector of their corresponding proposal probabilities, and a vector that records (for each time period) whether or not the route flow matrix was updated.

```
accept=function(day,Yt,Qt,mut,stops,u,v,p,tom,OD){
    cand=fey(day,stops,u,v,p,tom,OD)
    ycand=cand$Yt
    qcand=cand$Qt
    count=numeric(day)
    for(i in 1:day){
        logppro= sum(LL(ycand[,,i],mut),na.rm=TRUE)+Qt[i]-
                 sum(LL(Yt[,,i],mut),na.rm=TRUE)-qcand[i]
        ppro=exp(logppro)
        U=runif(1)
        if(U<ppro){ {Yt[,,i]=ycand[,,i]}
                     {Qt[i]=qcand[i]}
                     {count[i]=1} }
        if(U>=ppro){{Yt[,,i]=Yt[,,i]}
                     {Qt[i]=Qt[i]}
                     {count[i]=0} }
    }
    return(list(Yt=Yt,Qt=Qt,yc=count))
}
################################################################################
```

This function performs step 2 of the overall MCMC sampler, outlined in section 2.2.3, where we sample a candidate $\boldsymbol{\theta}$ conditional on the current values of the route flows $\boldsymbol{y}$. The matrix `help` is a matrix of the same dimensions as the route flow matrix and serves as an aid for matching the values of the $\boldsymbol{\theta}$ vector with the values of the route flow vector within the matrix format.
We assume that the prior is a conjugate Gamma prior, so this function is a Gibbs sampler, that is, this function directly returns the candidate value of $\boldsymbol{\theta}$ drawn, as the acceptance probability is 1.

```
mus=function(Yt,day,help){
    help*
    rgamma(help,0.1+apply(Yt,c(1,2),sum,na.rm=T),0.1+day)
}
################################################################################
```

We combine all the previous functions into one function `gibbs` which forms the overall MCMC sampling algorithm.
The three arguments `N`, `L` and `burnin` are the number of MCMC iterations, R, the thinning argument and the number of MCMC runs discarded as burn-in, B, respectively.
The output of this function are an array with the estimates of the travel demand parameters $\boldsymbol{\theta}$ and a vector with the acceptance rates as defined by the counters keeping track of whether the route flows were updated or not.

```
gibbs=function(Yt,Qt,mut,N,L,day,OD,stops,u,v,p,tom,help,burnin){
   mu=array(rep(0,stops^2*N),dim=c(stops, stops, N))]
   mu[,,1]=mut
   accy=rep(0,day)
   for(t in 1:N){
     for(k in 1:L){
       mutmp=mus(Yt,day,help)
       current.ys=accept(day,Yt,Qt,mutmp,stops,u,v,p,tom,OD)
       Yt=current.ys$Yt
       Qt=current.ys$Qt
       accy=accy+current.ys$yc
     }
    mu[,,t]=mutmp
   }
 allmu=apply(mu[,,(N*burnin):N],c(1,2),mean)
 return(list(mu_estimates=allmu,y_acceptance_rate=accy/(N*L)))
}
################################################################################
```

We present some code as an example where we implement the full MCMC sampler for the simple four node tree network shown in Figure 2.8, for which the pre-processing table and passenger choice matrix based on an example set of data was shown.
We ran the MCMC sampler for a total of 100 simulated data sets; here we present the code where we assume high traffic demand and that data was collected at 100 time points. We obtain a vector `Com`, with one entry for each generated data set. Each element is the average difference of the estimates from the actual parameter values used to generate the data. This measure of accuracy follows the definition given in Equation 2.29.
The average of all the values returned in `Com` are the values reported in the Tables of the simulation studies performed in section 2.4 of this thesis.

```
tom1=data.frame(cbind(
c(0,1,2,2),  # predecessor
c(0,0,1,0),  # via left
c(0,0,1,1),  # term
c(0,1,0,0))) # split
names(tom1)=c('pred','left','term','split')
stops1=nrow(tom1)
help1=is.od(stops1,tom1$pred,tom1$term)
OD1=sum(help1)
initialmu1 = help1*.01
N=1000000
```

```
L= 200
burnin=0.2
# generation of data:
mN1 = round(help1*runif(stops1^2,1,100),2)
N1=c(c(t(mN1)))
Com=numeric(100)
for(i in 1:100){
   s124=newdata(day4,N1,stops1)
   p124=turn(s124$v,s124$u,day4,stops1,tom1)
   fey124=fey(day4,stops1,s124$u,s124$v,p124,tom1,OD1)
   initialYt124=fey124$Yt
   initialQt124=fey124$Qt
   tmp124= gibbs(initialYt124,initialQt124, initialmu1,N,L,day4,OD1,
                 stops1,s124$u,s124$v,p124,tom1,help1,burnin)
   Com[i]= sqrt(sum((tmp124$mu_estimates-mN1)^2,na.rm=T))/sum(mN1)
}
```

Finally, we note that in order to run some of the functions described here we need to load the R library MASS.

Bibliography

H. Bar-Gera. Evaluation of a cellular phone-based system for measurements of traffic speeds and travel times: A case study from Israel. *Transportation Research Part C*, 15:380–391, 2007. 78

M.G.H. Bell. The estimation of origin-destination matrices by constrained generalized least squares. *Transportation Research Part B*, 25(1):13–22, 1991. 12, 13, 43, 44, 78, 89, 91, 101, 103

J. Besag. Markov chain Monte Carlo for statistical inference. Working paper 9, Centre for Statistics and the Social Sciences, University of Washington, 2001. 19

T. Bollerslev. Generalised autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31:307–327, 1986. 115

P. Bonsall, P. Firmin, M. Anderson, I. Palmer, and P. Balmforth. Validating the results of a route choice simulator. *Transportation Research Part C*, 5(6):371–387, 1997. 3

A. Brix and P.J. Diggle. Spatiotemporal prediction for log-gaussian Cox processes. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 63(4):823–841, 2001. 115

Bureau of Public Roads. *Traffic Assignment Manual*. U.S. Department of Commerce, Washington, D.C., 1964. 59

G.E. Cantarella and E. Cascetta. Dynamic processes and equilibrium in transportation networks: Towards a unifying theory. *Transportation Science*, 29:305–329, 1995. 9, 10, 56, 57, 108, 114

J. Cao, D. Davis, S. Vander Wiel, and B. Yu. Time-varying network tomography: Router link data. *Journal of the American Statistical Association*, 95(452):1063–1075, 2000. 13, 104

E. Cascetta. Estimation of trip matrices from traffic counts and survey data: a generalized least squares estimator. *Transportation Research Part B*, 1(4-5):289–299, 1984. 11, 12, 88

E. Cascetta. A stochastic process approach to the analysis of temporal dynamics in transportation networks. *Transportation Research Part B*, 23:1–17, 1989. 8, 9, 10, 56, 57, 108, 114, 124

E. Cascetta and G.E. Cantarella. A day-to-day and within-day dynamic assignment model. *Transportation Research Part A*, 25(5):277–292, 1991. 10, 56

E. Castillo, J.M. Menendez, and P. Jimenez. Trip matrix and path flow reconstruction and estimation based on plate scanning and link observations. *Transportation Research Part B*, 42(5):455–481, 2008a. 17, 78

E. Castillo, J.M. Menendez, and S. Sanchez-Cambronero. Predicting traffic flow using Bayesian networks. *Transportation Research Part B*, 42:482–509, 2008b. 16

E. Castillo, I. Gallego, and J.M. Menendez. Optimal use of plate-scanning resources for route flow estimation in traffic networks. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):380–391, 2010. 78

R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu. Network tompgraphy: Recent developments. *Statistical Science*, 19(3):499–517, 2004. 79

C.F. Daganzo and Y. Sheffi. On stochastic models of traffic assignment. *Transportation Science*, 11:253–274, 1977. 7, 108

G.A. Davis and N.L. Nihan. Large population approximations of a general stochastic traffic assignment model. *Operations Research*, 41(1):169–178, 1993. 9, 114

A. Ehrlert, M.G.H. Bell, and S. Grosso. The optimization of traffic count locations in road networks. *Transportation Research Part B*, 40:460–479, 2006. 4

D. Gamerman and H.F. Lopes. *Markov chain Monte Carlo: stochastic simulation for Bayesian Inference*. Texts in statistical science. Chapman and Hall, 2nd edition, 2006. 28, 29, 74

A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian Data Analysis*. Texts in statistical science. Chapman and Hall, 2nd edition, 2004. 21

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6): 721–741, 1984. 23

J. Geweke. Evaluating the accuracy of sampling-based approaches to calculating posterior moments. In J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, editors, *Bayesian statistics 4*, pages 169–193. Clarendon Press, Oxford, UK, 1992. 46, 47

R. Hamerslag and M.C. Huisman. Binaire calibratie. *Verkeerskunde*, pages 166–168, 1978. 11

W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. 25

M.L. Hazelton. Some remarks on stochastic user equilibrium. *Transportation Research Part B*, 32:101–108, 1998. 8, 9

M.L. Hazelton. Estimation of origin-destination matrices from link flows on uncongested networks. *Transportation Research Part B*, 34:549–566, 2000. 14, 15

M.L. Hazelton. Estimation of origin-destination trip rates in Leicester. *Applied Stastistics*, 50 (4):423–433, 2001a. 21, 53, 105

M.L Hazelton. Inference for origin-destination matrices: estimation, reconstruction and prediction. *Transportation Research Part B*, 35(7):667–676, 2001b. 12, 13, 16, 63

M.L. Hazelton. Day-to-day variation in Markovian traffic assignment models. *Transportation Research Part B*, 36:637–648, 2002. 9

M.L. Hazelton. Some comments on origin-destination matrix estimation. *Transportation Research Part A*, 37:811–822, 2003. 13, 54, 82, 104

M.L. Hazelton. Statistical inference for time-varying origin-destination matrices. *Transportation Research Part B*, 42:542–552, 2008. 16, 18, 21, 101

M.L. Hazelton. Bayesian inference for network-based models with a linear inverse structure. *Transportation Research Part B*, 44:674–685, 2010a. 15

M.L. Hazelton. Statistical inference for transit system origin-destination matrices. *Technometrics*, 52(2):221–230, 2010b. 19, 22, 23, 30, 34, 36, 40, 53, 64, 65, 66, 73, 75, 76

M.L. Hazelton and D.P. Watling. Computation of equilibrium distributions of Markov traffic assignment models. *Transportation Science*, 38:331–342, 2004. 9, 58, 60, 114

J.C. Herrera, D.B. Work, R. Herring, X. Ban, Q. Jacobson, and A.M. Bayen. Evaluation of traffic data obtained via GPS-enabled mobile phones: The mobile century field experiment. *Transportation Research Part C*, 18:568–583, 2010. 78

A.E. Hoerl and R.W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 42(1):80–86, 2000. 88

IBM. New Zealand Commuter Pain Study, 2010. URL `http://www-03.ibm.com/press/nz/en/presskit/33122.wss`. 1

Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley, 1986.

B. Li. Bayesian inference for origin-destination matrices of transport networks using the EM algorithm. *Transportation Science*, 38:331–342, 2005. 17

B. Li. Markov models for Bayesian analysis about transit route origin-destination matrices. *Transportation Research Part B*, 43(3):301–310, 2008. 22, 34, 35, 36, 37, 53

M.J. Maher. Inferences on trip matrices from observations on link volumes: A Bayesian statistical approach. *Transportation Research Part B*, 17B(6):435–447, 1983. 15, 16, 44, 79, 103

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953. 23

J. Møller and R.P. Waagepetersen. Modern statistics for spatial point processes. *Scandinavian Journal of Statistics*, 34:643–684, 2007. 115

S. Nakayama, R.D. Connors, and D.P. Watling. Estimation of parameters of network equilibrium models: A maximum likelihood method and statistical properties of network flow. In W.H.K. Lam, S.C. Wong, and H.K. Lo, editors, *Transportation and Traffic Theory*, chapter 3, pages 40–56. Springer Science and Business Media, 2009. 14

Y. Nie. A cell-based Merchant-Nemhauser model for the system optimum dynamic traffic assignment problem. *Transportation Research Part B*, 45:329–342, 2011. 55

K. Parry and M.L. Hazelton. Estimation of origin-destination matrices from link counts and sporadic routing data. *Transportation Research Part B*, 46:175–188, 2012. 12

S. Peeta and A.K. Ziliaskopoulos. Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics.*, 1:233–265, 2001. 55

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0 edition, 2009. 135

C. Robert and G. Casella. A short history of Markov chain Monte Carlo: Subjective recollections from incomplete data. *Statistical Science*, 0(00):1–14, 2011. 23

Y. Sheffi. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall Inc, 1985. 4, 7

M. Smith, M.L. Hazelton, H. Lo, G.E. Cantarella, and D.P. Watling. The long term behaviour of day-to-day traffic assignment models. *Transportmetrica*, 2012.

C Tebaldi and M. West. Bayesian inference on network traffic using link count data (with discussion). *Journal of the American Statistical Association*, 93:557–576, 1998. 5, 16, 22, 31, 32, 52, 57, 64, 65, 74, 75, 76, 128

A. Thavaneswaran, S. S. Appadoo, and C. R. Bector. Recent developments in volatility modeling and applications. *Journal of Applied Mathematics and Decision Sciences*, 2006:1–23, 2006. 115

Y. Vardi. Network tomography: Estimating source-destination intensities from link data. *Journal of the American Statistical Association*, 91:365–377, 1996. 13, 16, 22

M.P. Wand. Vector differential calculus in statistics. *The American Statistician*, 56(1):55–62, 2002. 83

M.P. Wand and M.C. Jones. *Kernel smoothing*, volume 44. Chapman and Hall, London, 1995. 69

J.G. Wardrop. Some theorectical aspects of road traffic research. *Proceedings of the Institute of Civil Engineers*, Part II(1):325–378, 1952. 6, 108

D.P. Watling. Maximum likelihood estimation of an origin-destination matrix from a partial registration plate survey. *Transportation Research Part B*, 28(4):289–314, 1994. 17, 77

D.P. Watling. Stability of the stochastic equilibrium asssignment problem: a dynamical systems approach. *Transportation Research Part B*, 33:281–312, 1999. 109

D.P. Watling. A second order stochastic network equilibrium model. I: Theorectical foundation. *Transportation Science*, 36(2):149–166, 2002a. 8, 14, 15

D.P. Watling. A second order stochastic network equilibrium model. II: Solution method and numerical experiments. *Transportation Science*, 36(2):167–183, 2002b. 8, 15

D.P. Watling and M.L. Hazelton. The dynamics and equilibria of day-to-day assignment models. *Networks and Spatial Economics.*, 3:349–370, 2003. 10, 56

D.P. Watling and T. Van Vuren. The modelling of dynamic route guidance systems. *Transportation Research Part C*, 1(2):159–182, 1993. 9, 55

M. Westerman, R. Litjens, and J.P. Linnartz. Integration of probe vehicle and induction loop data- estimation of travel times and automatic incident detection. PATH Research Report UCB-ITS-PRR-96-13, Institute of Transportation Studies, University of California, Berkeley, CA., 1996. 79

H. Yang and J. Zhou. Optimal traffic counting locations for origin-destination matrix estimation. *Transportation Research Part B*, 32(2):109–126, 1998. 4

H.J. Zuylen and L.G. Willumsen. The most likely trip matrix estimated from traffic counts. *Transportation Research Part B*, 14:281–293, 1980. 10, 11