BAYESIAN ANALYSIS OF NETWORK FLOW PROBLEMS

by

Claudia Tebaldi

Institute of Statistics and Decision Sciences
Duke University

Date: _____

Approved:

_____
Mike West, Supervisor

_____
Valen Johnson

_____
Dalene Stangl

_____
Alan Karr

Dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Institute of Statistics and Decision Sciences
in the Graduate School of
Duke University

1997

ABSTRACT

(Statistics)

BAYESIAN ANALYSIS OF NETWORK FLOW PROBLEMS

by

Claudia Tebaldi

Department of Institute of Statistics and Decision Sciences
Duke University

Date: _____

Approved:

_____
Mike West, Supervisor

_____
Valen Johnson

_____
Dalene Stangl

_____
Alan Karr

An abstract of a dissertation submitted in partial
fulfillment of the requirements for the degree
of Doctor of Philosophy in the Department of
Institute of Statistics and Decision Sciences in the Graduate School of
Duke University

1997

Abstract

I study Bayesian models and methods for analysing network traffic counts in problems of inference about the traffic intensity between directed pairs of origins and destinations in networks. This class of problems has been of interest in both communication and transportation network studies. The thesis develops the theoretical framework of variants of the origin-destination flow problem, and introduces Bayesian approaches to analysis and inference. As the first and fundamental stage, the so-called fixed routing problem is addressed. Traffic or messages pass between nodes in a network, with each message originating at a specific source node, and ultimately moving through the network to a predetermined destination node. All nodes are candidate origin and destination points. The framework assumes no travel time complications, considering only the number of messages passing between pairs of nodes in a specified time interval. The route count, or route flow, problem is to infer the set of actual number of messages passed between each directed origin-destination pair in the time interval, based on the observed counts flowing between all directed pairs of adjacent nodes. Based on some development of the theoretical structure of the problem and assumptions about prior distributional forms, I develop posterior distributions for inference on actual origin-destination counts and associated flow rates. This involves iterative simulation methods, or Markov chain Monte Carlo (MCMC), that combine Metropolis-Hastings steps within an overall Gibbs sampling framework. I discuss issues of convergence and related practical matters, and illustrate the approach in a network previously studied in Vardi's 1996 article [37]. I explore both methodological and applied aspects much further in a concrete problem of a road network in North Carolina, studied in transportation flow assessment contexts by civil engineers. This investigation generates critical insight into limitations of statistical analysis, and

particularly of non-Bayesian approaches, due to inherent identification problems. A truly Bayesian approach, imposing partial stochastic constraints through informed prior distributions, offers a way of resolving these problems, and is also perfectly consistent with prevailing trends in updating traffic flow intensities in this field. The second type of problem explored introduces elements of uncertainty about routes taken by individual messages in terms of Markov selection of outgoing links for messages at any given node. For specified route choice probabilities, I introduce the concept of a super-network, namely a fixed routing problem in which the stochastic problem may be embedded. This neatly leads to solution of the stochastic version of the problem using the methods developed for the original formulation of the fixed routing problem. This is also illustrated. The final part of the thesis is devoted to the analysis of real traffic flows along a highway, adopting a different perspective, since the goal gets shifted to the estimation and prediction of the intensity of traffic along a linear path without distinguishing different origin-destination labels. An hierarchical model is built to estimate the fundamental parameters that rule the evolution of the flow through space and time, and a dynamic linear model is used to update posteriors for out-of-sample data as we move through time. Other possible, future directions of investigation are indicated in both the areas touched in this work.

Acknowledgements

I am indebted to Prof. Mike West not only for his indispensable, constant input of ideas and suggestions which contributed to this specific work, but also for his example of an enthusiastic, broad and multiform love for the discipline of statistics: his energy and open-mindedness have been an invaluable inspiration to me as a graduate student.

I am also grateful for the generous help of professors Peter Mueller, Val Johnson, Giovanni Parmigiani and Merlise Clyde.

Although I cannot acknowledge them all by name, I am indebted to everyone at ISDS who contributes to its uniquely friendly and supportive atmosphere, without which my years there would not have been nearly so productive and rewarding.

I thank NISS, and particularly Alan Karr and Jerry Sacks for supporting my research and contributing the data analysed in Chapter 5.

Thanks to the group at the Civil Engineering department of North Carolina State University for the traffic simulator applications, which turned out to play a crucial role in the development of this thesis.

Last, particular credit is deserved by my private editor.

Contents

List of Figures

Chapter 1

Introduction

1.1 The Problem, Its History

The problem I deal with is an old one, and has been mainly the subject of analyses by transportation engineers: given a network of directed links between pairs of nodes, I wish to infer the underlying traffic intensities and actual counts for each origin-destination (**OD**) pair of nodes, based on observed traffic counts on the links.

The word "*network*" refers to both "communication" and "transportation" networks with the counts being either "messages" or "vehicles". Because determining the actual number of **OD** messages in a given network is difficult and costly, the approach of estimating messages by link counts and historical data is developed. The problem is by its very nature underspecified: given a set of link counts there is usually more than one solution (the number of **OD** pairs is in most cases larger than the number of links), and a criterion for determining the "best" solution is needed.

Traditionally, this literature deals with **OD** *matrices* T, whose entry T_{ij} gives the number of trips between source i and destination j. It is usually the case that a given ij trip can be made along a number of alternative paths. With respect to this so called "assignment" problem, a known assignment solution is often assumed

to hold, using a fixed proportional design. The proportion of ij trips that travel along link a is fixed and denoted p_{ij}^a, or in the 0-1 setting that translates directly in the "deterministic routing" framework, all the ij trips travel along one and only one path, so p_{ij}^a is either 0 or 1.

Another concept introduced in the transportation literature involves cost functions which deal with cases in which link-congestion determines assignment. In this setting, cost becomes an endogeneous variable, and the solution of the problem involves finding the entries of both **OD** matrices and assignment matrices, using equilibrium principles. It is the case that most equilibrium based models have not been applied to networks of "realistic" sizes.

All approaches to this problem use a *target* **OD** matrix, $\hat{T}$, as "prior" information. This prior information is usually estimated from historical records and the solution must therefore trade off between this target matrix and the actual observed counts. Some studies assume exact traffic counts; some admit measurement errors or missing link data.

As a general expression, the problem can be regarded as the following optimization program:

$$\begin{aligned} \min F(T,V) &= \delta_1 D_1(T,\hat{T}) + \delta_2 D_2(V,\hat{V}) \quad V,T \geq 0 \\ s.t.\ V &= assign(T) \end{aligned}$$

Where V and T are respectively the exact underlying link counts and **OD** traffic intensities, mutually consistent (that is V can be generated by the collection of T along some specified assignment criterion) and are the least distant from the observed - perhaps incorrectly recorded - link counts $\hat{V}$ and the target matrix $\hat{T}$, originated by historical or other prior information. D_1 and D_2 are some distance measures, often an entropy measure for the first and the Euclidean distance for the second. The relative magnitude of $\delta_1 : \delta_2$ is a function of the reliability of the observations and the

historical data, and can vary among different modeling approaches which can assign different relative weights to the two arms of the problem.

Two categories of modeling approaches can be distinguished: one based on *minimum information* (or the equivalent *maximum entropy*) that derives a direct solution of $V = assign(T)$, or one based on statistical inference, using maximum likelihood techniques, generalized least square estimates, or simple versions of Bayesian analysis. Gradient based solutions have been coded for dealing with the optimization problem specified above, whichever perspective is adopted. The following is a summary of the different solutions proposed.

The idea of the *minimum information* approach is to determine a **OD** matrix that adds as little information as possible to the target/prior one. The justification for this principle is that the information in the link counts is by itself insufficient to determine the correct solution. The estimated minimum information matrix is obtained by minimizing the Kullback-Leibler relative entropy measure:

$$D_1 = \sum_{ij} T_{ij} ln(\frac{T_{ij}}{\hat{T}_{ij}}),$$

where this solution is constrained to generate the observed counts V through the assignment function; that is

$$V = assign(T).$$

The solution to this equation is

$$T_{ij} = \hat{T}_{ij} e^{\lambda_1 p_{ij}^1 + \lambda_2 p_{ij}^2 + ... + \lambda_k p_{ij}^k},$$

where the $(\lambda_1, \lambda_2, \ldots, \lambda_k)$ are the Lagrangian multipliers constraining each link flow. Also, note that this solution assumes proportional assignment. This approach has been described by Van Zuylen & Willumsen [36].

A refinement of this approach which introduces user-equilibrium ideas to account for congestion impedments was proposed by Fisk [12]. Her work belongs to the area of combined trip distribution/assignment models as in the previous analysis of Erlander, Nguyen & Stewart [11], and subsequently by Fisk & Boyce [13]. A technical report by Erlander,Jornsten & Lundgren [10], and a paper by Yang, Iida & Sasaki [39] perform an investigation into the properties of this estimation problem. It is shown that the bilevel optimization models mentioned above can be transformed into single convex programs, under the assumptions that the traffic counts on each network link are available and constitute a user optimal flow pattern.

From another perspective, one of the first formulations of the equilibrium based **OD** matrix estimation problem was described in a technical report by Nguyen [24]. Nguyen presents the possible solutions that satisfy observed traffic counts, but leaves the problem of choosing among them open. This issue was later addressed by an entropy maximizing and a minimum least squares techniques respectively by Jornsten & Nguyen [19], and in a paper by Leblanc & Farhangian [20].

From a statistical inference point of view the most popular approach has been maximum likelihood estimation. ML estimation requires maximizing the likelihood function of observing the target matrix and the traffic counts. To obtain the MLE, the entries of the matrix $\hat{T}$ are regarded as observations, and combined with the observed traffic flows, usually assuming both to be statistically independent. The likelihood function resulting from these assumptions is

$$\mathcal{L}(\hat{T}, \hat{V}|T) = \mathcal{L}(\hat{T}|T) \cdot \mathcal{L}(\hat{V}|T)$$

Under assumptions of stable travel pattern, the matrix $\hat{T}$ is usually assumed to follow a Multinomial distribution when the sampling size is assumed small, a Poisson distribution when the fraction is large. A Poisson distribution is also often assumed for the link counts, and a Multivariate Normal distributed error of measurement

completes the model.

When the three preceding distributional assumptions are valid, the proportional assignment model can be applied and the matrix estimation problem simplifies to maximizing:

$$\max \sum_{ij} (-\alpha_i T_{ij} + \hat{T}_{ij} \ln(\alpha_i T_{ij})) + \sum_{a \in \hat{A}} (\hat{V}_a \ln(V_a(T)) - V_a(T))$$

under the constraints:

$$\begin{cases} \sum_{ij} p_{ij}^a T_{ij} = V_a \ \forall a \in A \\ T_{ij} \geq 0 \end{cases}$$

The solution was given by Spiess [30].

An alternative method can be postulated by assuming that the target matrix $\hat{T}$ represents the true T plus error, and that the observed traffic counts are generated according to the equation:

$$\begin{aligned} \hat{T} &= T + \eta \\ \hat{V} &= V(T) + \epsilon. \end{aligned}$$

Assuming that the error terms have zero mean and finite covariance, the generalised least square estimator can be obtained. Often the covariance matrix has been simply represented by unity diagonal matrices. This implies no covariances between the traffic counts. In fact, Cascetta [5] proves that even with a crude approximation of the dispersion matrix, the result of this method is better than that obtained by a maximum entropy approach, in terms of a smaller mean square error. In this setting, the parameters are the two dispersion matrices and the two quantities $\hat{T}$ and $\hat{V}$ assumed independent. This approach has been developed by the same author jointly with Nguyen [6], by Bell [1], and by Bierlaire & Toint [4]. It has been shown that under particular formulations in terms of **OD** matrix and traffic counts distributions,

the result is the same as that obtained by a stylised Bayesian analysis, or as that obtained by a minimum information approach.

Also, this technique has been extended by considering an equilibrium assignment of the **OD** matrix, a difficult setting that has been approached by using an heuristic algorithm, in a paper by Yang, Sasaki, Iida & Asakura [40], and in a later one by some of the same authors, Yang, Iida & Sazaki [39].

The only Bayesian inference approach present in this literature considers the target **OD** matrix as offering a prior probability distribution for the matrix to be estimated and the observed counts as representing a source of information to update this prior belief. The posterior distribution is then computed by Bayes rule:

$$P(T|\hat{V}) \propto \mathcal{L}(\hat{V}|T) \cdot P(T),$$

but, due to practical computational complications only point estimates could generally be obtained. The computations are performed by assuming either Poisson or Normal probability distributions for the observed traffic counts, and a Multinomial distribution for the prior probability over T. Also, the proportional assignment holds in this setting. For very small networks these computations have been performed by Maher [22], who considers the three cases of

- multiple independent observations,
- observations without error,
- least informative prior.

The whole ensemble of these analyses deals with an optimization problem, where it is a matter of formulating the general double level structure **OD** /equilibrium assignment. The algorithms developed for its solution have been mainly gradient based techniques, by which the target **OD** matrix is taken as an initial solution

adjusted to reproduce the traffic counts by iteratively calculating directions based on the gradient of the objective function. Different proposals for the solution appear in a technical report by Drissi-Kaitouni & Lundgren [9], in a Ph.D. thesis by Chen [7], in proceedings of a conference by Florian & Chen [14], and in another paper by Spiess [31]. A number of comments on these approaches follow.

In each of the analyses sketched above, computational complexity stands in the way of the application of the theory to real data. On only a few occasions do the authors mention having applied their algorithms to a real world problem, and in these cases small networks are undertaken to avoid computational difficulties.

Another weakness of the entropy-maximization or information-minimization approaches is the need of a 'prior' guess for the **OD** matrix to be estimated. This matrix in fact heavily influences the result of the procedure and thereby generates the tautological situation that, in order to begin the analysis, it is necessary to have a good idea of what will be its result. The Bayesian approach will rephrase this crucial piece of information in terms of a prior opinion on the parameters to be estimated, without penalizing the departure from this first input; that is, allowing in fact the result to disagree with it when the data have enough strength to drive the solution to other regions of the space of possible solutions. It is judged more meaningful to use prior distributions more or less diffuse, rather than setting up a complicated penalized optimization problem. Furthermore, in cases in which the structure of the problem does not suffer from problems of identification, it is possible to be so liberal as to start with an uninformative prior idea, so allowing the solution to be independent of any "target structure". But, in the more realistic and common settings where problems of identification arise, the natural and sensible way to find the solution uses a more binding prior distribution, building a bridge towards the traditional engineering approach.

Last but not least in this list of comments, the assumption of independence between traffic counts along different links is rejected as unrealistic. Only computational advantages recommend it, and it is much more sensible to think that adjacent links, lying along one or more common paths, show counts positively correlated.

1.2 The Problem: Categorisation

At this point, I codify and categorise the problem in the notation upon which my discussion shall be based.

The goal is to estimate the origin-destination (**OD**) traffic intensities in a network of routes, from measurements of the traffic along the links. As a specification of the assignment method, which will not constitute an object of my analysis, but I take as given, I assume a proportionality assignment, independent of the link flow, which is a sensible assumption in situations of no congestion. This is the same as saying that given their **OD** addresses,

- the messages choose unique known paths, in case of fixed routing;
- different known paths in known, fixed proportions (i.e. probabilities) in the random routing design.

I also assume that route assignment will not lead to network congestion due to the magnitude of the link flows.

Thus, given a set of nodes connected by directed links, and a number of **OD** pairs, the unique path (= route) along which all the *messages* having a specific **OD** address travel is assumed known, as is the range of the different choices these messages have when dealing with random routing designs. By observing the number of messages along each link, the question: *"how many messages are travelling between each of the **OD** pairs?"* is addressed.

At this point it should be noted that I am taking a "one-shot perspective". In other words, a time interval is fixed during which all the traffic is measured without error: no car or message was in the network before the start of recording, no car or message is left in the network when the counting stops.

The Poisson distribution is assumed to characterize the number of messages travelling between each **OD** pair, throughout. This is actually not as restrictive an hypothesis as it might seem at a first glance. It is often possible to recast a complex distributional structure in terms of a *conditionally* Poisson form, just by adding further levels to the hierarchy, so that even if the data have a different marginal distribution, by conditioning on a set of hyperparameters they can be represented as Poisson variables.

Therefore I postulate that the number of messages having a specific **OD** address follows a Poisson distribution and that each distribution is independent of the number of messages having different **OD** addresses. Note, though, that also this assumption of independence can be relaxed without introducing overly complex computational adjustments by adding an upper level to the hierarchy of the model.

The parameters of these distributions will be the focus of my analysis, which will precedes from the observation of the number of messages travelling along each link. This number is the sum of different traffic counts. Along a particular link, messages are observed that have different origins and different destinations, and happen to share the same fragment of route for a part of their trip. By the observation of the traffic counts on the links in the entire network in a given time interval, and given the pattern of dependencies drawn by the knowledge of the path(s) followed by each **OD** trip, I want to "distinguish" the number of messages having different **OD** identity.

The formulation of the problem already points at further more complex and realistic settings.

- Both the fixed, or deterministic routing networks and random routing networks with a specified fixed assignment constitute special, easy cases of the more complex setting, which generally obtains in the presence of congested traffic and in which the proportions/probabilities depend on the link flow. In our simplified setting, the random routing scheme imposes a fixed known Markov Chain, specific to the **OD** address.

- The Poisson assumption can be criticized in some cases, even from the more general perspective mentioned above.

- Not only a one-shot observation but also a time series of observations can be made, based on a sequence of measurements along each link in time. Such time series observations constitute the object of analysis of the second part of this thesis.

- There can be missing data, when not all the links are measured, or errors of measurement. More generally, there can be situations in which the network is so complex that it would be computationally impossible to keep records on every link. In these cases 'design' questions arise of the type: "assuming that a particular subset of **OD** pairs is relevant, which links should be measured, and what is the degree of approximation introduced when the remaining ones are disregarded?".

- In the case of a time series of observations, sudden jumps in some link intensities make a procedure desirable which would indicate the most probable **OD** pairs responsible for that sudden change.

I have mainly developed the fixed routing setting, addressing the random routing setting more cursorily.

In the second part of this thesis, a first simplified sketch of the solution to the problem of "timed observation" is presented; its main body, however, consists of the analysis of a data set which records traffic flows along highways' segments through time. An entirely different approach is taken, involving time series analysis and dynamic linear modeling, in the view of addressing questions about the evolution and prediction of traffic loads in space and time.

Chapter 2

OD Routing Problems

2.1 Basic Notation and Structure: The Fixed Routing Case

2.1.1 Mathematical Structure

Following Vardi [37], I consider a network whose c nodes have been arbitrarily labelled, under observation for a fixed period of communication or passage of traffic. During this interval of time a collection of messages move through the network; each message originates from one node (its origin **O**) and travels to another (its destination **D**). Denote by X_j the number of transmitted messages for the **OD** pair j, where j runs from 1 to $n = c(c-1)$. Denote by Y_i the number of messages counted along link i, where i runs from 1 to k. A"link" constitutes a path between two nodes that communicate directly, without intervening nodes.

The probabilistic assumption of Poisson distribution for the number of messages having a specific **OD** address is made: $X_j \sim \mathcal{P}(\lambda_j)$ and X_j is independent from X_h if $j \neq h$.

I assume deterministic routing: the jth **OD** pair follows one specific path, and only that one. Let A_{ij} be 1 if the ith link belongs to the path followed by messages

traveling between the **OD** pair j; otherwise $A_{ij} = 0$. Let $\mathbf{A}$ be the matrix $\{A_{ij}\}$. Then the structure of the network is concisely described by the $k \times n$ **OD** matrix, or "routing" matrix, $\mathbf{A}$.

The matrix equation

$$\mathbf{Y} = \mathbf{A} \cdot \mathbf{X}$$

expresses each component of the vector of counts $\mathbf{Y}$ as the result of a sum: the counts of link i, Y_i, result from the sum of the messages traveling between all the **OD** pairs that use link i as part of their path.

Note that the matrix $\mathbf{A}$ is typically singular, with a number of columns larger than the number of rows. In a realistic setting it has no duplicate columns and each column has at least one non-zero entry, which are two conditions sufficient to ensure identifiability of the vector λ. It is easy to see why these two requirements are usually met. A duplicate column means a redundancy in the **OD** pair specification: if two pairs of nodes share the same path, there is no need to distinguishing them. A zero column means that the corresponding **OD** pair is not communicating, while a zero row means that the corresponding link lies outside the network.

The matrix $\mathbf{A}$ also delivers interesting information about the pattern of dependencies in the network. For example the result of the multiplication $\mathbf{A}\mathbf{A}'$ gives a $(k \times k)$ matrix the diagonal element of which AA'_{ii} indicates the number of components of $\mathbf{X}$ involved in Y_i, while the off-diagonal element AA'_{ij} indicates the number of components of $\mathbf{X}$ that Y_i and Y_j have in common. The other product $\mathbf{A}'\mathbf{A}$ gives a $(n \times n)$ matrix whose elements $A'A_{ii}$ indicate the number of $Y's$ in which X_i appears, while $A'A_{ij}$ indicate the number of $Y's$ in which X_i and X_j appear together. These summaries can be looked at as a quick description of the degree of complexity of the network, and of the critical role of the different links and **OD** pairs.

The goal now is to estimate the vector $\lambda = (\lambda_1, \ldots, \lambda_n)$ of parameters of the

Poisson distributions which the various **OD** messages follow, and the actual counts $\mathbf{X}$, based on the observed data $\mathbf{Y}$.

2.1.2 Vardi's Solution

Recently Vardi [37] dealt with this problem by approaching it in a way that is closer to my own methodology than to that adopted in the transportation engineering literature. Nonetheless, my work does not corroborate his solutions.

What follows is a summary of Vardi's conclusions, for the case of a deterministic-routing network.

In all but trivial cases the likelihood of the data is impossible to maximize in the exact form, but, given the distributional assumptions, the likelihood equations $0 = \frac{\partial l}{\partial \lambda_j}$ for $j = 1, \ldots, n$ (where $l = \log L$) in vector notation can be expressed as follows. Assuming K repeated observations on the network, denoted by $\mathbf{Y}^k$, and with corresponding **OD** flows $\mathbf{X}^k$, the likelihood equations, in vector notation, are

$$0 = \frac{1}{K} \sum_{k=1}^{K} E_\lambda[\mathbf{X}^k | \mathbf{Y}^k = \mathbf{A}\mathbf{X}^k] - \lambda$$

. Theoretically, the EM algorithm could be used to search for a solution. Its step:

$$\lambda^{n+1} = E[\bar{X} | \mathbf{Y}^1, \ldots, \mathbf{Y}^K, \lambda^n]$$

becomes, by the linearity of the operator $E[\cdot]$ and independence across the K observations:

$$\lambda^{n+1} = \frac{1}{K} \sum_{k=1}^{K} E[\mathbf{X}^k | \mathbf{Y}^k, \lambda^n].$$

Complications arise from the fact that the single summand of the type $E[\mathbf{X}^k | \mathbf{Y}^k, \lambda^n]$ requires first finding all the solutions in natural numbers of $\mathbf{AX} = \mathbf{Y}$.

Often different points of the feasible space for the vector λ constitute stationary points for the iterations, thus the algorithm gets stuck in one of them, depending on the starting point, and converges to a non-MLE point. The problem can be circumvented providing there are a 'sufficient number' of repeated observations of $\mathbf{Y}$, using a 'natural' Normal approximation, and this approximation can be applied to two components of the problem. As a first possibility, each of the summands can be normally approximated, but this is not the best approach for several reasons: the approximation is poor unless we know *a priori* that all the λ_i are large; besides it causes cumulative approximation errors at each step of the iterations, introducing numerical instability and the possibility of converging to negative values of λ_i. The second possible approximation relies on the Central Limit Theorem, by which the distribution of $\bar{Y}$ can be assumed Normal, completely determined by the mean vector $\mathbf{A}\lambda$ and the covariance matrix $\frac{1}{K}\mathbf{A}\Lambda\mathbf{A}'$, where Λ is the diagonal matrix having the components of the vector λ as non-zero elements. In this case the log-likelihood based on $\bar{Y}$ is

$$l(\lambda) = -log|\mathbf{A}\Lambda\mathbf{A}'| - K(\bar{Y} - \mathbf{A}\lambda)'(\mathbf{A}\Lambda\mathbf{A}')^{-1}(\bar{Y} - \mathbf{A}\lambda).$$

When K is large the second term is the dominant and its argmin constitutes a reasonable large-sample substitute for the MLE. However, the resulting estimate is not a simple weighted square with positive constraints because the weights are themselves functions of λ. Vardi then notes that the sample's first and second moments can be equated to their theoretical values to obtain a system of estimating equations linear in λ:

$$\begin{pmatrix} \bar{Y} \\ \underline{S} \end{pmatrix} = \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix} \lambda,$$

where $\underline{S}$ is the sample covariance matrix stretched out as a vector, $\mathbf{B}$ is a $(k(k+1)/2 \times n)$ matrix whose ii' element row is the element-wise product of row i and row i' of the matrix $\mathbf{A}$. If we left-multiply the likelihood equation by $\mathbf{A}$, we see that

any stationary point of the log-likelihood function l also satisfies the first-moment equations $\bar{Y} = \mathbf{A}\lambda$. Note, though, that the converse would hold only if the columns of $\mathbf{A}$ were linearly independent, which is not the case since $n > k$ generally obtains in this setting. It is also to be noted that because of nonnegativity constraints on λ the MLE could be a boundary point, and not necessarily a stationary one for l.

The EM algorithm is then applied to solve the moment-equations iteratively. Given the linear system expressed in block form:

$$\begin{pmatrix} \bar{Y} \\ \underline{S} \end{pmatrix} = \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix} \lambda,$$

the form of the algorithm is, at each iteration

$$\lambda_j \leftarrow \frac{\lambda_j}{\sum_{i=1}^{k} a_{ij} + \sum_{i=k+1}^{k+m} b_{ij}} \{ \sum_{i=1}^{k} \frac{a_{ij} \bar{Y}_i}{\sum l = 1^n a_{il} \lambda_l} + \sum_{i=k+1}^{k+m} \frac{b_{ij} S_i}{\sum l = 1^n b_{il} \lambda_l} \} \ , \ j = 1, \ldots, n,$$

where $m = k(k+1)/2$ is the length of the vector $\underline{S}$. Now this equation can be rewritten as

$$\lambda_j \leftarrow \frac{a_{\cdot j}}{a_{\cdot j} + b_{\cdot j}} \hat{\lambda}_j(\mathbf{A}, \mathbf{Y}, \lambda) + \frac{b_{\cdot j}}{a_{\cdot j} + b_{\cdot j}} \hat{\lambda}_j(B, \underline{S}, \lambda),$$

with "$\cdot$" denoting summation over the subscript that it replaces, and where $\hat{\lambda}_j$ is defined by the canonical EM iteration for this type of "linear inverse positive" problems as in Vardi and Lee [38]:

$$\hat{\lambda}_j(\mathbf{A}, \mathbf{Y}, \lambda) \equiv \frac{\lambda_j}{a_{\cdot j}} \sum_i \frac{a_{ij} y_i}{\sum_l a_{il} \lambda_l} \, , \quad j = 1, \ldots, n \, ,$$

for any matrix $\mathbf{A}$ and vectors $\mathbf{Y}$ and λ with their respective positive constraints.

In the implementation of this approach there are a number of issues that warrant closer consideration, mainly in relation to the positivity constraints on the variances of the variables, and the fact that these variances coincide with the means under the

Poisson assumption, since due to sampling variability, the empirical first and second moments often satisfy neither the positivity nor the identity constraint.

Vardi acknowledges these problems but leaves them unresolved.

2.1.3 A Bayesian Approach

Here the problem is approached from a Bayesian point of view, so that inference about the vector λ is directed to obtaining its posterior distribution starting from some prior beliefs and then updating by the information contained in the observed counts.

The computation of the posterior distribution in closed form would in most of the cases become unmanageable. The adoption of the powerful tool of iterative simulations, however, allows for a conclusive analysis. The idea of **Gibbs Sampling** permits to simulate from a manageable distribution as a means to gaining insight into a more complex one. The literature on this topic has quickly become huge, so I just refer to the book by Tanner [33], for a thorough development of the subject. In this setting the specific technique of **Chained Data Augmentation** proves to be useful in simplifying the algorithm of iterative simulation: given the set of $\mathbf{X}$ together with $\mathbf{Y}$ the posterior for λ would become conditionally independent of $\mathbf{Y}$, i.e.,

$$P(\lambda|\mathbf{Y},\mathbf{X}) \equiv P(\lambda|\mathbf{X}).$$

Therefore,

$$P(\lambda|\mathbf{Y}) = \int_{\mathcal{X}} P(\lambda|\mathbf{Y},\mathbf{X})P(\mathbf{X}|\mathbf{Y})d\mathbf{X}$$

and

$$P(\mathbf{X}|\mathbf{Y}) = \int_{\Lambda} P(\mathbf{X}|\mathbf{Y},\lambda)P(\lambda|\mathbf{Y})d\lambda$$

simplify by noticing that

$$\lambda|\mathbf{X},\mathbf{Y} \equiv \lambda|\mathbf{X}.$$

These two distributions, $P(\lambda|\mathbf{X})$ and $P(\mathbf{X}|\mathbf{Y},\lambda)$ are iteratively simulated to eventually draw $(\lambda,\mathbf{X})$ from the posterior distribution of interest, $P(\lambda,\mathbf{X}|\mathbf{Y})$.

Notice that in this way a bridge is built between Vardi's approach to the problem and that of the transportation engineers, in the sense that a by-product of the analysis is a whole set of $\mathbf{X}$ (that is, a whole set of **OD** matrices) that constitute the central focus of the attention in the transportation engineers' approach. In fact, one way to express the result of the analysis is in terms of representative sets of **OD** trips, as an alternative to the more abstract concept of "distribution of the λ parameter of the Poisson distributions". When dealing with problems of larger dimensions, however, it should be remembered that finding the set of the most likely joint set of **OD** pairs poses a significant challenge to the normal "three dimensional" parameters of speculative inquiry!

It is often the case that the problem at hand is so complex, or so new, that no "prior knowledge" is available. In this case, it is exclusively the data which determine the result of the analysis. The concept of non-informative priors plays the leading role here, but this approach is dangerously open to the risk of identification problems, as will be shown in Chapter 4. Thus in cases where a set of prior information about the **OD** intensities λ is available it is incorporated as a guidance for the estimation procedure.

Despite the danger, sometimes no precise idea about the distribution of the parameters that rule the behavior of each Poisson count is gathered 'a-priori', and a "non-informative" prior that expresses the independence of each λ_j from the others and assigns a Uniform distribution to each of them on a "large enough" support, is chosen. The computation of the posterior for λ given the $\mathbf{X}$ is in this case trivial, being

$$P(\lambda|\mathbf{X}) \propto P(\lambda)P(\mathbf{X}|\lambda) \propto P(\mathbf{X}|\lambda)$$

where the last term is a product of Poissons, delivering, as a function of λ, the product of kernels of Gamma distributions. We get at the end that for each $i = 1, \ldots, n$

$$P(\lambda_i|\mathbf{X}) \equiv P(\lambda_i|X_i) \propto \lambda_i^{X_i} \mathrm{e}^{-\lambda_i},$$

that is each λ_i has a Gamma distribution of parameters $(X_i + 1, 1)$.

In more informed situations a constraint can be assigned to the support of the λ_i, by limiting on the left or on the right the prior Uniform distribution. In this case the corresponding posterior would simply be a truncated Gamma with the same parameters as before. A different solution would be to specify a conjugate Gamma prior, whose parameters (α_i, β_i) would deliver well chosen prior mean and prior variance. In this other case the posterior distribution would be still a Gamma with parameters $(\alpha_i + X_i, \beta_i + 1)$.

As for the distribution of $\mathbf{X}$ given $\mathbf{Y}$ and λ, it is always possible, given the singular structure of the network under observation, to build its conditional distribution by partitioning it into a number of terms by the multiplication rule:

$$P(X_1, \ldots, X_n|\mathbf{Y}, \lambda) = P(X_1|\mathbf{Y}, \lambda)P(X_2|X_1, \mathbf{Y}, \lambda) \cdots P(X_n|X_1, \ldots, X_{n-1}, \mathbf{Y}, \lambda)$$

in which the individual factors can be simplified since not every component of $\mathbf{X}$ depends on all the others, and some of the distributions degenerate in a single point mass, when the value of the corresponding X_i is exactly determined by a linear function of the $\mathbf{Y}$ vector and the subset of $\mathbf{X}$ to which it is conditioned. This is actually the first approach taken, in analyzing two networks chosen as "guinea pigs". It turns out to be unmanageable, though, when the structure of the network becomes more complex, or when the link counts are large, expanding the range of the support for the $\mathbf{X}$'s components.

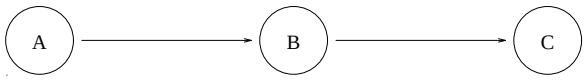


Figure 2.1: The smallest non-trivial network.

2.1.4 The Small Network

The network in Figure 2.1 has only two links:

$$\begin{aligned} Y_1 &= AB \\ Y_2 &= BC \end{aligned}$$

and only three **OD** pairs:

$$\begin{aligned} X_1 &= AB \\ X_2 &= BC \\ X_3 &= AC. \end{aligned}$$

Therefore its origin-destination matrix is the 2×3 matrix:

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

For this simple structure, the product of the three conditional distributions $p(X_1|Y_1, Y_2, \lambda)$, $p(X_3|X_1, Y_1, Y_2, \lambda)$ and $p(X_2|X_1, X_3, Y_1, Y_2, \lambda)$ reduces to the product of the three Poisson distributions and two indicator functions. In fact:

$$\begin{aligned} P(X_1|Y_1, Y_2, \lambda) &\propto \frac{e^{-\lambda_1}\lambda_1^{X_1}}{X_1!} \cdot \frac{e^{-\lambda_3}\lambda_3^{(Y_1 - X_1)}}{(Y_1 - X_1)!} \cdot \frac{e^{-\lambda_2}\lambda_2^{(Y_2 - Y_1 + X_1)}}{(Y_2 - Y_1 + X_1)!}, \\ P(X_3|X_1, Y_1, Y_2, \lambda) &\equiv P(X_3|X_1, Y_1) \equiv I_{\{Y_1 - X_1\}}, \\ P(X_2|X_1, X_3, Y_1, Y_2, \lambda) &\equiv P(X_2|X_3, Y_2) \equiv I_{\{Y_2 - X_3\}}. \end{aligned}$$

Therefore solving this simple example is just a matter of simulating a variable from a discrete distribution on the support $S_{X_1} = [\max(0, Y_1 - Y_2), Y_1]$ with point masses given by the product of the three Poissons above.

Once X_1, X_2, X_3 are given, the values for the three components of λ are generated from three Gammas whose scale parameter is one and whose shape parameter is the corresponding component of the vector $\mathbf{X}$ augmented by one, which is the posterior distribution corresponding to the non-informative Uniform prior.

A simple S-Plus [32] function ran several thousands of iterations and produced a sample of $10,000$ values for X_1 and consequently for (X_2, X_3) and for the vector λ.

Before exploring the simulations, it is useful to consider this simple structure, where the true posterior distributions for the parameters in λ are easily computed and compared to the results of the simulation experiment. In fact,

$$\begin{aligned}
p(\lambda|\mathbf{Y}) &\propto P(\mathbf{Y}|\lambda) \cdot \prod_{i=1}^{3} p(\lambda_i) \\
&\propto \sum_{\mathcal{X}_3} P(Y_1, Y_2, X_3|\lambda) \cdot \prod_{i=1}^{3} p(\lambda_i) \\
&\propto \sum_{\mathcal{X}_3} P(Y_1, Y_2|X_3, \lambda) \cdot P(X_3|\lambda_3) \cdot \prod_{i=1}^{3} p(\lambda_i) \\
&\propto \sum_{\mathcal{X}_3} P(X_1 = Y_1 - X_3|X_3, \lambda) \cdot P(X_2 = Y_2 - X_3|X_3, \lambda) \cdot P(X_3|\lambda_3) \cdot \prod_{i=1}^{3} p(\lambda_i) \\
&\propto \sum_{\mathcal{X}_3} P(X_1 = Y_1 - X_3|\lambda_1) \cdot P(X_2 = Y_2 - X_3|\lambda_2) \cdot P(X_3|\lambda_3) \cdot \prod_{i=1}^{3} p(\lambda_i) \\
&\propto \sum_{\mathcal{X}_3} \mathcal{P}(Y_1 - X_3; \lambda_1) \cdot \mathcal{P}(Y_2 - X_3; \lambda_2) \cdot \mathcal{P}(X_3; \lambda_3) \cdot \prod_{i=1}^{3} p(\lambda_i).
\end{aligned}$$

Given this joint distribution, we integrate out two of the three λ terms to get the posterior for the single component. Under the assumed Uniform prior on the support for each λ_i, the last term in each right-hand-side formula disappears as part of the normalizing constant.

In the computation of the marginal distribution of the single λ_i the two other

components of λ are integrated out by integrals of the form:

$$\int_0^A \frac{e^{-\lambda_j}\lambda_j^x}{x!} d\lambda_j,$$

where A is the "large enough" superior limit of the support for λ_j, and x changes with the index j, being $Y_1 - X_3\,, Y_2 - X_3\,, X_3$ respectively for $j = 1\,, 2\,, 3$. The integrand is easily recognized as a manageable Gamma density of parameters $(x+1, 1)$; thus, the integral has the value of its cumulative distribution function at the point A.

At this point, the true posterior distributions of interest can be re-written in three simple lines:

$$\begin{aligned}
P(\lambda_1|Y_1, Y_2) &\propto \sum_{\mathcal{X}_3} \mathcal{P}(Y_1 - X_3; \lambda_1) \cdot \mathcal{G}(A; Y_2 - X_3 + 1, 1) \cdot \mathcal{G}(X_3 + 1, 1) \\
P(\lambda_2|Y_1, Y_2) &\propto \sum_{\mathcal{X}_3} \mathcal{P}(Y_2 - X_3; \lambda_2) \cdot \mathcal{G}(A; Y_1 - X_3 + 1, 1) \cdot \mathcal{G}(X_3 + 1, 1) \\
P(\lambda_3|Y_1, Y_2) &\propto \sum_{\mathcal{X}_3} \mathcal{P}(X_3; \lambda_3) \cdot \mathcal{G}(A; Y_1 - X_3 + 1, 1) \cdot \mathcal{G}(Y_2 - X_3 + 1, 1),
\end{aligned}$$

where the first $\mathcal{P}$ is intended as the probability function of a Poisson, while the $\mathcal{G}$ stands for the cumulative distribution function of a Gamma density.

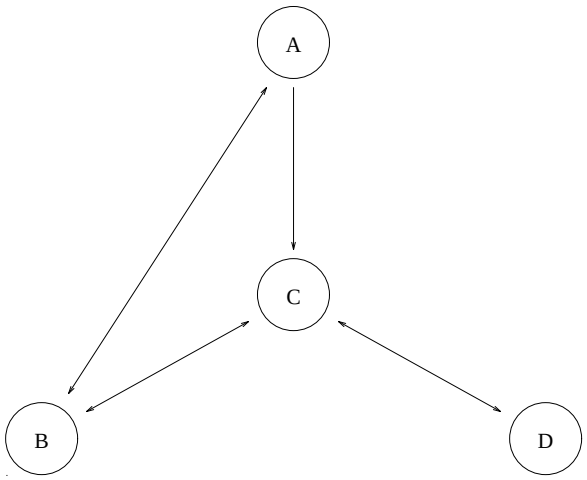


Figure 2.2: An example of a larger network.

2.1.5 The Large Network

The larger network in Figure 2.2 has seven links

$$\begin{aligned} Y_1 &= AB \\ Y_2 &= BA \\ Y_3 &= AC \\ Y_4 &= BC \\ Y_5 &= CB \\ Y_6 &= CD \\ Y_7 &= DC, \end{aligned}$$

and twelve **OD** pairs:

$$\begin{aligned} X_1 &= AB \\ X_2 &= AC \\ X_3 &= ACD \\ X_4 &= BA \end{aligned}$$

$$
\begin{aligned}
X_5 &= BC \\
X_6 &= BACD \\
X_7 &= CBA \\
X_8 &= CB \\
X_9 &= CD \\
X_{10} &= DCBA \\
X_{11} &= DCB \\
X_{12} &= DC.
\end{aligned}
$$

Therefore its 7×12 **OD** matrix is:

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

For this more complex structure, the joint distribution of the 12 **OD** pairs is the product of 12 uni-dimensional ones, some of them reducing to a degenerate point-mass, by the effect of the constraints introduced by the marginal sums given by the values of $\mathbf{Y}$. The computation of the twelve conditional distributions becomes immediately much more burdensome.

A graphical scheme of the dependencies among the $\mathbf{X}$ and $\mathbf{Y}$ values helps to structure the sequential conditioning in the correct way. In Figure 2.3, square boxes indicate the observed variables $\mathbf{Y}$, whereas circles represent the unobserved randomly generated variables $\mathbf{X}$, among which those components that are deterministic functions of the others are highlighted by double edged circles.

An arrow from one variable to another indicates a dependency: the "parent" node

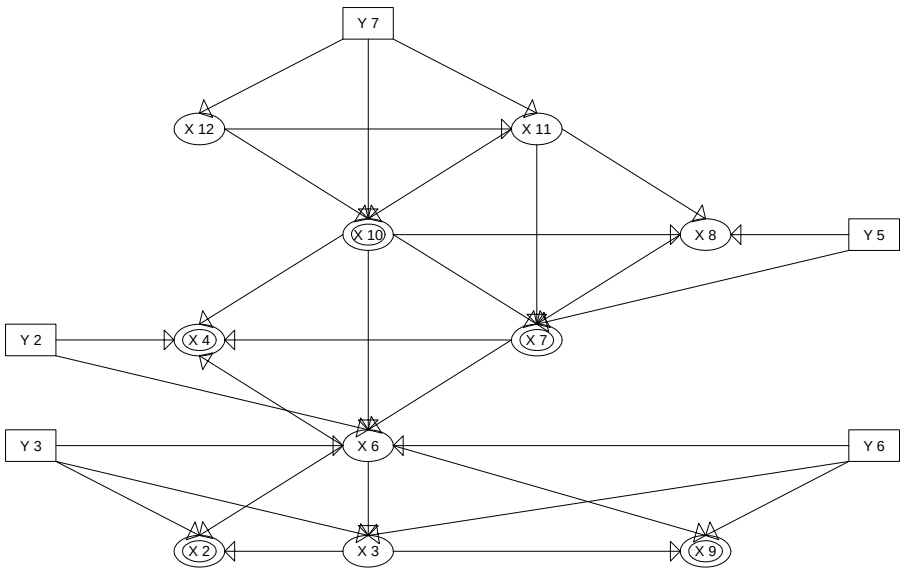


Figure 2.3: Influence diagram for the 7-node network.

represents a constraint on the support of the "child" node. When the child node is in a double edged circle, its value is exactly determined by those of all its parents.

The joint distribution of the 12 components of $\mathbf{X}$ is the long formula below:

$$\begin{aligned} P(X_1,\ldots,X_{12}|\mathbf{Y},\lambda) &= e^{-\sum_{i=1}^{12}\lambda_i}\cdot\frac{\lambda_{12}^{x}}{x!}\cdot I_{\mathcal{X}}\cdot\frac{\lambda_{11}^{y}}{y!}\cdot I_{\mathcal{Y}(x)} \\ &\quad\cdot\frac{\lambda_{10}^{(Y_7-x-y)}}{(Y_7-x-y)!}\cdot\frac{\lambda_{8}^{z}}{z!}\cdot I_{\mathcal{Z}(x,y)} \\ &\quad\cdot\frac{\lambda_{7}^{(Y_5-z-Y_7+x)}}{(Y_5-z-Y_7+x)!}\cdot\frac{\lambda_{6}^{q}}{q!}\cdot I_{\mathcal{Q}(x,y,z)} \\ &\quad\cdot\frac{\lambda_{4}^{(Y_2-q-Y_5+z+y)}}{(Y_2-q-Y_5+z+y)!}\cdot\frac{\lambda_{3}^{v}}{v!}\cdot I_{\mathcal{V}(x,y,z,q)} \end{aligned}$$

$$\cdot\frac{\lambda_9^{(Y_6-q-v)}}{(Y_6-q-v)!}\cdot\frac{\lambda_2^{(Y_3-q-v)}}{(Y_3-q-v)!}$$

$$\cdot\frac{\lambda_5^{Y_4}}{Y_4!}\cdot\frac{\lambda_1^{Y_1}}{Y_1!},$$

where the various support sets are determined as follows:

$$\begin{aligned}
\mathcal{X} &= \{0,\ldots,Y_7\}\\
\mathcal{Y}(x) &= \{y : y\geq 0 \ \&\ Y_7-x-y\geq 0\}\\
\mathcal{Z}(x,y) &= \{z : z\geq 0 \ \&\ Y_5-z-Y_7+x\geq 0\}\\
\mathcal{Q}(x,y,z) &= \{q : q\geq 0 \ \&\ Y_2-q-Y_5+z+y\geq 0\}\\
\mathcal{V}(x,y,z,q) &= \{v : v\geq 0 \ \&\ Y_6-q-v\geq 0 \ \&\ Y_3-q-v\geq 0\}.
\end{aligned}$$

It is now easier to write out the unidimensional conditional distributions. I omit for simplicity of notation the conditioning on $(\mathbf{Y},\lambda)$. Then:

$$P(X_1)=I_{\{Y_1\}}$$

$$P(X_5)=I_{\{Y_4\}}$$

$$P(X_{12}=x)=\sum_{y\in\mathcal{Y}(x)}\sum_{z\in\mathcal{Z}(x,y)}\sum_{q\in\mathcal{Q}(x,y,z)}\sum_{v\in\mathcal{V}(x,y,z,q)}P(X_1,\ldots,X_{11},X_{12}=x)$$

$$P(X_{11}=y|X_{12}=x)=\sum_{z\in\mathcal{Z}(x,y)}\sum_{q\in\mathcal{Q}(x,y,z)}\sum_{v\in\mathcal{V}(x,y,z,q)}P(X_1,\ldots,X_{11}=y,X_{12}=x)$$

$$P(X_{10}|X_{11}=y,X_{12}=x)=I_{\{Y_7-x-y\}}$$

$$P(X_8=z|X_{11}=y,X_{12}=x)=$$

$$\sum_{q\in\mathcal{Q}(x,y,z)}\sum_{v\in\mathcal{V}(x,y,z,q)}P(X_1,\ldots,X_8=z,X_{10}=Y_7-x-y,X_{11}=y,X_{12}=x)$$

$$P(X_7|X_8=z,X_{11}=y,X_{12}=x)=I_{\{Y_5-z-Y_7+x\}}$$

$$P(X_6=q|X_8=z,X_{11}=y,X_{12}=x)=$$

$$\sum_{v\in\mathcal{V}(x,y,z,q)}P(X_1,\ldots,X_6=q,X_7=Y_5-z-Y_7+x,X_8=z,$$

$$X_{10} = Y_7 - x - y, X_{11} = y, X_{12} = x)$$

$$P(X_4|X_6 = q, X_8 = z, X_{11} = y, X_{12} = x) = I_{\{Y_2 - q - Y_5 + z + y\}}$$

$$P(X_3 = v|X_6 = q, X_8 = z, X_{11} = y, X_{12} = x) =$$

$$P(X_1, \ldots, X_3 = v, X_4 = Y_2 - q - Y_5 + z + y, X_6 = q,$$

$$X_7 = Y_5 - z - Y_7 + x, X_8 = z, X_{10} = Y_7 - x - y, X_{11} = y, X_{12} = x)$$

$$P(X_9|X_3 = v, X_6 = q, X_8 = z, X_{11} = y, X_{12} = x) = I_{\{Y_6 - q - v\}}$$

$$P(X_2|X_3 = v, X_6 = q, X_8 = z, X_{11} = y, X_{12} = x) = I_{\{Y_3 - q - v\}}$$

In this larger network the posterior distributions for the components of λ are no more trivial to compute, and neither are the previous conditional ones, which already suggests that some sort of automation needs to be achieved, since the "case by case approach" is unmanageable in all but the most simple situations.

It is possible to build an algorithm to simulate from these conditional distributions and generate values for the vector $\mathbf{X}$ and, subsequently, for the λ, but the computation of the support and of the probability masses becomes overly cumbersome when the counts of the vector $\mathbf{Y}$ are such that the range of each support is wider than a few units.

Nevertheless, it is interesting to compare the results from this exact computation to those empirically produced by the algorithm which is developed next. Therefore, extremely low values in $\mathbf{Y}$ are chosen and input, in order to generate results which can subsequently be compared with those of the automated approach. To start this simulation we do not need anything beyond the observation values and an initial guess for λ. The final results showed independence from this initial guess. Comparing the simulation results to the exact one for the small networks provides an important check of the reliability of the technique developed.

2.1.6 Some Theory

The algebraic structure of the problem is now analysed. This leads to a very general algorithm for network analysis.

The relation between the observed data and the "hidden", missing data in which the interest lies (i.e. the vector $\mathbf{X}$), can be thought in terms of an underdetermined system of linear equations:

$$\mathbf{Y} = \mathbf{A}\mathbf{X}$$

where the $(k \times n)$ matrix $\mathbf{A}$ is "longer" than "high", that is to say, it is usually $n > k$.

As a result of this indeterminism, the vector $\mathbf{X}$ can be subdivided into two subsets, one free to vary and one fully determined by the constraints, that arise once this first one has been fixed.

With this goal in mind, the matrix $\mathbf{A}$ is partitioned in two parts:

- one square invertible submatrix, $\mathbf{A}_1$
- a rectangular one, $\mathbf{A}_2$.

The columns of this second matrix correspond to the portion of $\mathbf{X}$ that can be randomly generated, from now on denoted by $\mathbf{X}_2$, while the columns of the other matrix correspond to the portion of $\mathbf{X}$ that is a deterministic function of both the $\mathbf{Y}$'s and the generated values of $\mathbf{X}_2$, and from now on will be indicated by $\mathbf{X}_1$.

In fact,

$$\begin{aligned} \mathbf{Y} &= \mathbf{A}\mathbf{X} \\ &= \mathbf{A}_1\mathbf{X}_1 + \mathbf{A}_2\mathbf{X}_2, \end{aligned}$$

so:

$$\mathbf{X}_1 = \mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2).$$

There are k deterministic variables and $n-k$ random ones. In the above set of equations, $\mathbf{A}_1$ is a $(k \times k)$ matrix, $\mathbf{A}_2$ is a $(k \times (n-k))$ matrix.

To perform the subdivision of the $\mathbf{A}$ matrix in these two sub-matrices the rank must be k. This ensures that $\mathbf{A}_1$ is invertible and doesn't introduce any loss of generality: the fact that the matrix has k linear independent rows (and thus k linear independent columns) is seen as a preliminary requirement and can be achieved by simplifying a "taller" matrix, by deleting those rows that fail to be linear independent of the others. This can be done without loss of information, as those link counts are forgotten which don't deliver any disaggregated piece of information not already provided by other counts, the former being just linear combinations of other recorded counts. So the k linear independent rows that are produced by a possibly necessary preliminary simplification of $\mathbf{A}$ ensure k linear independent columns, and different choices are then available to isolate the $(k \times k)$ matrix $\mathbf{A}_1$.

The easiest situation is one in which all the links connect an **OD** pair; this being the case, the matrix $\mathbf{A}_1$ will be easily isolated as an identity matrix.

In general, given a $(k \times n)$ matrix $\mathbf{M}$ of full rank k, the Gramm-Schmidt algorithm [18] can orthogonalize k of the n columns and thus produce a decomposition of the original matrix in a $(k \times k)$ orthogonal matrix $\mathbf{Q}$ and a $(k \times n)$ upper triangular $\mathbf{R}$, whose products gives back the original $\mathbf{M}$. The orthogonal $\mathbf{Q}$ is obtained by binding together the result of the Gramm-Schmidt process; the upper-triangular $\mathbf{R}$ contains the coefficients for the linear combinations of this orthogonal basis, that reconstruct the columns of the original matrix $\mathbf{M}$. More specifically the coefficients to re-obtain the ith column of $\mathbf{M}$ are the entries of the ith column of $\mathbf{R}$. By the properties of orthogonalization, each column of $\mathbf{R}$ is merely a function of the column of $\mathbf{M}$ that it redelivers. So there is a one-to-one correspondence between the columns of $\mathbf{R}$ and the columns of $\mathbf{M}$. Thus, the first k columns of $\mathbf{R}$ correspond reliably to k linearly

independent columns of $\mathbf{M}$, due to the upper triangular form of the matrix: each of the first k linear combinations "adds" an orthogonal coordinate to the previous one. In no case can one of the first k combinations lie in the same space of another.

The `qr()` routine in S-Plus [32] takes as input a matrix $\mathbf{M}$ and produces a list of output which includes the vector of index permutations of the columns of $\mathbf{M}$ that correspond to the column of $\mathbf{R}$; this vector is precisely what is needed for the **OD** pairs to be reordered in such a way that, after the corresponding permutation of its columns, the new matrix $\mathbf{A}$ is the plain juxtaposition of the full-rank $\mathbf{A}_1$ and the remaining part $\mathbf{A}_2$. That is, the first k indices of this result correspond to the first k columns of the $\mathbf{R}$ matrix, which is its upper triangular subsection,and to the k linearly independent columns of $\mathbf{A}$.

In the particular case under consideration, $\mathbf{M} = \mathbf{A}$, $k = c = 7$, $n = r = 12$, and, by starting with

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix},$$

the `qr(A)` call in S-Plus produces, among other output, the vector

`$pivot`

[1] 1 2 3 4 5 7 10 11 12 9 8 6.

By applying this permutation to the columns of $\mathbf{A}$,

$$[\mathbf{A}_1\, \mathbf{A}_2] = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

is obtained, where the first 7 columns are linearly independent as needed.

Once the split matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ are available, an algorithm can be set up that

1. randomly generates the subvector $\mathbf{X}_2$;
2. computes the subvector $\mathbf{X}_1$ as a function of the previous one;
3. draws from the conditional density of each λ_i separately, given the current value of the correspondent X_i.

Thus the essential pieces to be implemented in the simulation algorithm are at our disposal.

2.1.7 Convexity of the Supports for the Components of X and Irreducibility of MCMC Scheme to Be Adopted

A critical issue in the performance of a Markov Chain Monte Carlo algorithm is that of irreducibility. When simulating values for a n-dimensional vector of quantities, say $\mathbf{X}$ (for a moment using this notation in a general sense without referring strictly to the previous $\mathbf{X}$), by iterating drawing from conditional distributions of the type

$$P(X_i|X_{(i)} = x_{(i)}) \; i = 1, \ldots, n,$$

the support of these conditional distributions needs to be positive wherever that of the joint distribution is positive, that is:

$$\begin{aligned} P(X_i = x_i|\mathbf{X}_{(-i)} = \mathbf{x}_{(-i)}) &> 0 \\ \text{if } P(X_i = x_i, X_{(i)} = x_{(i)}) &> 0. \end{aligned}$$

In the present case, the question of irreducibility is considered in a setting in which the generation of the new vector of $\mathbf{X}_2$ is performed by cycling through the $n-k$ values one at a time, conditioning on the remaining $n-k-1$. In such a case, the

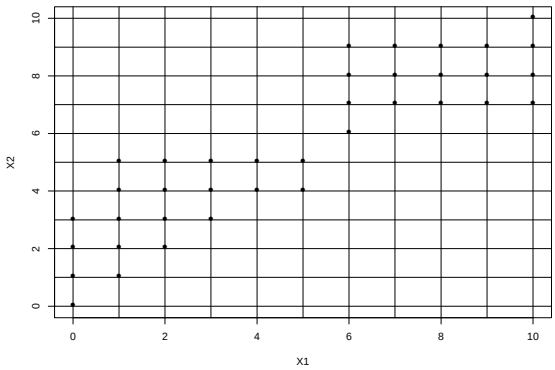


Figure 2.4: Joint support of a two-dimensional vector $\mathbf{X}$ in a case of non-irreducibility.

support of the single component is conditional on the remaining fixed components, and even if the conditional support is convex, the possibility that two or more subsets of the joint (unconditional) support be separated is considered and proved wrong, ensuring that the chain of subsequently generated values can explore the complete space.

Take for instance a situation like that described by Figure 2.4, which shows the joint support in the simplified case of a two-dimensional vector $\mathbf{X}_2$. Should the chain start in the left-bottom part of the joint support, it won't be able to visit the right-top subspace. But it is easily shown that this is impossible, given the structure of the problem at hand. That is, assuming that both $\mathbf{X}_2$ and $\mathbf{X}_2 + (1,1)'$ are feasible, either $\mathbf{X}_2 + (1,0)'$ or $\mathbf{X}_2 + (0,1)'$ or both must be feasible as well.

The proof considers first the two dimensional case, then is extended to the general k-dimensional setting; it procedes by contradiction, by showing that the two assumptions:

$$\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2) \geq 0$$

$$\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2(\mathbf{X}_2 + (1,1)') \geq 0$$

and the two others together:

$$\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2(\mathbf{X}_2 + (1,0)') < 0$$

$$\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2(\mathbf{X}_2 + (0,1)') < 0$$

are not compatible.

It can in fact be rewritten. Indicating the single elements of the matrix by a double apex:

$$\begin{aligned}
\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2(\mathbf{X}_2 + (1,1)') &\geq 0 \\
\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2 - (A_2^{11} + A_2^{12}, A_2^{21} + A_2^{22})') &\geq 0 \\
\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2 - (A_2^{11}, A_2^{21})' - (A_2^{12}, A_2^{22})') &\geq 0,
\end{aligned}$$

and then adding $\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2) \geq 0$ by assumption, leaves the inequality valid:

$$\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2) + \mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2 - (A_2^{11}, A_2^{21})' - (A_2^{12}, A_2^{22})') \geq 0.$$

Collecting and redividing the terms gives:

$$\begin{aligned}
\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2 + \mathbf{Y} - \mathbf{A}_2\mathbf{X}_2 - (A_2^{11}, A_2^{21})' - (A_2^{12}, A_2^{22})') &\geq 0 \\
\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2 - (A_2^{11}, A_2^{21})' + \mathbf{Y} - \mathbf{A}_2\mathbf{X}_2 - (A_2^{12}, A_2^{22})') &\geq 0.
\end{aligned}$$

The last cannot inequality contrasts with the two ones in the assumptions, which stated the negativity of the two addenda. That is, the first two inequalities cannot be consistent with the last two. The shape of the support is then such that irreducibility is ensured.

The proof goes along the same lines in situations in which $\mathbf{X}_2$ and $\mathbf{X}_2 - (1,1)'$ are feasible, implying that at least one of $\mathbf{X}_2 - (1,0)$ or $\mathbf{X}_2 - (0,1)$ is (or analogously, $\mathbf{X}_2$ and $\mathbf{X}_2 + (-1,1)'$ or $\mathbf{X}_2$ and $\mathbf{X}_2 + (1,-1)'$ is).

Now I extend the proof to the n-dimensional case. By assumption:

$$\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2) \geq 0$$

and

$$\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2(\mathbf{X}_2 + (1, 1, \ldots, 1)')) \geq 0,$$

where now the vectors $\mathbf{X}_2$, and $(1, 1, \ldots, 1)'$ are n-dimensional.

Analogous to what has been done above, rewrite:

$$\begin{aligned}
\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2(\mathbf{X}_2 + (1, 1 \ldots, 1)')) &\geq 0 \\
\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2 - (A_2^{11} + A_2^{12} + \ldots + A_2^{1n}, \ldots, A_2^{n1} + A_2^{n2} + \ldots + A_2^{nn})') &\geq 0 \\
\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2 - (A_2^{11}, A_2^{21}, \ldots, A_2^{n1})' - \ldots - (A_2^{1n}, A_2^{2n}, \ldots, A_2^{nn})') &\geq 0,
\end{aligned}$$

and then add n times $\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2) \geq 0$ by assumption, leaving the inequality valid:

$$n(\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2)) + \mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2 - (A_2^{11}, A_2^{21}, A_2^{n1})' - \ldots - (A_2^{1n}, A_2^{2n}, \ldots, A_2^{nn})') \geq 0.$$

As done before, we can redistribute the terms in such a way as to obtain a sum of those quantities that should be by assumption all negative, contradicting the fact that the starting point of the manipulation was a non-negative quantity, and we have just added n other non-negative ones:

$$\begin{aligned}
\mathbf{A}_1^{-1}(n(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2) - (A_2^{11}, A_2^{21}, \ldots, A_2^{n1})' &- \ldots - (A_2^{1n}, A_2^{2n}, \ldots, A_2^{nn})') = \\
\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2 - (A_2^{11}, A_2^{21}, \ldots, A_2^{n1})') + \ldots &+ \mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2 - (A_2^{1n}, A_2^{2n}, \ldots, A_2^{nn})').
\end{aligned}$$

This proves our proposition in the general case.

The important property of the convexity of the support of a single-component distribution, that is the support for $X_{2i}|\mathbf{X}_{2(-i)}$ $i = 1, \ldots, n-k$ was just mentioned above, and is formally articulated below.

It has to be shown that if two (n-k)-dimensional vectors $\underline{x}, \underline{y}$ are distinct only because of one of their components, that is $\underline{x} = (x_1, x_2, \ldots, x_i, \ldots, x_{n-k})'$ and $\underline{y} = (x_1, x_2, \ldots, y_i, \ldots, x_{n-k})'$ and both $\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\underline{x}) \geq 0$ and $\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\underline{y}) \geq 0$, then

also the vector

$$\underline{z} = (x_1, x_2, \ldots, \alpha x_i + (1-\alpha)y_i, \ldots, x_{n-k})'$$

is such that $\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\underline{z}) \geq 0$ when $\alpha \in [0,1]$.

Write $\underline{z} = \alpha\underline{x} + (1-\alpha)\underline{y}$ and substitute this in the formula

$$\begin{aligned}
\mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\underline{z}) &= \mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2(\alpha\underline{x} + (1-\alpha)\underline{y})) \\
&= \mathbf{A}_1^{-1}(\alpha(\mathbf{Y}) + (1-\alpha)(\mathbf{Y}) - \mathbf{A}_2(\alpha\underline{x} + (1-\alpha)\underline{y})) \\
&= \mathbf{A}_1^{-1}(\alpha)(\mathbf{Y} - \mathbf{A}_2\underline{x}) + \mathbf{A}_1^{-1}(1-\alpha)(\mathbf{Y} - \mathbf{A}_2\underline{y}).
\end{aligned}$$

This is a mean between two non-negative quantities by assumption; thus it is itself a non-negative quantity. Hence irreducibility is assumed for the MCMC scheme described, and to be developed in Chapter 3.

2.2 Random Routing

A model extension is now made by relaxing the assumption that all messages or trips between a specified **OD** pair take the single route specified in the 0/1 routing matrix **A**.

Vardi [37] has developed the case of Markovian routing, in which a message travelling between a specified **OD** pair exits its "current" node in the network on one of possibly several links consistent with a route to the specified destination. In communications networks, such random routing may arise as a result of control procedures to avoid or redress queuing questions. In urban road transportation networks, there typically exist one or a small number of primary routes between specified **OD** pairs, such routes being used by the bulk of the traffic, but with several additional, secondary routes used less frequently either as alternatives in times of congestion on primary routes or as occasional alternatives for other reasons. Primary and secondary routes

between a specified **OD** pair typically share a common subset of links, with a few additional links being specific to one or a few routes.

I extend the development of the fixed routing networks in order to apply the Bayesian framework to this more complex setting. But first, I describe the solution proposed by Vardi.

2.2.1 Vardi's Solution

In accordance with the Markovian assumption it is assumed that the message having **OD** address j chooses link i with probability a_{ij} conditioned only on the message's being at the source node of link i. The network can still be translated into a matrix **A** whose entries are now numbers in the interval $[0, 1]$, expressing the conditional probabilities.

Now column j of the matrix can be seen as specifying a network of alternative paths, for the **OD** destination j, each one with a certain probability that can be computed by the entries of the same column. These resulting probabilities are now to be viewed as 'absolute' probabilities of choosing that specific path, given that the message has that specific address. Despite the more complicated nature of the problem, the first and second moments of the observed data are still linear functions of the components of λ, and the moment equations express the same kind of 'linear inverse positive problem' that can be solved by the EM algorithm. The complication is now in the computation of the first and second moments $E(Y_i)$ and $Cov(Y_i, Y_j)$. It is possible though to simplify the computational procedure by exploiting the Poisson distributional assumption and the relation between Poisson and Multinomial random variables, the so called "thinning property". This way probabilities of the types:

- P_i^j that a packet with address j passes through link i
- $P_{ii'}^j$ that a packet with address j passes through both link i and i'

are easily obtained.

As a result, a matrix P having the same dimensions as $\mathbf{A}$ whose ij entry is P_i^j as defined above, is obtained and the mean and variance of the vector $\mathbf{Y}$ are computed noting that

$$Y_i = \sum_j Y_i^j \sim \mathcal{P}(\sum_j \lambda_j P_i^j)$$

where Y_i^j is the number of messages with address j passing through link i

So

$$E(\mathbf{Y}) = P\lambda \text{ and } Var(\mathbf{Y}) = P\lambda$$

are the first ingredients needed and computed.

Now for the covariance between Y_i and $Y_{i'}$ the distribution of the messages traveling along both links i , i' is needed, which is again, for each j:

$$Y_{ii'}^j \sim \mathcal{P}(\lambda_j P_{ii'}^j)\,,\ 1 \leq i, i' \leq k\,,\ j = 1, \ldots, n$$

with $P_{ii'}^j$ defined above.

By this result

$$\begin{aligned} Cov(Y_i, Y_{i'}) &= Cov(\sum_j Y_i^j, \sum_j Y_{i'}^j) \\ &= \sum_j Cov(Y_i^j, Y_{i'}^j) \\ &= \sum_j Cov(Y_{ii'}^j, Y_{ii'}^j) \\ &= \sum_j Var(Y_{ii'}^j) \\ &= \sum_j \lambda_j P_{ii'}^j \end{aligned}$$

are computed, where the simplifications are due to the independence between messages having the same **OD** address but a different path, following from the thinning property.

Once these quantities have been obtained, still by linear combinations of the observed data and the parameters to be estimated, they are equated to the sample moments which are computed from repeated measurements of the link traffic counts; the EM algorithm is applied as exemplified in Section 2.2.1, relative to the deterministic routing case.

2.2.2 A "Super Deterministic Network"

Given a specific **OD** pair, say from node A to node B, messages leaving the source node may now take different routes to their destination. The Markovian routing model simply assumes that, at any node on the way to the destination, each message exits on a link determined by a set of link choice probabilities, independently of the path taken to the current node and independently across messages. In some cases there is just one exit link possible, in other cases there may be several. As Vardi [37] does, the set-up can be summarised through a modified routing matrix that reports the link choice probabilities for all possible **OD** pairs.

As in the fixed routing case, the matrix has rows indexing links in the network and columns indexing **OD** pairs. Now however, the entries are probabilities determining the selection of links (rows) on trips between specified **OD** pairs (columns). An example matrix from Vardi [37] is represented below, with rows and columns labelled

by links and **OD** pairs:

	AB	AC	AD	BA	BC	BD	CA	CB	CD	DA	DB	DC
$A \to B$	.8	.2	.2	0	0	0	0	0	0	0	0	0
$A \to C$	.2	.8	.8	0	1	1	0	0	0	0	0	0
$B \to A$	0	0	0	1	.2	.1	1	0	0	1	0	0
$B \to C$	0	.8	0	0	.8	.1	0	0	0	0	0	1
$B \to D$	0	.2	1	0	0	.8	0	0	1	0	0	0
$C \to B$	.8	0	.2	0	0	0	.8	.8	.2	1	1	0
$C \to D$	.2	0	.8	0	0	1	.2	.2	.8	0	0	0
$D \to B$	1	0	0	0	0	0	1	1	0	.8	.8	.2
$D \to C$	0	1	0	0	0	0	0	0	0	.2	.2	.8

Here there are 4 nodes, $k = 9$ directed links and $n = 12$ **OD** pairs. As in the fixed routing case, rows and columns correspond to directed links and **OD** pairs, respectively. The entries are now conditional probabilities of traversing the link (row) during a trip between the **OD** pair (column). Consider, for example, the **OD** pair AB. Each trip from A to B has an 80% chance of moving directly along link $A \to B$, and terminating there; with the complementary 20% chance it travels from $A \to C$. Assuming it follows this latter path, it then travels along $C \to B$ directly, and terminates, with an 80% chance; otherwise, it moves along the two consecutive links $C \to D \to B$ and terminates.

By elementary probability computations various interesting marginal and conditional probabilities can be deduced , such as

- the probability that a message with address j passes through link i (unconditional),
- the probability that a message with address j passes through both link i and

link i',

- the probability that a message with address j passes through link i but not link i'.

Now consider the questions of inference about **OD** counts $\mathbf{X}$ based on observed link counts $\mathbf{Y}$, as earlier. Retain the same modeling assumptions, so that the components of $\mathbf{X}$ are conditionally independent $X_i \sim Po(\lambda_i)$ with specified independent priors on the rates λ_i.

By embedding the random routing problem in a fixed routing problem on an artificial "super-network", the theory and methods of Section 2.1 can be applied. To motivate this, consider the matrix represented above and focus on the single **OD** pair AB. As above, trips from A to B travel one of three routes: $A \to B$, $A \to C \to B$, or $A \to C \to D \to B$, with corresponding marginal probabilities 0.8, 0.16 and 0.04. So the pair AB can be viewed as comprising three artificial **OD** pairs corresponding to these three distinct routes. The X_1 trips originating at A and travelling to B can be viewed as initially assigned to one of these three routes according to the marginal route selection probabilities 0.8, 0.16 and 0.04, respectively. Once all assignments are made, the subsets of the X_1 trips travel their allocated routes, and the framework of fixed routing is regained. This leads to the general approach of constructing a super-network with fixed routing. For each column of the probabilistic routing matrix, all possible routes between the corresponding **OD** pair can be identified and listed as a subset of possible fixed routes; the corresponding probabilities assigned to these routes are easily computed. As an example, going back to the matrix depicted and considering the first column whose entries are $c(.8, .2, 0, 0, 0, .8, .2, 1, 0)$ and translates

into the three-columns submatrix

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

the three-component vector of probabilities for the three possible routes are easily obtained by taking the respective column of the $0-1$ matrix, dot-multiplying it by the original column with the Markovian probabilities, and multiplying the non-zero entries of the result. This procedure can be automated, so that by simply inputting the original probabilistic routing matrix, the corresponding enlarged 0/1 matrix and the vectors of probabilities corresponding to each **OD** address are produced. Returning to the complete example, the 9×12 probabilistic routing matrix generates the extended fixed routing matrix in Figure 2.5. Here, in addition to labelling groups of columns with their common **OD** pair, the probabilities of assignment are assigned to the columns (i.e. fixed routes) within such subsets. In this case, the original 12 **OD** pairs become 27 in the super-network. Now I turn to the issue of modeling and inferring route counts.

Consider any **OD** pair in the original network, with corresponding counts X_j. The process of creating a corresponding set of fixed routes between the pair generates some number, say n_j of such routes. Given the original probability routing matrix, the resulting probabilities $\mathbf{p}_j \equiv (p_{j,1}, \ldots, p_{j,n_j})'$ of selecting each of these fixed routes are trivially computed, and these n_j probabilities summing to one. Write $X_{j,t}$ for the number of trips that take fixed route t among the **OD** messages X_j. The counts $X_{j,1}, \ldots, X_{j,n_j}$ are conditionally multinomially distributed among the n_j fixed

	1	2	3	4	5	6	7	8	9	10	11	12
	AB	AC	AD	BA	BC	BD	CA	CB	CD	DA	DB	DC
	.8.1604	.1604.8	.2.1664	1	.2 .8	.1 .1 .8	.8 .2	.8 .2	.2 .8	.8 .2	.8 .2	.2 .8
1 A→B	1 0 0	1 1 0	1 0 0	0	0 0	0 0 0	0 0	0 0	0 0	0 0	0 0	0 0
2 A→C	0 1 1	0 0 1	0 1 1	0	1 0	1 0 0	0 0	0 0	0 0	0 1	0 1	0 1
3 B→A	0 0 0	0 0 0	0 0 0	1	1 0	1 0 0	1 1	0 0	0 0	1 1	0 0	0 0
4 B→C	0 0 0	1 0 0	0 0 0	0	0 1	0 1 0	0 0	0 0	0 0	0 0	0 0	1 0
5 B→D	0 0 0	0 1 0	1 1 0	0	0 0	0 0 1	0 0	0 0	1 0	0 0	0 0	0 0
6 C→B	0 1 0	0 0 0	0 1 0	0	0 0	0 0 0	1 0	1 0	1 0	0 1	0 1	0 0
7 C→D	0 0 1	0 0 0	0 0 1	0	0 0	1 1 0	0 1	0 1	0 1	0 0	0 0	0 0
8 D→B	0 0 1	0 0 0	0 0 0	0	0 0	0 0 0	0 0	0 1	0 0	1 0	1 0	1 0
9 D→C	0 0 0	0 1 0	0 0 0	0	0 0	0 0 0	0 0	0 0	0 0	0 1	0 1	0 1

Figure 2.5: The matrix corresponding to the super-network, in a tabular format, so that the various **OD** pairs are divided and the probabilities of following each alternative path are written at the top of each column.

routes, out of the total X_j and with route selection probabilities $\mathbf{p}_j$. Under the earlier assumption that $X_j \sim Po(\lambda_j)$, this trivially implies that the disaggregated, fixed route counts are themselves marginally independent and Poisson distributed, with $X_{j,t} \sim Po(p_{j,t}\lambda_j)$ for $t = 1, \ldots, n_j$. Hence, independently across **OD** pairs we have

$$p(X_{j,1}, \ldots, X_{j,n_j} | \lambda_j, \mathbf{p}_j) = \prod_{t=1}^{n_j} \frac{(p_{j,t}\lambda_j)^{X_{j,t}} e^{-p_{j,t}\lambda_j}}{X_{j,t}},$$

which reduces to

$$p(X_{j,1}, \ldots, X_{j,n_j} | \lambda_j, \mathbf{p}_j) \propto \lambda_j^{X_j} e^{-\lambda_j} \prod_{t=1}^{n_j} p_{j,t}^{X_{j,t}}$$

as a function of $(\lambda_j, \mathbf{p}_j)$, noting that $X_j = \sum_{t=1}^{n_j} X_{j,t}$.

The immediate consequence is the viability of the original approach to inference about route counts in the fixed routing problem in relation to a super-network. Conditional on the parameters λ and each of the $\mathbf{p}_j$ vectors, there is simply an expanded network with implied route counts for each of the $\sum_{i=1}^{n} n_j$ fixed routes. The compatibility of the independent Poisson models for the priors on route counts implies that the construction of conditional posterior distributions for route counts is structurally unchanged. Hence the components of the MCMC analysis for simulating route counts apply and will produce samples of the full set of route counts,

$$\mathbf{X} = \{X_{j,t}, t = 1, \ldots, n_j; j = 1, \ldots, n\}.$$

From these the totals X_j for each **OD** pair j are trivially deduced.

Now turn to inference on the Poisson rates. It is clear that, conditional on all route counts $\mathbf{X}$, the likelihood function for the underlying rates λ is just:

$$p(X_{j,1}, \ldots, X_{j,n_j} | \lambda_j, \mathbf{p}_j) \propto \lambda_j^{X_j} e^{-\lambda_j} \prod_{t=1}^{n_j} p_{j,t}^{X_{j,t}}$$

This is, again, of exactly the form arising in the original fixed routing problem, and so the construction of posterior samples for λ follows that development.

This discussion is all conditional on known and fixed routing probabilities; i.e., known and fixed vectors $\mathbf{p}_j$ for each **OD** pair j. Note that extensions to incorporate inference on these probabilities are essentially direct. From the likelihood above, a conditional likelihood function for each of the $\mathbf{p}_j$ vectors given imputed route counts in the super-network and the λ_j is obtained. On this basis, the likelihood function factorizes into a set of n components of the same form as the one written above, and so the iterative simulation analysis is trivially generalised by linking in a component to sample each $\mathbf{p}_j$ from the corresponding posterior distributions. Assumptions about prior distributions will depend on context, but no significant difficulties arise.

The algorithm for the random markovian routing network is thus ready to be specified along the same lines as the previous deterministic routing network.

The problem of specifying an initial point for $\mathbf{X}$ to be simulated is exactly the same as the one that is encountered in the fixed routing setting, so it has to be solved by the same heuristic approach, of which we are going to talk extensively in Chapter 3.

2.3 Highways Entrances and Exits

The estimation of **OD** traffic between different points along a highway, by means of measurements of flows along entrance and exit ramps, is another variation of the same problem.

In a simplified framework, consider a uni-directional path along which measurements of entering and exiting flows are taken for a fixed period of time. Such measurement are of a different type than those dealt with so far. For each node $i = 1, \ldots, c'$ the bi-dimensional vector $\mathbf{Y}_i = (Y_{i1}, Y_{i2})$ indicates the flows recorded entering and

exiting respectively. Two fictitious nodes are added; one representing everything upstream from the first recorded intersection as the "origin", the other everything downstream from the last recorded intersection as the "end". Thus the number of nodes become $c = c' + 2$

To summarize, consider a $c \times c$ matrix whose rows and columns correspond to the nodes and whose entries are in one-to-one correspondence with the **OD** addresses. Let each of the entries in the upper triangular portion above the diagonal be 1. The rows and columns of this matrix can be thought of as convenient indicators for the content of the observed counts in the following manner: let $\mathbf{Y}_1 = (Y_{11}, 0)$, the number of cars driving between the origin node and the first recorded node, $\mathbf{Y}_2 = (Y_{21}, Y_{22})$, the numbers of cars entering the first recorded junction and exiting it respectively, and so on up to $\mathbf{Y}_c = (0, Y_{c2})$, the number of cars driving between the last recorded junction and the end node. These vectors $\mathbf{Y}_i = (Y_{i1}, Y_{i2})$ can be expressed simply as a sum of **OD** messages corresponding to the entries of the upper triangular matrix. The first element of each vector Y_{i1} will be the sum of the **OD** messages corresponding to the 1's in the ith row of the upper triangular matrix, while the second component will be the sum of the **OD** messages corresponding to the 1's in the ith column of the matrix.

A case in which four nodes are recorded - A,B,C,D to which O and E (for Origin

and End) are added - results in the following matrix:

$$\begin{array}{c} \\ O \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{cccccc} O & A & B & C & D & E \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Identifying which of the 15 **OD** adresses contribute, for instance, to count Y_{41} can be accomplished by adding up the messages corresponding to the **OD** addresses whose entries in the 4th row of the matrix are non-zero, which in this case are CD and CE, or in other words those entering the 4th node C whose destinations can be either the next recorded node D or the End point, not recorded.

Count Y_{32} can be considered analogously. It adds up those **OD** messages corresponding to the non-zero entries of column 3; that is, OB and AB. In fact, Y_{32} indicates the number of cars exiting node B, that can only be those that were coming from the unrecorded origin, with B destination, or the ones coming from node A.

Again if the goal of the analysis is to estimate the **OD** flows for all the pairs of subsequent nodes, we are in a perfect setting for the application of our fixed routing problem, and the procedure is easily automated to come up with the **A** matrix for the problem and the corresponding **Y** vector.

2.4 Summary

In this chapter the structure of the problem is made clear and the mathematical development of a general algorithm to produce posterior estimates of the parameters

and quantities of interest by means of MCMC simulation is presented. Solutions of both the deterministic-routing case and the random-routing case are demonstrated. Examples of networks for both the deterministic and random routing cases are given. Finally a brief description of how the scheme of highway entrances and exits can be reduced to the present framework is sketched.

Chapter 3

Markov Chain Monte Carlo Development

3.1 Fixed Routing

3.1.1 Setting up the Metropolis-Hastings Algorithm

A Metropolis-Hastings algorithm is employed to explore the distribution of the variables under consideration. The well-known idea is briefly sketched below, by applying it to the case in exam, where the link counts $\mathbf{Y}$ are a linear function of the **OD** counts $\mathbf{X}$ through the matrix $\mathbf{A}$, i.e.

$$\mathbf{Y} = \mathbf{A}\mathbf{X},$$

and both the matrix and the vector $\mathbf{X}$ have been subdivided into two components: $(\mathbf{A}_2, \mathbf{X}_2)$ related to the portion to be randomly generated, and $(\mathbf{A}_1, \mathbf{X}_1)$ related to the part that is produced as a linear function of the previous.

For each i, the distribution of $X_{2i} = x$ conditional on the remaining values of the vector $\mathbf{X}_2$ is:

$$\begin{aligned} f(x|\mathbf{X}_{2(-i)}, \lambda, \mathbf{Y}) &\propto \pi(x) \cdot f(\mathbf{X}_1(\mathbf{X}_2)|\lambda, \mathbf{Y}) && (3.1) \\ &\propto Po(x; \lambda_{2i}) \cdot Pr(\mathbf{X}_1 = \mathbf{A}_1^{-1}(\mathbf{Y} - \mathbf{A}_2\mathbf{X}_2)) && (3.2) \end{aligned}$$

$$\propto \quad Po(x;\lambda_{2i}) \cdot \prod_{j=1}^{k} Pr(X_{1j} = x_{1j}) \tag{3.3}$$

$$\propto \quad Po(x;\lambda_{2i}) \cdot \prod_{j=1}^{k} Po(x_{1j};\lambda_{1j}) \tag{3.4}$$

$$\propto \quad \frac{\lambda_{2i}^{x} e^{-\lambda_{2i}}}{x!} \cdot \prod_{j=1}^{k} \frac{\lambda_{1j}^{x_{1j}} e^{-\lambda_{1j}}}{x_{1j}!}, \tag{3.5}$$

where $\mathbf{X}_2 = (X_{21}, \ldots, X_{2i-1}, x, X_{2i+1}, \ldots, X_{2n-k})$.

Call the first factor $h(x)$ and the second $\psi(x)$ so that the distribution from which to simulate is expressed by

$$f(x) \propto h(x)\psi(x).$$

If the prior $h(x)$ is chosen as the proposal distribution, that is the distribution from which the candidate value for the next step of the chain is generated, then

$$\alpha(y, x) = min\{\frac{\psi(x)}{\psi(y)}, 1\}$$

provides a Metropolis-Hastings probability of accepting the candidate value, and by accepting it, of moving from y to x. If the ratio turns out to be bigger than one, the candidate is accepted, if it is zero the step is not made, and the acceptance is randomized by the value of the ratio if this turns out to be a number in the open interval $(0, 1)$.

If the proposal distribution is not determined by the prior but, for example, by a more "neutral" distribution, like a Uniform distribution over some support, the quantities in the ratio, also known as "probing probabilities" need to include the Poisson probability for the X_{2i}^{new} in the numerator and that for the X_{2i}^{old} in the denominator, as the proposal otherwise does not include them.

In the previous notation, the ratio is now:

$$\alpha(y, x) = min\{\frac{h(x)\psi(x)}{h(y)\psi(y)}, 1\},$$

where

$$h(x)\psi(x) = Po(x, \lambda_{2i}) \cdot \prod_{j=1}^{k} Po(x_{1j}; \lambda_{1j}).$$

Replacing x by y and the new $\mathbf{X}_1$ components by the components of the old $\mathbf{X}_1$, $h(y)\psi(y)$ has the same form.

Because of the form of the acceptance/rejection ratio in both cases the range of the support of $f(x|\mathbf{X}_{2(-i)}, \lambda, \mathbf{Y})$ needs not be taken into explicit consideration. In fact, if x is not in the support it generates negative values in the new vector $\mathbf{X}_1$, in which case the numerator of the ratio becomes zero, leading to the rejection of the candidate x with certainty.

The updating of the vector $\mathbf{X}_2$ is performed by cycling over all the values X_{2i} for $i = 1, \ldots, n-k$. Once a new set of values for the vector $\mathbf{X}_2$ is produced, $\mathbf{X}_1 = f(\mathbf{X}_2)$ is computed, and, given the entire vector $\mathbf{X}$, the posterior distribution for each λ is gained separately, because by conditioning on $\mathbf{X}$ and assuming a set of independent Uniform priors, the joint distribution of λ factors in a product of n Gamma distributions, i.e. $\forall i = 1, \ldots, n$:

$$\lambda_i | X_i \sim \mathcal{G}(X_i + 1, 1)$$

As should be clear from the preceding explanation, a decisive factor in the Metropolis simulation is the choice of the proposal distribution; that is, the density from which we generate candidates that will be accepted or rejected depending on the ratio of the posterior probability of the new point and that of the old one. In the present setting, there are a number of possible choices: using the prior distribution (a Poisson) with parameters given by the current set of λ, using a Uniform proposal on a

"large-enough" support, or investing some computational effort in determining the exact support of the variable about to be simulated, given the current values of the others. A detailed discussion of these alternatives follows.

3.1.2 Poisson Proposal

A set of plausible values for the components of $\mathbf{X}_2$ and a vector of values for the λ are needed. As a first step, one of the components of the vector $\mathbf{X}_2$ is chosen, and the set of values over which it can range is determined to be either the single point zero, or a larger set of values. This is performed by considering what remains after subtracting from the observed values $\mathbf{Y}$ the values of the remaining components of $\mathbf{X}_2$ and taking the minimum among those differences that should involve the chosen component X_{2i}.

If the support is larger than the single point zero, a Poisson with parameter λ_{2i} is used as a proposal distribution, and the ratio of the two likelihood functions of the data computed, with the numerator function of the new candidate X_{2i}^{new}, and the denominator function of the old value X_{2i}^{old}, keeping everything else fixed.

More precisely, when a new value for X_{2i} is proposed, a set of values for the remaining part of the vector is already available, $\mathbf{X}_{2(-i)}$ and the vector $\mathbf{X}_1$ is computed as a function of these latter and the new X_{2i}. The likelihood of $\mathbf{X}$ is therefore just a product of Poissons, and the ratio simplifies to that of the product of Poissons for $\mathbf{X}_1^{new}$ and the product of Poissons for the $\mathbf{X}_1^{old}$, the vector $\mathbf{X}_{2(-i)}^{new}$ being the same as $\mathbf{X}_{2(-i)}^{old}$ and not including the term for X_{2i}^{new} which is already taken into account in the proposal.

3.1.3 Uniform Proposal

A Uniform distribution can be thought of as a more "neutral" proposal, giving equal probabilities to the set of points in the support. It was shown above how this choice affects the form of the probing ratio for the probability of accepting/rejecting the candidate value. The numerator now appears as:

$$Po(x, \lambda_{2i}) \cdot \prod_{j=1}^{k} Po(x_{1j}; \lambda_{1j}),$$

while the denominator has the same form for y instead of x and the old $\mathbf{X}_1$.

As for the support for the Uniform distribution, it is possible, given availability of sufficient computing time to determine the true support at each simulation step by exploiting the property of convexity, which reduces the problem to a search for the extremes of the range. For each of the X_{2i}, 0 can be taken as a safe lower bound, while the upper bound is easily found as the $\min\{Y^{(i)} - X_{2(i)} \cdot I_{(i)\in i}\}$; in other words, from all the components of $\mathbf{Y}$ in which X_{2i} appears subtract the values of the other $\mathbf{X}_{2(-i)}$ that are involved in the same sum, and take the minimum of these values. These are only conservative bounds, though, which the actual minimum and maximum values of X_{2i} may not reach. Therefore an algorithm has been set up that performs the search by a trial and error procedure, starting from the bottom at zero and from the top at the previously specified upper bound. For each value, the full vector $\mathbf{X}$ is computed, and if it shows all non-negative components, the value is accepted; if any of them turns out to be less than zero the value is either incremented (if we are looking for the minimum of the range) or decremented (if we are looking for the maximum) and the step is repeated. It is clear that the dimensions and degree of complexity of the network ultimately determine the possibility of performing this search at each step of the simulation.

After comparing the possible choices for the Metropolis step, the results in terms

of convergence seem to indicate better performances for the algorithm that uses the Poisson proposal. The results of this analysis were also checked by performing the same kind of simulation for the small network, whose exact posterior distributions are computable. The independence of the results from the initial values was tested, as well.

Summarizing the process, the algorithm permits one to input the structure of a network in terms of the $\mathbf{A}$ matrix. The algorithm then comes up with a posterior distribution for all the Poisson rates of interest, given an initial guess for $\mathbf{X}_2$ and λ.

A brief discussion of the important point of the starting guess for the vector $\mathbf{X}$ is needed.

The MCMC scheme is initialised with values of $\mathbf{X}_2$ chosen by inspection of the observed link flows, an easy task in this small network but one that will take more work in larger and more complex cases. Generally, identifying suitable initial values will involve coming up with a non-negative solution to an underdetermined system of equations. What follows is a description of the heuristic approach taken.

By using popular mathematical software (Mathematica [43]) a (possibly negative) general solution to the underdetermined system $\mathbf{Y} = \mathbf{AX}$, and the basis for the null space associated to the matrix $\mathbf{A}$ are easily found. Under the assumption that the link measurements are error-free, at least one positive solution exists. Because of the underdetermined nature of the system of equations, the set of the possible solutions constitutes a vector space, and in non-degenerate cases the existence of a multiplicity of positive solutions is ensured. By computing different linear combinations of the general solution with the basis of the null space a number of different starting values for the vector $\mathbf{X}$ were determined. Then results were compared across this widely ranging set of starting points.

For larger networks this procedure is more burdensome, and has to rely on software

able to automate the search for a positive solution to an underdetermined system of inequalities. The independence of the result from the starting guess has been established. The latter is not tied to historical data or other a-priori justifiable guesses. It has only to be a correct mathematical solution to an integer problem.

This at least is what should be used to initialize the random chain if no reliable or justified prior belief on the actual traffic intensities is available. In some cases, however, like one developed in Chapter 4 of this work, the use of historical data can prove quite valuable to eliminating ambiguity from the results.

3.1.4 Results from the Small Simulated Network

In this section the results of the analysis of the three-nodes network presented in Section 2.1.4 of Page 20 are shown. Here it is possible to compare the true posterior distributions with the results of the straight Gibbs and those of the Metropolis algorithm, as shown in Figure 3.1.

There is substantial agreement among the output of the two simulated procedures and the exact computation. The convergence has been checked by running an analysis of each output through the CODA software [3]. The following is a report of the results of several tests on the output of the Gibbs simulation, all of which are satisfactory:

- Raftery and Lewis convergence diagnostic:

$$\begin{aligned} \text{Iterations used} &= 1:10000 \\ \text{Thinning interval} &= 1 \\ \text{Sample size per chain} &= 10000 \end{aligned}$$

$$\begin{aligned} \text{Quantile} &= 0.025 \\ \text{Accuracy} &= +/-0.005 \end{aligned}$$

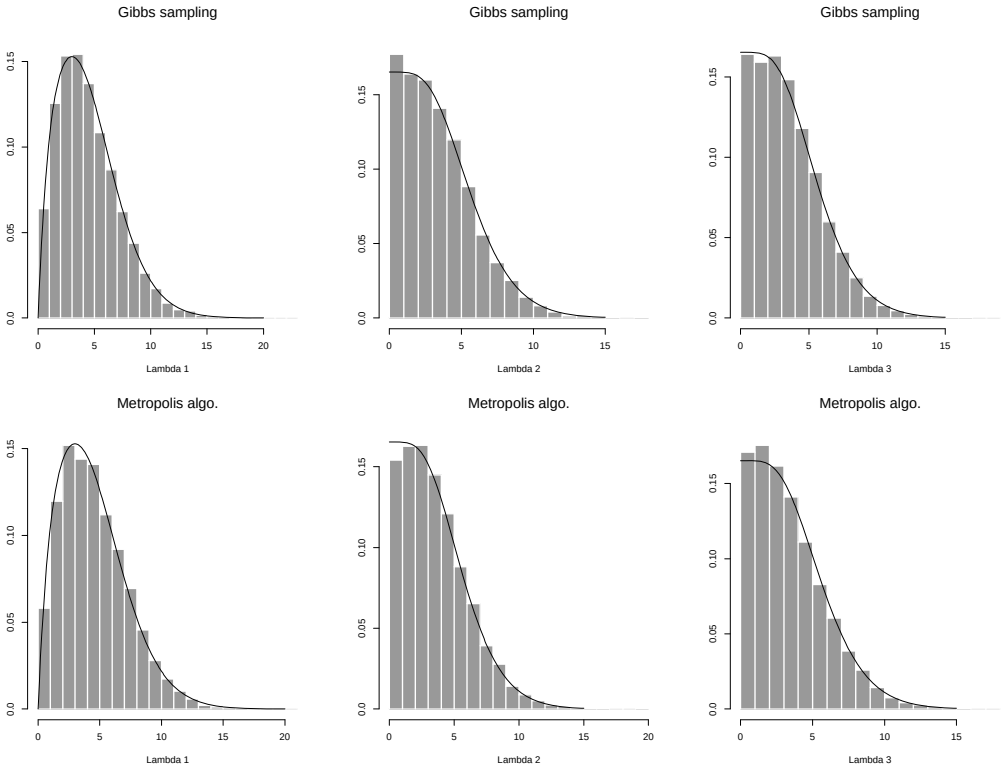


Figure 3.1: Exact posterior distributions and results from the two alternative algorithms (Gibbs and Metropolis-Hastings) for the vector λ in the network described in Section 2.1.4.

Probability = 0.95

VARIABLE	Thin (k)	Burn-in (M)	Total (N)	Lower bound (Nmin)	Dependence factor (I)
lambda 1	2	6	9266	3746	2.47
lambda 2	1	3	4520	3746	1.21
lambda 3	1	4	4635	3746	1.24

- Heidelberger and Welch stationarity and halfwidth test:

Iterations used = 1 : 10000

Thinning interval = 1

Sample size per chain = 10000

Precision of halfwidth test = 0.1

VARIABLE	Stationarity test	# of iters. to keep	# of iters. to discard	C-vonM stat.
lambda 1	passed	10000	0	0.303
lambda 2	passed	10000	0	0.407
lambda 3	passed	9000	1000	0.251

VARIABLE	Halfwidth Test	Mean	Halfwidth
lambda 1	passed	4.46	0.107
lambda 2	passed	3.42	0.104
lambda 3	passed	3.47	0.107

- Geweke convergence diagnostic:

Iterations used = 1 : 10000

Thinning interval = 1

Sample size per chain = 10000

Fraction in 1st window = 0.1

Fraction in 2nd window = 0.5

VARIABLE	z-score
lambda 1	-1.69
lambda 2	-2.65
lambda 3	2.41

Same for the result of the Metropolis simulation:

- Raftery and Lewis convergence diagnostic:

Iterations used = 1 : 10000

Thinning interval = 1

Sample size per chain = 10000

Quantile = 0.025

Accuracy = $+/-0.005$

Probability = 0.95

VARIABLE	Thin (k)	Burn-in (M)	Total (N)	Lower bound (Nmin)	Dependence factor (I)
lambda 1	2	6	8024	3746	2.14
lambda 2	2	8	11194	3746	2.99
lambda 3	2	8	10450	3746	2.79

- Heidelberger and Welch stationarity and halfwidth test:

Iterations used = 1 : 10000

Thinning interval = 1

Sample size per chain = 10000

Precision of halfwidth test = 0.1

VARIABLE	Stationarity test	# of iters. to keep	# of iters. to discard	C-vonM stat.
lambda 1	passed	10000	0	0.1300
lambda 2	passed	10000	0	0.0988
lambda 3	passed	10000	0	0.1750

VARIABLE	Halfwidth Test	Mean	Halfwidth
lambda 1	passed	4.56	0.122
lambda 2	passed	3.57	0.122
lambda 3	passed	3.43	0.121

- Geweke convergence diagnostic:

Iterations used = 1 : 10000

Thinning interval = 1

Sample size per chain = 10000

Fraction in 1st window = 0.1

Fraction in 2nd window = 0.5

VARIABLE	z-score
lambda 1	1.07
lambda 2	1.46
lambda 3	-2.05

I sketch below a brief explanation of the meaning of the tests performed.

- The diagnostic proposed by Geweke is based on standard time series methods. For each variable the chain is divided in two "windows" containing the first 10% and the last 50% of the iterates respectively. If the whole chain is stationary, the means of the values early and late in the sequence should be similar. The Z-score is the difference between these two means, divided by the asymptotic standard error of their difference (determined by spectral density estimation).

  As the chain length goes to infinity the sampling distribution of Z approaches a Standard Normal if the chain has converged. Hence values of Z which fall in the extreme tails of the distribution suggest that the chain was not fully converged during the first window. By discarding the first 10% of the observation and re-running the test one can decide on the burn-in necessity.

- Raftery and Lewis' method is intended both to detect convergence to the stationary distribution and to provide bounds for the accuracy of the estimated quantiles of functions of variables of interest.

  The CODA output reports `Nmin` the minimum number of iterations that would be needed to estimate the 2.5% quantiles to the .005 degree of accuracy with probability .95 if the samples of the chain are independent (Theoretical value based on the binomial variance, which provides a lower bound for the run-length of the Gibbs sampler).

  If sufficient iterations are available CODA reports `N`, the total number of iterations that should be run for each variable, `M` the number of initial iterations to discard as the "burn-in" and `k`, the thinning interval to be used. The final column is the ratio `I` between `N` and `Nmin`. This measures the increase in number of iterations needed to reach convergence due to dependence between the samples. Values of `I` much greater than 1.0 (Raftery and Lewis suggest `I` > 5.0) indicate problems.

- The test designed by Heidelberger and Welch is based on Brownian bridge theory and uses the Cramer-von Mises statistic to test the null hypothesis that the sampled values for each variable form a stationary process. If the null hypothesis is rejected for a given variable the test is repeated after discarding the first 10% of iterations. If it is again rejected a further 10% of iterations are discarded. This process is repeated until either a portion of the chain (of length $\geq 50\%$ of the total number of iterations) passes the stationary test or 50% of the iterations have been discarded and the null hypothesis rejected.

  If the stationarity test is passed CODA reports the number of iterations to keep, the number of initial iterations to discard and the Cramer-von Mises statistic.

  A halfwidth test is then performed by estimating the asymptotic standard error of the mean in the portion of the chain passing the stationarity test. CODA reports the mean and the halfwidth of the 95% confidence interval. If the halfwidth is less than 0.1 times the sample mean the halfwidth test is passed.

3.1.5 Results from the Larger Simulated Network

The results of the Gibbs and Metropolis algorithms are also compared for the larger 7-links, 12-**OD** pairs network, presented in Section 2.1.5.

As was mentioned in that same section, in which the conditional distributions were computed, the computational burden of the Gibbs approach demands a simulated trial with very low counts. Therefore the vector $\mathbf{Y}$ doesn't carry the original counts (from Vardi's paper) but the following set of values:

$$\begin{aligned} Y_1 &= AB = 1 \\ Y_2 &= BA = 4 \\ Y_3 &= AC = 3 \end{aligned}$$

$$
\begin{aligned}
Y_4 &= BC = 1 \\
Y_5 &= CB = 4 \\
Y_6 &= CD = 3 \\
Y_7 &= DC = 3
\end{aligned}
$$

Notice that in this instance we are not comparing "true" distributions with the result of the two simulations, but the agreement between the outputs of the two algorithms. Thus the "true" intensities of the vector λ are not relevant, but rather the plots of the marginal distributions of the components of $\mathbf{X}$ will undergo comparison.

The general solution to the underdetermined system already satisfies the non-negativity constraint, and was consequently input as the starting guess for the Metropolis simulation, producing the results shown in Figure 3.2 for the $\mathbf{X}$ vector. The results of running the straight Gibbs using the exact conditionals computed as demonstrated in Chapter 2 are displayed analogously in Figure 3.3 in the same kind of display. The agreement of the two output is satisfactory.

Also for these results the CODA convergence tests were run. As for the Metropolis output:

- Heidelberger and Welch stationarity and halfwidth test:

$$
\begin{aligned}
\text{Iterations used} &= 1:10000 \\
\text{Thinning interval} &= 1 \\
\text{Sample size per chain} &= 10000
\end{aligned}
$$

$$
\text{Precision of halfwidth test} = 0.1
$$

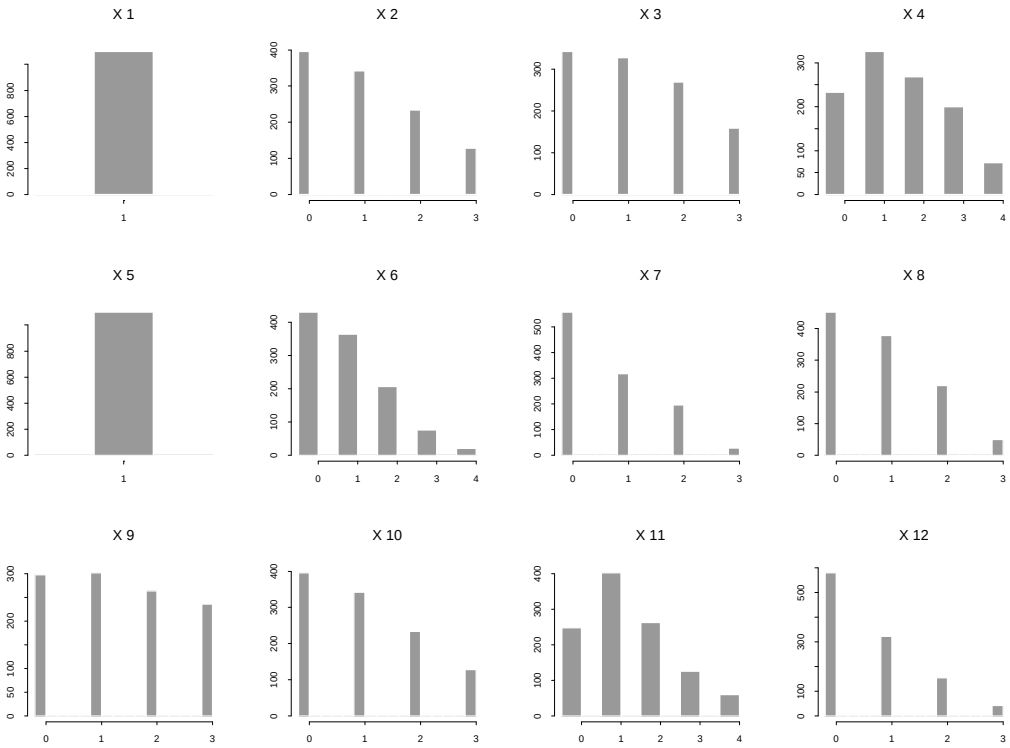


Figure 3.2: Histograms for the **X** components in the larger network of Section 2.1.5, with low counts, generated by the Metropolis algorithm.

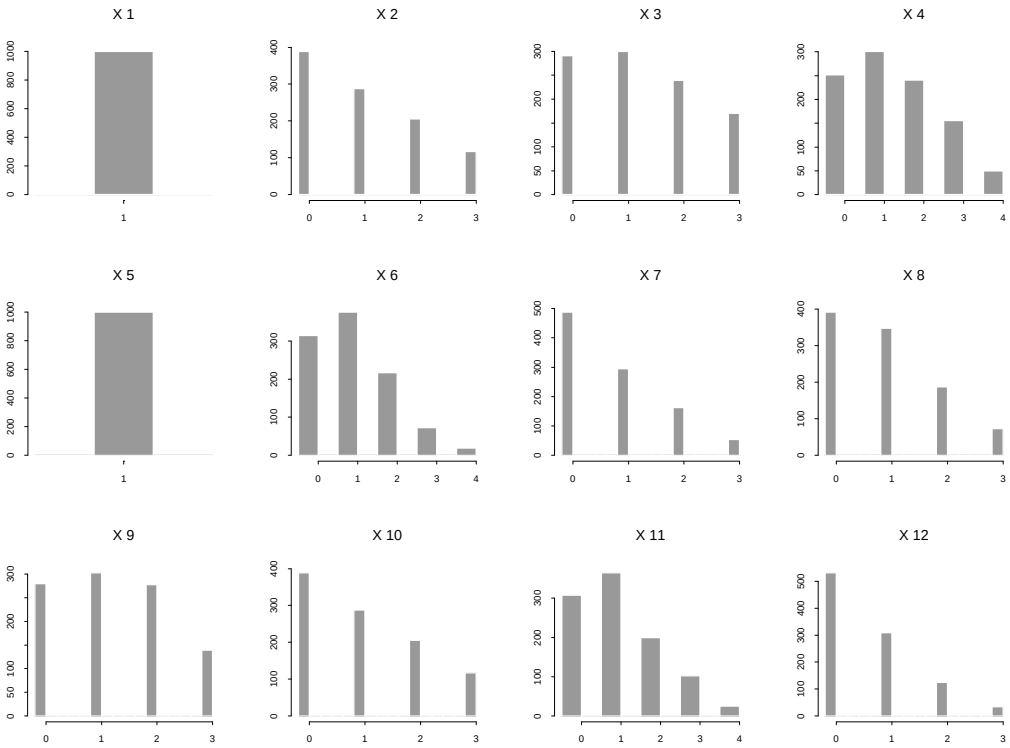


Figure 3.3: Histograms for the **X** components in the larger network of Section 2.1.5, with low counts, generated by the Gibbs algorithm.

VARIABLE	Stationarity test	# of iters. to keep	# of iters. to discard	C-vonM stat.
X 1	NA	NA	NA	NA
X 2	passed	10000	0	0.317
X 3	passed	10000	0	0.206
X 4	passed	10000	0	0.370
X 5	NA	NA	NA	NA
X 6	passed	10000	0	0.390
X 7	passed	10000	0	0.392
X 8	passed	6000	4000	0.375
X 9	passed	10000	0	0.361
X 10	passed	10000	0	0.317
X 11	passed	6000	4000	0.354
X 12	passed	10000	0	0.450

VARIABLE	Halfwidth Test	Mean	Halfwidth
X 1	NA	NA	NA
X 2	passed	1.09	0.156
X 3	passed	1.22	0.135
X 4	passed	1.59	0.152
X 5	NA	NA	NA
X 6	passed	0.99	0.133
X 7	passed	0.72	0.116
X 8	passed	0.88	0.164
X 9	passed	1.40	0.123
X 10	passed	1.09	0.156
X 11	passed	1.41	0.177
X 12	passed	0.69	0.125

- Geweke convergence diagnostic:

$$\text{Iterations used} = 1:10000$$

$$\text{Thinning interval} = 1$$

$$\text{Sample size per chain} = 10000$$

$$\text{Fraction in 1st window} = 0.1$$

Fraction in 2nd window = 0.5

VARIABLE	z-score
X 1	NA
X 2	-0.0661
X 3	2.5900
X 4	1.4600
X 5	NA
X 6	1.665
X 7	2.090
X 8	-1.123
X 9	1.228
X 10	-0.0661
X 11	2.104
X 12	-1.170

As for the Gibbs output:

- Heidelberger and Welch stationarity and halfwidth test:

Iterations used = 1 : 10000

Thinning interval = 1

Sample size per chain = 10000

Precision of halfwidth test = 0.1

VARIABLE	Stationarity test	# of iters. to keep	# of iters. to discard	C-vonM stat.
X 1	NA	NA	NA	NA
X 2	passed	10000	0	0.2020
X 3	passed	10000	0	0.1770
X 4	passed	6000	4000	0.3070
X 5	NA	NA	NA	NA
X 6	passed	10000	0	0.2690
X 7	passed	10000	0	0.3500
X 8	passed	9000	1000	0.4440
X 9	passed	10000	0	0.1540
X 10	passed	10000	0	0.2020
X 11	passed	10000	0	0.1180
X 12	passed	10000	0	0.0719

VARIABLE	Halfwidth Test	Mean	Halfwidth
X 1	NA	NA	NA
X 2	passed	1.05	0.1230
X 3	passed	1.29	0.1180
X 4	passed	1.45	0.1520
X 5	NA	NA	NA
X 6	passed	1.11	0.1070
X 7	passed	0.78	0.0845
X 8	passed	0.94	0.0907
X 9	passed	1.28	0.1020
X 10	passed	1.05	0.1230
X 11	passed	1.17	0.1070
X 12	passed	0.66	0.0771

- Geweke convergence diagnostic:

$$\begin{aligned} \text{Iterations used} &= 1:10000 \\ \text{Thinning interval} &= 1 \\ \text{Sample size per chain} &= 10000 \\ \\ \text{Fraction in 1st window} &= 0.1 \end{aligned}$$

Fraction in 2nd window = 0.5

VARIABLE	z-score
X 1	NA
X 2	0.693
X 3	0.344
X 4	-1.450
X 5	NA
X 6	2.050
X 7	0.882
X 8	-1.690
X 9	1.360
X 10	0.693
X 11	-0.679
X 12	-1.160

3.2 Random Routing

3.2.1 Results from a Small Simulated Network

A simple simulation was run, using the structure of the problem as it was presented in Section 2.2 of Chapter 2 dedicated to the random routing model. Twelve counts were generated from twelve Poisson distributions with parameters ranging from 1 to 12. The resulting vector $\mathbf{X}$ is $(1, 2, 3, 8, 6, 9, 10, 9, 11, 12, 13, 13)'$. By applying the `qr()` decomposition to the (9×27) matrix $\mathbf{A}$, the (9×9) invertible $\mathbf{A}_1$ and the remaining portion (9×18) $\mathbf{A}_2$ were separated.

The single components of $\mathbf{X}$ were subdivided into addenda whose values are proportional to the probability of the different routes. They resulted in

$$\mathbf{X}_1 = (1, 0, 0, 1, 0, 1, 1, 2, 8)'$$

and

$$\mathbf{X}_2 = (1, 5, 1, 1, 7, 8, 2, 8, 1, 2, 9, 9, 3, 10, 3, 3, 10, 0)'.$$

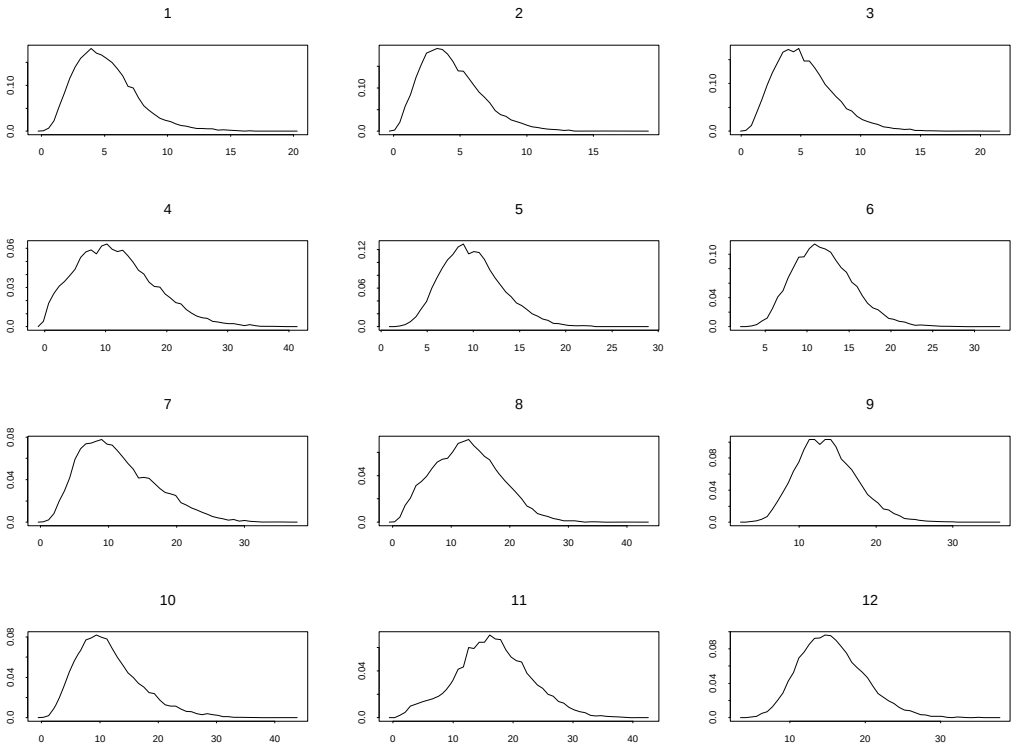


Figure 3.4: Densities of the posterior distributions of the 12 λ_i parameters, obtained by the Metropolis Simulation as described in Chapter 2, Section 2.2.

The values of the link counts, obtained from this route assignment turned out to be $\mathbf{Y} = (3, 15, 32, 10, 10, 24, 16, 25, 16)$.

Once these three vectors are computed, the algorithm is ready to be implemented, with a starting point generated again by an heuristic linear combination of the general solution to the underdetermined system of 9 equations in 27 unknown with its nullspace basis. In Figure 3.4 and in Figure 3.5 the results in terms of posterior distributions for λ and the components of $\mathbf{X}$ are shown. Recall that the true values were $\lambda_i = i$ for $i \in \{1, \ldots, 12\}$.

The distributions of the components of the two vectors λ and $\mathbf{X}$ are still quite dispersed, perhaps hinting to the need of a more precise prior distribution, but their consistently shifting masses towards the right along with the index of the component

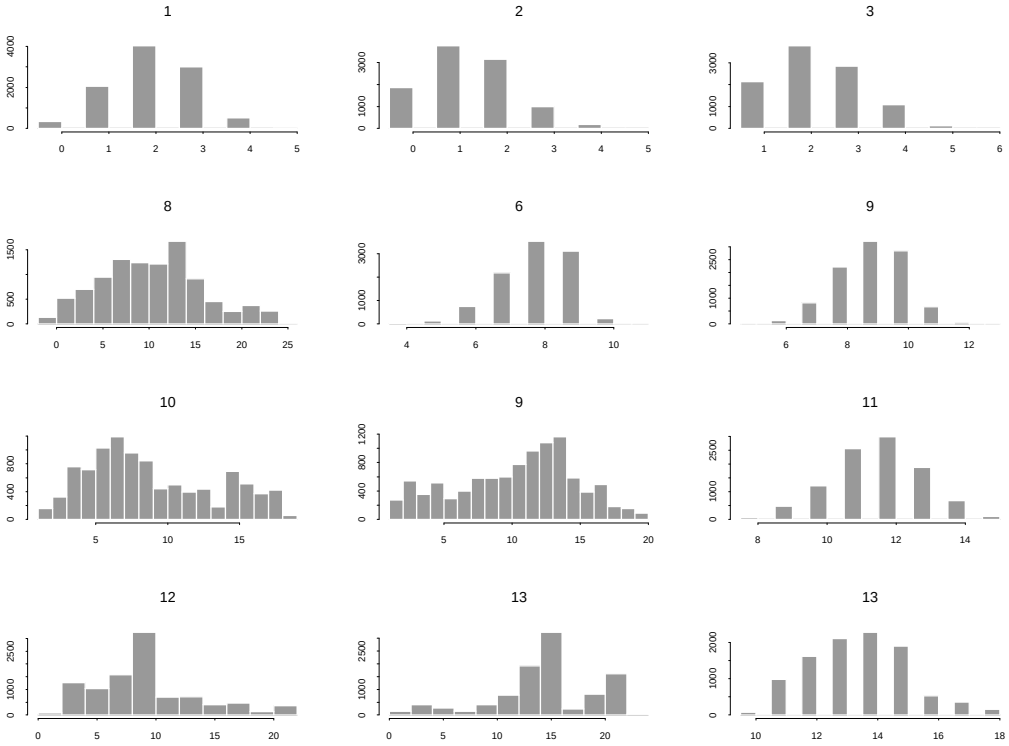


Figure 3.5: Histograms of the posterior distributions of the 12 **OD** counts, obtained by the Metropolis Simulation as described in Chapter 2, Section 2.2. The original values of the **OD** intensities chosen to check this result appear on top of each graph.

is what it must be expected given how the values were generated.

3.3 Summary

In this chapter the general algorithm is set up and put into operation. The two deterministic-routing networks presented in Chapter 2, Section 2.1.4 and Section 2.1.5 are analyzed. The results are compared with the exact posterior distributions when available and with the results produced by the straight Gibbs sampling approach described in those same sections. The results prove satisfactory in terms of agreement of the posterior distributions. Different choices of proposal distributions are discussed. Convergence of the chains is verified. A simple random-routing example is analyzed along the same lines, and the algorithm is found to perform satisfactorily in this case as well.

Chapter 4

The Monroe Network

4.1 First Analysis and Troublesome Findings

After developing the abstract approach to the problem, and measuring its performances on artificial networks of small size and balanced **OD** flows, the algorithm is applied to data generated by a traffic flow simulator, capable of approximating a real world situation.

A study group in transportation planning at the engineering department of North Carolina State University at Raleigh makes use of an operating simulation model developed at Queen's University, Canada, whose aim is to mimic what happens in a real network of roads. The program INTEGRATION [35] operates at a micro perspective, simulating the behavior of each car in the network, its interactions with the other cars, real world circumstances like traffic jams, accidents, traffic light cycles, merging of lanes, left and right turns, etc. For our purposes here the data of a run of the simulator over a network representing segments of roads interlinked in the township of Monroe, NC are of particular interest.

The model generated flows between 64 **OD** pairs, on the basis of historical estimates of the intensities of traffic between them. Inputs of cars in the network are

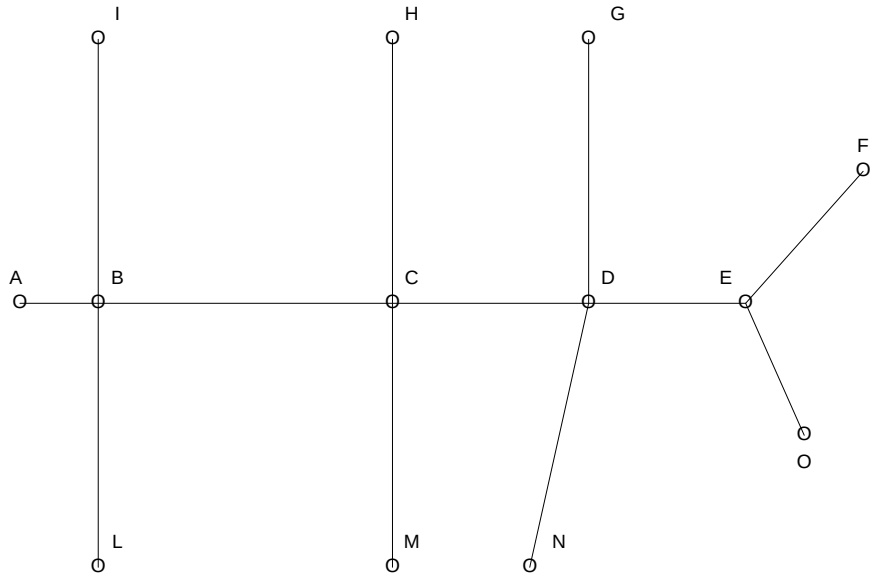


Figure 4.1: The configuration of the Monroe network.

based on those estimates, and the simulation is run for one hour, producing a complex output. As a part of it, the complete set of counts along each link, from the start when no cars were in the network, to the end when no cars were left is available. Given these recorded counts on the links, the model proposed here produces inference for the mean traffic intensities of each of the intervening **OD** pairs, and for the actual flows. How these inferences compare to the real known intensities used as input for the traffic simulator is the question addressed here.

The structure of the network is shown in Figure 4.1 All the couples of end-point nodes constitute **OD** pairs except:

$$(M,G),(M,L),(G,M),(G,L),(N,L),(L,M),(L,G),(L,N).$$

The links are to be considered as the segments joining adjacent nodes, in both direc-

tions. There are 24 such links. Thus the **A** matrix is by initial construction 24 by 64, but turns out to be not of full rank, which signals the redundancy of some link counts, with respect to the set of **OD** pairs, the object of the estimation. I discussed in Chapter 2, Section 2.1.6 the possibility of reducing the matrix to full rank without loss of information. After deleting a sufficient number of rows to achieve full-rank, the matrix **A** is 20 by 64; therefore the algorithm is going to generate values for a 44-dimension vector $\mathbf{X}_2$ and obtain a 20-dimension $\mathbf{X}_1$ as a linear function of the previous one.

The actual link counts are listed below:

$$
\begin{array}{rrr}
Y_1 = & AB = & 1980 \\
Y_2 = & BC = & 1966 \\
Y_3 = & CD = & 1788 \\
Y_4 = & DE = & 2600 \\
Y_5 = & EF = & 2100 \\
Y_6 = & FE = & 1816 \\
Y_7 = & ED = & 2880 \\
Y_8 = & DC = & 2052 \\
Y_9 = & CB = & 1954 \\
Y_{10} = & BA = & 1772 \\
Y_{11} = & IB = & 338 \\
Y_{12} = & LB = & 284 \\
Y_{13} = & HC = & 306 \\
Y_{14} = & MC = & 176 \\
Y_{15} = & GD = & 68 \\
Y_{16} = & ND = & 1000 \\
Y_{17} = & OE = & 1104 \\
Y_{18} = & BI = & 488 \\
Y_{19} = & CM = & 274 \\
Y_{20} = & DN = & 926
\end{array}
$$

Even a cursory reading of these numbers reveals a fundamental disequilibrium in magnitude, which can be explained only by a great diversity among the intensities of the different **OD** pairs. But there are also very close values in the different links. As will be shown, this pattern in the data introduces systematic biases in the likelihood

functions, due entirely to the structure of the network and assumed distributional forms. Dealing with this issue necessitates a sounder evaluation and use of informed prior distributions of the **OD** intensities. Only by introducing this additional amount of information will the analysis be driven in the "right direction", as I shall elaborate.

By the heuristic method a number of different starting values for the **OD** pairs were computed and the posterior analysis simulation run for a large number of iterations to cope with a much more complex structure of dependency among link counts, which unavoidably introduces a large dependency between subsequent simulated values of the random vector $\mathbf{X}_2$. The chains move slowly now, but after a number of different trials, and after checking the consistency of the outputs of different chains, initialized at different starting points, the results have been considered reliable when coming from a run of 1000000 iterations, of which the last 500 values of each batch of 10000 were saved. This decision was intended to avoid burdensome computations. It was verified that the quality of the inference would not change when considering the whole simulated chain, rather than only these equally spaced samples.

The result was consistently the same for the different starting points. In Figure 4.2 a sequence of 64 boxplots is used to represent the posterior distributions for the components of the $\mathbf{X}$ vector whose true values are represented by dots. This result, however, consist of a large number of poorly fitted values, most of which are situated dramatically in the tail of the obtained empirical posterior distribution. As a consequence of this outcome a number of different trials were undertaken, in hopes of determining the cause of this failure in correctly estimating the $\mathbf{X}$ values.

Initially, however, a "cheating" version of the algorithm was run. This began the simulation at the true values of each **OD** pair, in order to verify that the result remained the same. These "good" starting values seemed not to be able to hold the simulation "in place", and the final results were exactly the same as before, producing

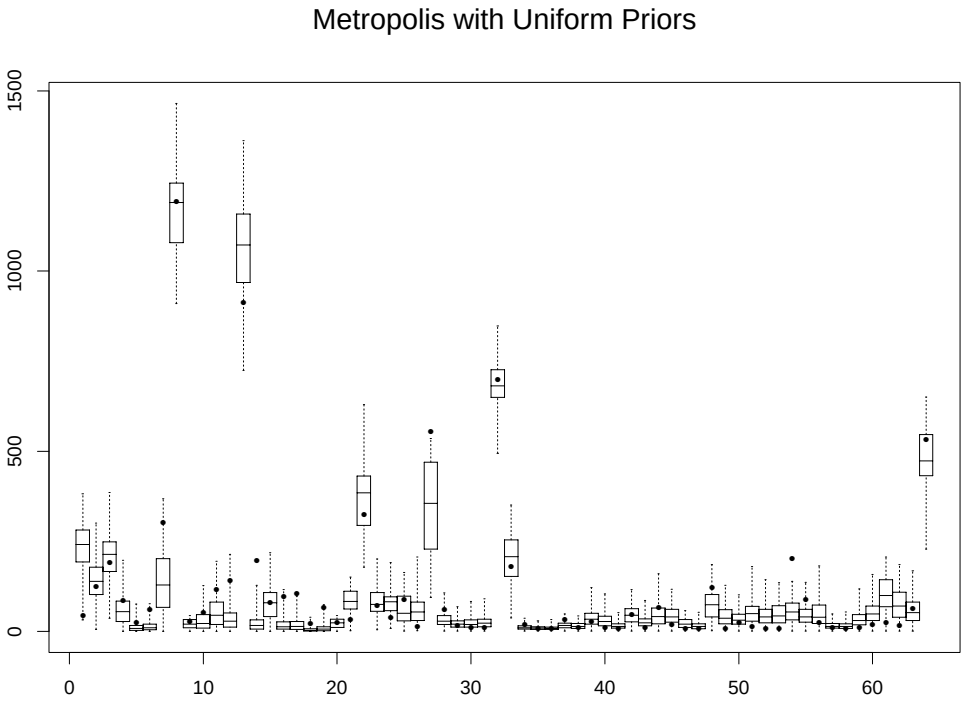


Figure 4.2: Posterior distributions for the components of $\mathbf{X}$, from the first Monroe network analysis. The point represents the actual value used by the network flow simulator.

the same posterior distribution. This is reassuring in regard to the convergence and independence of the result from the starting point of the chain, but is nonetheless an indication that the true starting points sit in a region of the **X** space that is very poorly supported under the joint posterior distribution.

4.2 Eliminating the Bad Eggs

As the first idea, I tried to detect some common characteristics among the **X** values whose fit was worse than the average. This was however unsuccessful as in the incriminated subset of **OD** pairs there are both very long paths, involving a high number of links to complete their way, and very short ones, made up by only a couple of links. There is no direct or inverse relation between the fact that one of the **OD** pairs belongs to this subset and the reverse one does or does not.

A second investigation attempted to control variability in the posterior by fixing the values of this subset of X components and thereby reducing the dimension of the randomly generated vector from 44 to 33. Changes in the posterior for the remaining components were expected. The fixed values were those of the following **OD** pairs:

$$
\begin{aligned}
AI &= 38 \\
AN &= 56 \\
AO &= 296 \\
FN &= 110 \\
FI &= 136 \\
IL &= 190 \\
HM &= 92 \\
MH &= 100 \\
NG &= 60
\end{aligned}
$$

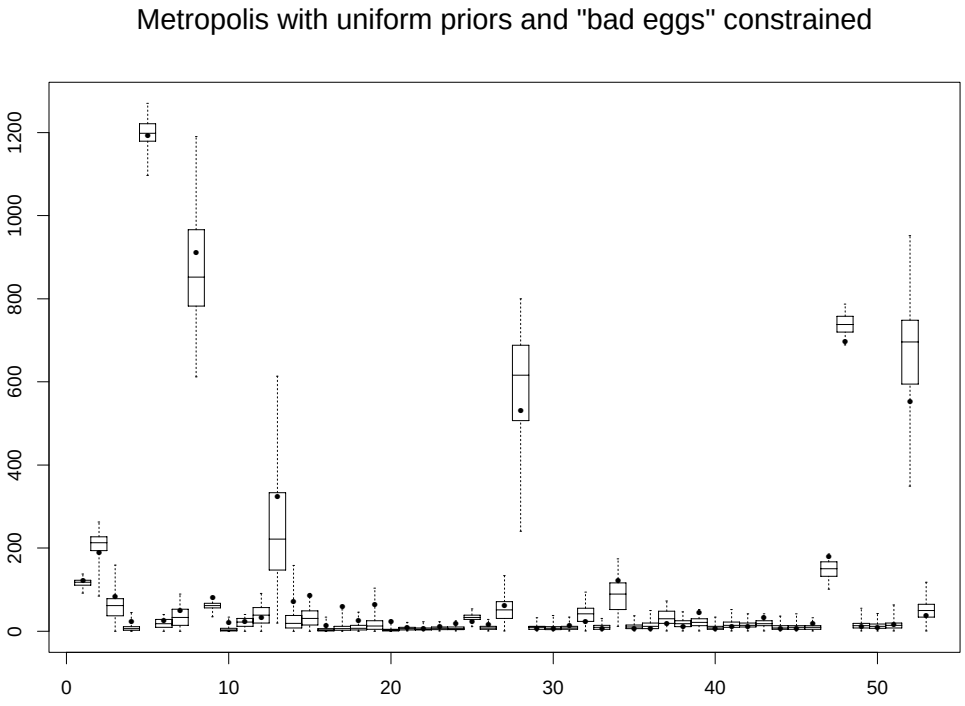


Figure 4.3: Posterior distributions for the components of **X**, after setting constant the values of the poorly fitted ones. The point represents the actual value used by the simulator.

$$NM = 196$$

and

$$NI = 84.$$

Again the result was not particularly interesting. Of course the estimated posterior distributions now have less dispersion than before, but the values that were poorly fitted remain so, and there is likewise no change among the relatively well estimated group of values. Figure 4.3 shows the result in terms of boxplots of the posterior distributions, to be compared with Figure 4.2.

4.3 Constraining the Support of the Poisson Parameters for Small OD Counts

A second investigation of this bias problem involved setting an upper bound on the support of the Uniform prior for the Poisson parameters controlling the behavior of **OD** counts with small values. With respect to the magnitude of the **OD** incriminated, this upper bound was chosen as the following list explains:

1. if $X_i <= 6$, the upper bound is chosen as 15;
2. if $6 < X_i <= 14$, the upper bound is chosen as 30;
3. if $14 < X_i <= 22$, the upper bound is chosen as 45;
4. if $22 < X_i <= 34$, the upper bound is chosen as 65;
5. if $34 < X_i < 60$, the upper bound is chosen as 120;
6. if $X_i >= 60$, the upper bound is not applied.

The posterior for the parameter λ_i now becomes a truncated Gamma, over the support specified, i.e. limited, by the prior.

An additional step in the algorithm was added to perform the generation of such truncated random variables according to the use of an auxilliary variable proposed in a recent paper by Cumbus, Damien & Walker [8]. What follows is the detailed construction for a generic variable λ whose distribution is proportional to a Gamma density of parameters $(x+1, 1)$ on the support $(0, m)$, that is

$$p(\lambda) \propto \lambda^x e^{-\lambda}, \text{ for } 0 < \lambda < m.$$

Consider a variable y whose distribution depends on λ in the following sense:

$$y|\lambda \sim \mathcal{U}(0, g(\lambda)),$$

where $g(\lambda)$ is a function still to be defined. Now write the joint distribution of the pair (λ, y) as

$$p(\lambda, y) \propto \frac{1}{g(\lambda)} \lambda^x e^{-\lambda} I\{0 < \lambda \leq m,\, 0 \leq y \leq g(\lambda)\}$$

from which the conditional distribution is easily derived, as

$$p(\lambda|y) \propto \frac{1}{g(\lambda)} \lambda^x e^{-\lambda} I\{0 < \lambda \leq m, \lambda \geq g^{-1}(y)\},$$

under the hypothesis that $g(\cdot)$ is a non-decreasing function, or rather

$$p(\lambda|y) \propto \frac{1}{g(\lambda)} \lambda^x e^{-\lambda} I\{0 < \lambda \leq m, \lambda \leq g^{-1}(y)\},$$

if $g(\cdot)$ is non-increasing.

So, by choosing $g(\lambda) = e^{-\lambda}$ the latter simplifies as

$$p(\lambda|y) \propto \lambda^x I\{\lambda \leq \min(m, \log(y))\}.$$

The algorithm of the Metropolis simulation now consists of an additional step at each simulation of a value λ_i. For each i the section of the chain that performs the simulation of λ is thus:

- if x_i is larger than 60, simply generate λ_i as before from the $Gamma(x_i + 1, 1)$
- if the value of x_i induces a limit on the support of λ_i,

  1. generate y_i given the current value of λ_i from a Uniform on $(0, e^{-\lambda_i})$
  2. generate λ_i from the cumulative distribution function

$$F(\lambda|x, y) = (\frac{\lambda}{m^*})^{x+1}$$

  where m^* here stands for the minimum between $(m, -log(y))$. Notice that this last generation is made easy by the invertiblity of $F(\cdot)$.

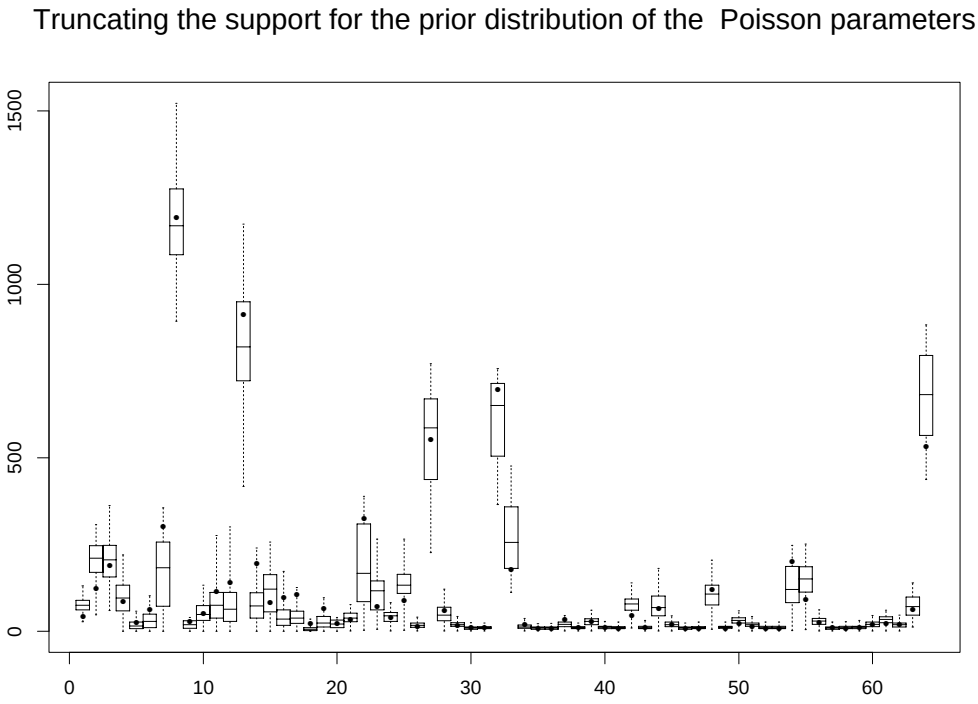


Figure 4.4: Posterior distributions for the components of $\mathbf{X}$, after constraining the values of the Poisson parameters. The point represents the actual value used by the simulator.

In general this approach induces the posterior to better support the constrained $\mathbf{X}$ values, but the components of $\mathbf{X}$ whose Poisson parameter was left free to range ended up with consistently larger estimated means with respect to the results of the previous analyses. Notice that, contrary to what might be expected, the distribution of the constrained parameters didn't pile up against their upper bound, suggesting a deep structural "problem" rather than just a flaw in the algorithm. The results are shown in Figure 4.4.

4.4 Uniformly Small Counts

As an additional check, new data for the same network were simulated, this time consisting of uniformly small counts for all the **OD** pairs.

The list of the new counts for the 20 links is reported below:

$$
\begin{array}{lll}
Y_1 = & AB = & 83 \\
Y_2 = & BC = & 148 \\
Y_3 = & CD = & 161 \\
Y_4 = & DE = & 140 \\
Y_5 = & EF = & 78 \\
Y_6 = & FE = & 79 \\
Y_7 = & ED = & 151 \\
Y_8 = & DC = & 172 \\
Y_9 = & CB = & 140 \\
Y_{10} = & BA = & 69 \\
Y_{11} = & IB = & 61 \\
Y_{12} = & LB = & 52 \\
Y_{13} = & HC = & 92 \\
Y_{14} = & MC = & 45 \\
Y_{15} = & GD = & 54 \\
Y_{16} = & ND = & 78 \\
Y_{17} = & OE = & 92 \\
Y_{18} = & BI = & 72 \\
Y_{19} = & CM = & 61 \\
Y_{20} = & DN = & 69
\end{array}
$$

Following the MCMC analysis, the posteriors do not so dramatically conflict with the actual **X** values, as Figure 4.5 demonstrates. This indicates in the large link counts of the original problem a possible source of bias, and this direction will be pursued further, in the following stages of the analysis.

4.5 A Shorter Run of the Traffic Simulator

In an important last trial the Monroe road network simulator ran the network again over a shorter interval of time, such that the traffic intensities maintained their rela-

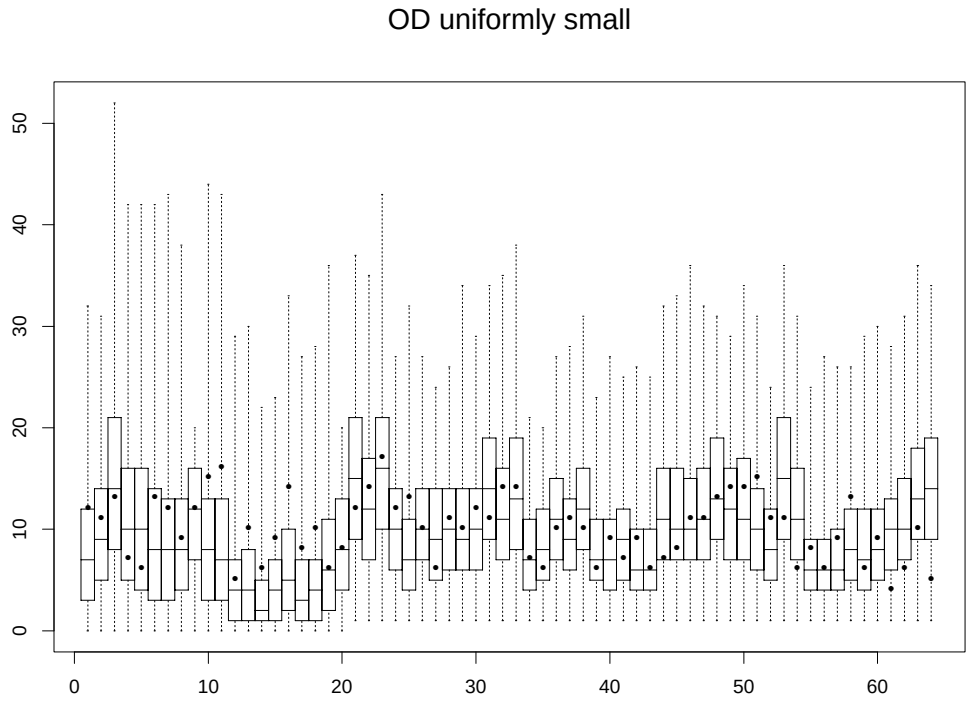


Figure 4.5: Posterior distributions for the components of $\mathbf{X}$, from data generated by low mean Poisson distributions. The point represents the actual value generated.

tive orders of magnitude. Thus although there were still significant differences among the different intensities, the $\mathbf{X}$ values remained uniformly smaller.

The new link counts, for an observation period lasting fifteen minutes, are listed below:

$$
\begin{array}{lll}
Y_1 = & AB = & 245 \\
Y_2 = & BC = & 238 \\
Y_3 = & CD = & 208 \\
Y_4 = & DE = & 302 \\
Y_5 = & EF = & 242 \\
Y_6 = & FE = & 222 \\
Y_7 = & ED = & 345 \\
Y_8 = & DC = & 238 \\
Y_9 = & CB = & 221 \\
Y_{10} = & BA = & 193 \\
Y_{11} = & IB = & 38 \\
Y_{12} = & LB = & 34 \\
Y_{13} = & HC = & 35 \\
Y_{14} = & MC = & 19 \\
Y_{15} = & GD = & 6 \\
Y_{16} = & ND = & 120 \\
Y_{17} = & OE = & 136 \\
Y_{18} = & BI = & 53 \\
Y_{19} = & CM = & 32 \\
Y_{20} = & DN = & 111
\end{array}
$$

The results of analysis are shown for the same subset of 16 parameters shown before, in Figure 4.6. As is evident nothing has dramatically changed. At this point a few hints from the different analyses performed indicate a clear path to follow: the source of bias seems to be

- either a uniformly large size of the link counts
- or the presence of unbalanced values among them, i.e. large link counts together with small link counts.

The following section finds an explanation for the rule governing the consistently biased results.

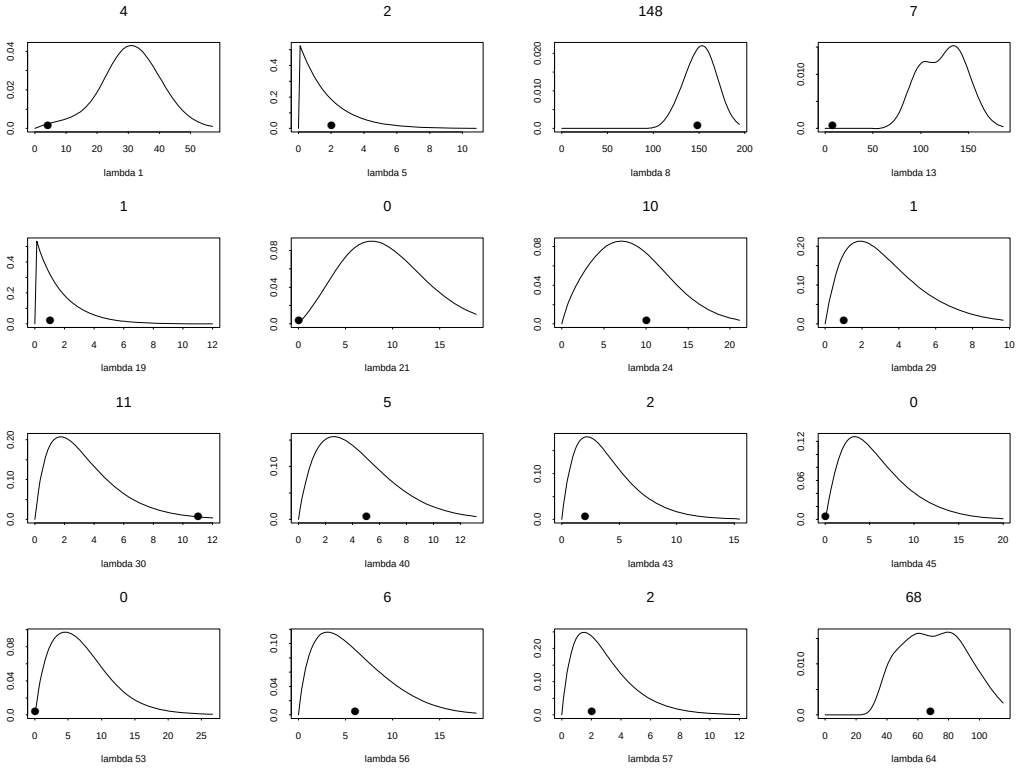


Figure 4.6: Posterior distributions for 16 components of λ, in the case of a fifteen minute interval recording. The point on the x-axis represents the actual value used by the simulator, which appears above the plot.

4.6 The Toy Network, again, Some Answers, and Better Results.

The simple 3 nodes network of Section 2.1.4, Figure 2.1 provided a clearer view of the weaknesses and ambiguities of the model, investigated so far in this chapter.

Again, consider the network represented by the matrix:

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix},$$

which is the same as writing

$$\begin{aligned} Y_1 &= X_1 + X_3 \\ Y_2 &= X_2 + X_3. \end{aligned}$$

Now re-express the equations in a way that clearly points at the indeterminacy of the problem:

$$\begin{aligned} Y_1 &= (X_1 + a) + (X_3 - a) \\ Y_2 &= (X_2 + a) + (X_3 - a). \end{aligned}$$

Here there are two potentially critical situations: a common value of Y_1, Y_2 (or, less strictly, close values for the two link counts), and/or large counts for both of them.

On the contrary, a situation that shows a big count Y_1 together with a low count Y_2 clearly indicates that the common **OD** count X_3 must take low values, and correspondently the other two counts (X_1, X_2) must take large ones. Thus the range of indeterminacy a will be limited.

Now consider carefully the situations designated dangerous above. Large and close counts (Y_1, Y_2) mean a large value of a. Moreover the Poisson assumption, through

the likelihood function, turns out to favor a large assessment for the common count X_3 and, subsequently low values will be favored for the counts X_1, X_2. In fact, the likelihood function computed for the triplet $(X_1, X_2, X_3|Y_1, Y_2)$ in cases in which the values of the link counts are close, expresses implicitly the probability that X_1, X_2 – independently Poisson distributed – assume close values. But the variance of the Poisson distribution is the same as its mean and it is clear that $X_1 \simeq X_2$ is more probable under low Poisson parameters rather than high.

I conducted a short additional experiment using the toy network. I took $X_1 \sim \mathcal{P}(\lambda)$, $X_2 \sim \mathcal{P}(\lambda)$ and $X_3 \sim \mathcal{P}(\mu)$, $Y_1 = Y_2 = y$, and discretise for demonstration purposes the values of the parameters over the grid $[0, 1, 2, 3...15]^2$. Now for each pair (λ, μ) the likelihood function is computed as proportional to

$$\sum_{x_3=0}^{y} \frac{e^{-2\lambda}\lambda^{y-x_3}}{[(y-x_3)!]^2} \frac{e^{-\mu}\mu^{x_3}}{x_3!}$$

The following substitutions in the distributional choice for the vector $\mathbf{X}$ can then be made:

- a Normal distribution with mean λ and variance λ for both X_1, X_2 and a Normal with mean μ and variance μ for X_3;
- a Normal distribution with mean λ for both X_1, X_2 and a Normal with mean μ for X_3. This time both the distributions had a common value for the variance, arbitrarily chosen as 25;
- a Normal distribution with mean λ and variance μ for both X_1, X_2 and a Normal with mean μ and variance λ for X_3.

In Figure 4.7 the results are shown for a value $y = 10$. As can be seen, the situation stays asymmetric for the first choice of Normals, becomes symmetric when

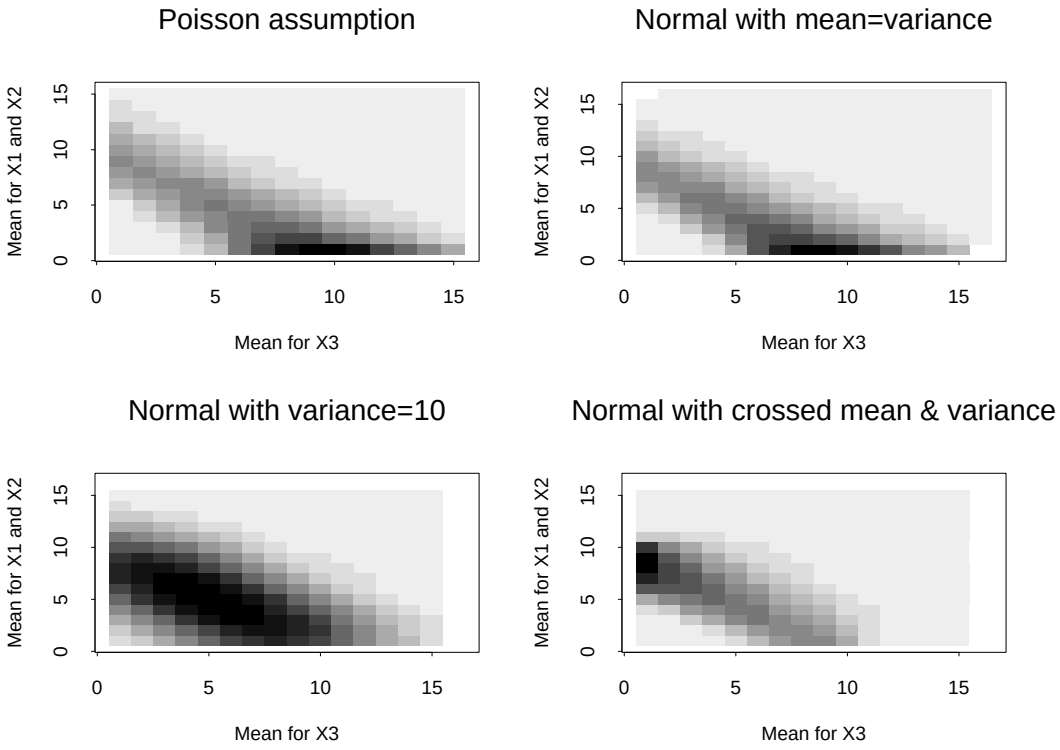


Figure 4.7: Discretized likelihood for the original Poisson choice and for the three approximations by Normal distributed **OD** counts.

the variances are the same, and gets reversed when the variances are interchanged. This proves the relevance of the Poisson distributional choice in getting those results biased consistently in the same direction.

Notice that this asymmetry in the behavior of the likelihood, perfectly justifiable from a mathematical point of view, doesn't have any substantial justification in the problem setting, where statistical independence of the **OD** intensities is assumed. Why should the situation that turns out to be the only "highly likely" be considered "better" than the one in which X_1, X_2 are high and X_3 low, or all three have almost the same "mean" value?

This is where the Bayesian approach can be exploited to the fullest. In this setting, where information, or prior opinions, about the single traffic intensities are

usually available to guide the posterior analysis, it is possible to directly address this "biased likelihood" issue through the prior.

The survey of the existing literature in the transportation engineering field (Chapter 1, Section 1.1) has already pointed out how the work in that area has been performed by assuming initial or target **OD** matrices to guide the estimation of the actual process underlying the data. Access to historical data or some kind of previous survey conducted along the road segments of the Monroe network is likely to be available. In fact, it was used to impute the traffic intensities for the simulator. This way an initial view of the pattern of traffic can be obtained and used to set up priors that are no longer non-informative. To illustrate, assume a Gamma prior for each λ_i, with independent marginals $\lambda_i \sim \mathcal{G}(\alpha_i, \beta_i)$. Then the analysis is still of a conjugate type, and doesn't introduce difficulties of computation. Where the posterior distribution of $\lambda_i|X_i$ was $\mathcal{G}(X_i + 1, 1)$ it is now $\mathcal{G}(\alpha_i + X_i, \beta_i + 1)$.

The choice of the prior can easily take into account more or less precise ideas about the different λ's, through the choice of the parameters. Say that $\lambda_i \sim \mathcal{G}(\frac{\mu_i}{a}, \frac{1}{a})$. That is, $\alpha_i = \frac{\mu_i}{a}$ and $\beta_i = \frac{1}{a}$. This way the distribution is centered around the chosen mean μ_i, and the variance is related to the magnitude of the mean, through the parameter a since $E(\lambda_i) = \mu_i$ and $Var(\lambda_i) = a\mu_i$. By different choices in terms of the Gamma parameter a more homogeneous values for the variances can be assigned, or, at the other extreme, the diversity carried on even further by choosing for each **OD** pair a specific couple of parameters. All these approaches were investigated: different values for the parameter a, different common values of the variance, different choices for the individual parameters. In this manner an informal sensitivity analysis was undertaken.

Figure 4.8 through Figure 4.10 show the results of three simulations conducted by using mean values within 10 unity from the true **OD** generated by the simulator,

with three different choices for the variances:

- value for a equal to 50,
- value of the variance fixed and equal to 500 for all the **OD** pairs,
- value of the variance tailored, by eye-ball inspection, to each **OD** pair with respect to the true value.

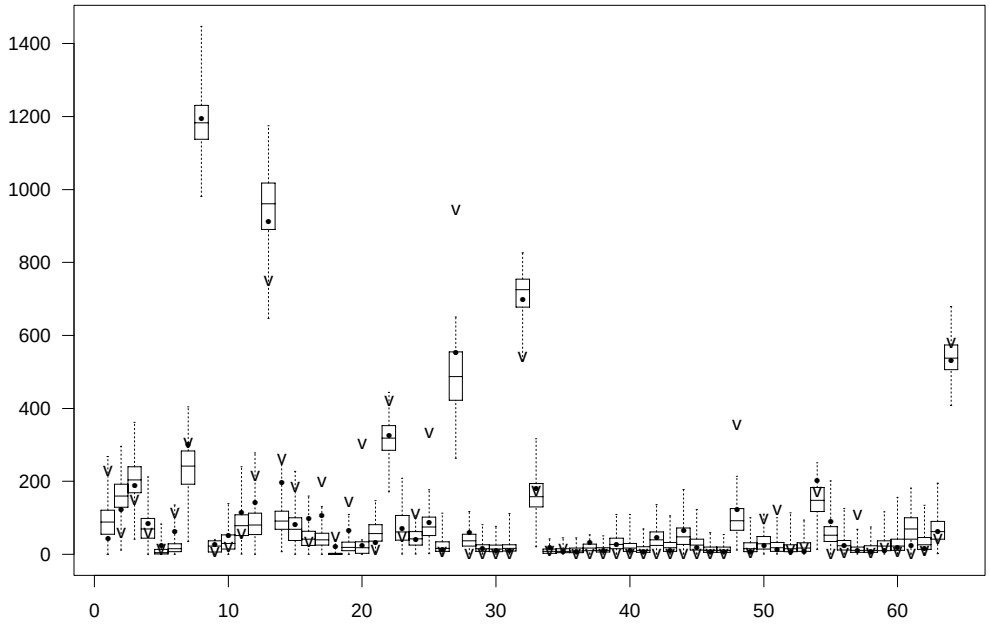


Figure 4.8: Posterior distributions for the components of $\mathbf{X}$, starting from Gamma priors on λ_i with $a = 50$. The point represents the actual value used by the simulator. The "v" comes from running Vardi's algorithm.

What appears represents a restatement of the need for informative priors when dealing with this problem. Notice in particular the "v" in the graphs. We implemented Vardi's EM algorithm suggested as a way of solving the problem. Because the

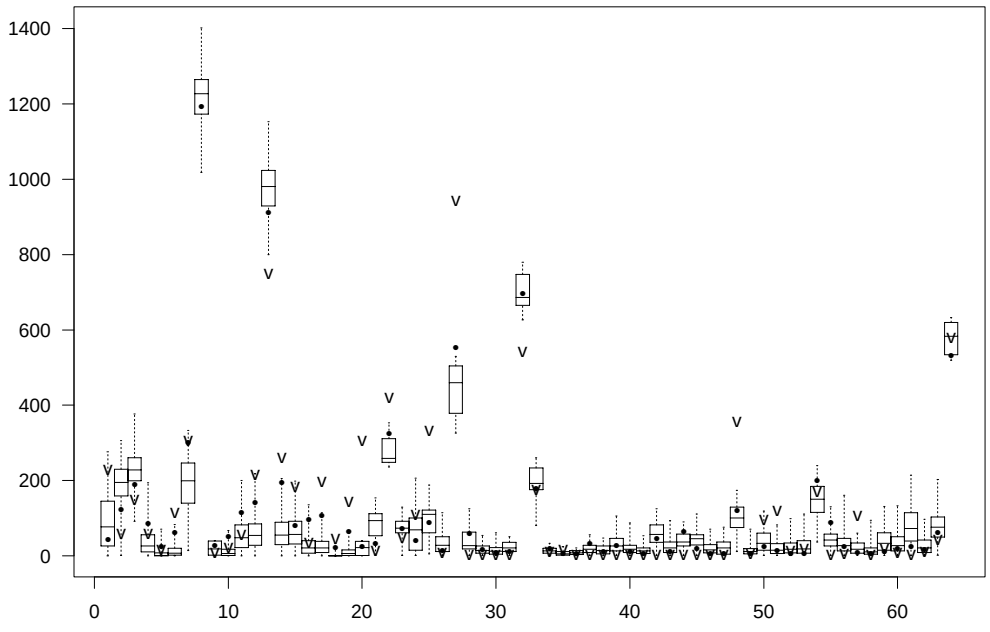


Figure 4.9: Posterior distributions for the components of $\mathbf{X}$, starting from Gamma priors on λ_i with variance set to 500 for each of the components of λ. The point represents the actual value used by the simulator. The "v" comes from running Vardi's algorithm.

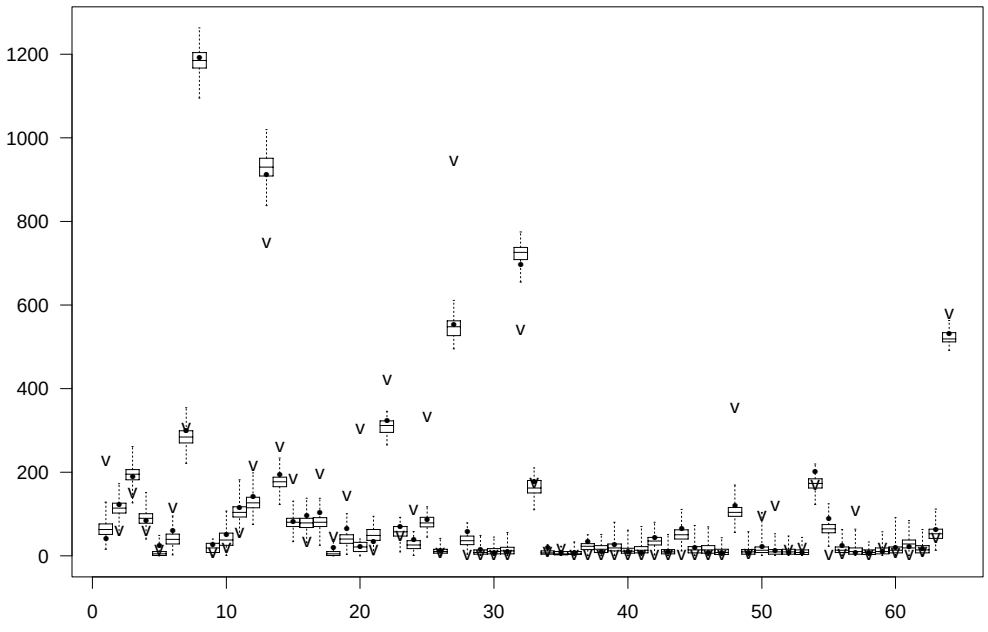


Figure 4.10: Posterior distributions for the components of $\mathbf{X}$, starting from Gamma priors on λ_i with mean and variance tailored with respect to each of the real known values of $\mathbf{X}$. The point represents the actual value used by the simulator. The "v" comes from running Vardi's algorithm.

algorithm works only in cases of repeated observations. Thirty vectors of 64 independent components were generated, each from a Poisson distribution whose parameter is the value used by the traffic simulator for that **OD** pair. Sample means and variances were computed and the updating for each value of λ_i reiterated 1000 times, using as starting points several of the original 64 dimension vectors generated. The values coming up from this solution are dramatically distant from the true values, suffering the same kind of distortion as my original algorithm does.

The good performance of the algorithm in the three cases described in the list above, and in Figure 4.8, Figure 4.9, and Figure 4.10, indicates how the only sensible way to overcome the empasse is through a fair informative prior elicitation, capable of guiding the posterior distribution in the "right direction". In these three cases the entire range of components of $\mathbf{X}$ is well estimated, even in the more liberal case represented by Figure 4.8.

This is a good point at which to draw a temporary line. It hints at possible future developments in terms of informed priors, and perhaps more complex settings which can allow for correlations between the different components of the vector λ. In any event, care must be taken when making inference about complex, large networks with diverse order of magnitude in the link counts, and/or large link counts. The estimation of the posterior distributions of the quantities of interest will in these instances suffer from a bias that can be avoided only by use of informative prior distributions.

4.7 Summary

In this chapter the algorithm is applied to data generated by a traffic flow simulator capable of approximating a real world situation. The inference for the mean traffic intensities produced by the model can be compared to the known actual intensities

used as input for the simulator. The result of a straightforward application of the algorithm as presented in the preceding chapter consists of a large number of poorly fitted values, most of which are situated dramatically in the tail of the obtained empirical posterior distribution. As a result of this outcome a number of different trials were undertaken in hopes of determining the cause of this failure to correctly estimate the $\mathbf{X}$ values.

After several different trials an explanation for the rule governing the consistently biased results is found. The poor fitting is attributed to an asymmetry in the behavior of the likelihood, which although mathematically perfectly justifiable – resulting from the Poisson distributional assumption – doesn't have any substantial justification in the problem setting where statistical independence of the **OD** intensities is assumed.

In such cases the Bayesian approach can be exploited to the fullest. Information, or prior opinions, about the single traffic intensities are usually available to guide the posterior analysis; thus it becomes possible to directly address this "biased likelihood" issue through the prior.

The choice of the prior can easily take into account more or less precise ideas about the different λ's through the choice of the parameters. The good performance of the algorithm after the implementation of more informed priors, indicates that a fair informative prior elicitation capable of guiding the posterior distribution in the "right direction"represents the only sensible means of circumventing the impasse. After this course is implemented, the entire range of components of $\mathbf{X}$ is well estimated.

Chapter 5

Seattle Highway I-5

5.1 Timed Networks

Starting from the present chapter I develop an analysis for a different kind of data, whose nature calls for a new perspective. I want first to translate briefly the previous setting in the present one, adding to that sort of analysis a new dimension, namely time. Initially I will just sketch what could be a way of tackling a set of measurements along a linear sequence of links now labeled by the variable t.

Take, for instance, a simple path from one node A to another B, passing through one or more additional nodes. The number of **OD** pairs is here up to

$$(n-1)+(n-2)+\ldots+1$$

if n is the number of nodes, because a request of "unidirectionality"is made. If the path is ABCD, there can be pairs AB,AC,AD,BC,BD,CD but no "loops" that pairs as DC,DB,DA,CB,CA would introduce. The data are now labeled with time: at time $T=t$ the number of messages that have passed through the nodes, up to that moment is recorded:

$$\underline{Z}_t = (Z_t^A, Z_t^B, Z_t^C, Z_t^D).$$

For each of the links, in this case

$$(ab, bc, cd)$$

the distribution of travel time, in terms of the cumulative distribution $F(t)$ is assumed known.

For now several simplifications are introduced:

- the travel time distribution is the same for each link, and
- is independent from one link to another (no congestion but free-flow conditions);
- counts are available for each node of the path;
- the random number of messages for each **OD** pair is still Poisson distributed.

The notation remains the same as before, so long as the same quantities of interest are involved: $\mathbf{X}$ will be the vector of **OD** traffic counts, whose intensities λ are still the focus of the analysis.

$\mathbf{X}$ is of course unknown, and needs to be inferred by the knowledge of $\underline{Z}_t$, the vector of counts at each node at time t and of $F(t)$, the cumulative distribution of travel time at time t, a value that from now on is denoted simply by ϕ, with the assumption that it is the same for all the links, and all the messages.

Introduction of the vector $\mathbf{X}$ as missing data, leads, as earlier, to

$$P(\lambda|\underline{Z}_t) = \int_{\mathcal{X}} P(\lambda|\underline{Z}_t, \mathbf{X}) P(\mathbf{X}|\underline{Z}_t) d\mathbf{X}$$

and

$$P(\mathbf{X}|\underline{Z}_t) = \int_{\Lambda} P(\mathbf{X}|\underline{Z}_t, \lambda) P(\lambda|\underline{Z}_t) d\lambda.$$

The distribution $P(\lambda|\underline{Z}_t, \mathbf{X})$ is the same as the one dealt with so far, since, given $\mathbf{X}$, λ becomes independent of $\underline{Z}_t$ as it did of $\mathbf{Y}$ in the previous settings.

The focus then is on the other conditional distribution, $P(\mathbf{X}|\underline{Z}_t)$. To gain insight into it, the idea is to further augment the data, introducing the full set of Z_{ij}, with i related to the node at which the counts have been recorded, and j related to the actual **OD** address of the messages. From now on all the quantities are to be thought related to time t, although for ease of notation the time suffix will be dropped.

The vector $\underline{Z}$ can be viewed as the result of a set of sums, in the same way as the vector $\mathbf{Y}$ was the result of a sum of components of $\mathbf{X}$ in the previous networks where time was not part of the picture. So, a matrix $\mathbf{Z}$ will be available, whose entries Z_{ij} are possibly zero when the counts at node i don't involve messages with address j, and for each line i the entries that are non-zero represent the set of constituents $\{Z_{ij}\}_{j\in J_{(i)}}$, $J_{(i)}$ meaning the set of individual **OD** messages that happen to pass through node i adding up to the number Z_i.

The simulation is going to work back and forth, first producing quantities from the distribution

$$P(\mathbf{X}|\underline{Z},\mathbf{Z})$$

then from the distribution

$$P(\mathbf{Z}|\mathbf{X},\underline{Z})$$

so that the resulting $\mathbf{X}$ can be thought of as coming from the distribution of original interest,

$$P(\mathbf{X}|\underline{Z}).$$

The double conditioning simplifies the corresponding distributions, as it is shown below in a simplified situation whose results can be extended to more general settings.

The path involves four nodes – say A,B,C,D – six **OD** pairs – $X_1 = AB$, $X_2 = BC$, $X_3 = CD$, $X_4 = ABC$, $X_5 = ABCD$, $X_6 = BCD$ – and three links – ab, bc, cd – that deliver three counts – Z_1 at node B, Z_2 at node C, and Z_3 at node D, all of them at time t.

The following three equations express $\underline{Z}$:

$$\begin{aligned} Z_1 &= Z_{11} + Z_{14} + Z_{15} \\ Z_2 &= Z_{22} + Z_{24} + Z_{25} + Z_{26} \\ Z_3 &= Z_{33} + Z_{35} + Z_{36}, \end{aligned}$$

and the matrix $\mathbf{Z}$ is here a (3×6):

$$\begin{pmatrix} Z_{11} & 0 & 0 & Z_{14} & Z_{15} & 0 \\ 0 & Z_{22} & 0 & Z_{24} & Z_{25} & Z_{26} \\ 0 & 0 & Z_{33} & 0 & Z_{35} & Z_{36} \end{pmatrix}.$$

From now on the distributions are conditioned on t and λ, although these two variables are left out of the notation. As for $P(\mathbf{Z}|\underline{Z}, \mathbf{X})$ it is convenient to split it following the natural order coming from the succession of nodes:

$$\begin{aligned} P(\mathbf{Z}|\underline{Z}, \mathbf{X}) &= P(Z_{11}, Z_{14}, Z_{15}|\underline{Z}, \mathbf{X}) \\ &\quad \times P(Z_{22}, Z_{24}, Z_{25}, Z_{26}|\underline{Z}, \mathbf{Z}_{1\cdot}, \mathbf{X}) \\ &\quad \times P(Z_{33}, Z_{35}, Z_{36}|\underline{Z}, \mathbf{Z}_{1\cdot}, \mathbf{Z}_{2\cdot}, \mathbf{X}), \end{aligned}$$

where

$$\begin{aligned} \mathbf{Z}_{1\cdot} &= \{Z_{11}, Z_{14}, Z_{15}\}, \\ \mathbf{Z}_{2\cdot} &= \{Z_{22}, Z_{24}, Z_{25}, Z_{26}\}. \end{aligned}$$

Now, the Markovian structure of the successive events (numbers of messages at successive nodes) is used by incorporating information carried by the previous counts:

$$\begin{aligned} P(\mathbf{Z}|\underline{Z}, \mathbf{X}) &= \binom{Z_1}{Z_{11}, Z_{14}, Z_{15}} X_1^{Z_{11}} X_4^{Z_{14}} X_5^{Z_{15}} (\Sigma_1)^{-Z_1} \\ &\quad \times \binom{Z_2}{Z_{22}, Z_{24}, Z_{25}, Z_{26}} X_2^{Z_{22}} Z_{14}^{Z_{24}} Z_{15}^{Z_{25}} X_6^{Z_{26}} (\Sigma_2)^{-Z_2} \end{aligned}$$

$$\times \binom{Z_3}{Z_{33}, Z_{35}, Z_{36}} X_3^{Z_{33}} Z_{25}^{Z_{35}} Z_{26}^{Z_{36}} (\Sigma_3)^{-Z_3},$$

where the quantities of the form $\binom{a}{b,c,d}$ are the Multinomial coefficients, namely:

$$\binom{a}{b, c, d} = \frac{a!}{b!c!d!}$$

and the quantities $\Sigma_i,\ i = 1, 2, 3$ stand for the sum of the X or Z quantities that appear in each equation, in order to deliver the correctly normalized values of the multinomial probabilities. That is:

$$\Sigma_1 = X_1 + X_4 + X_5,$$

$$\Sigma_2 = X_2 + Z_{14} + Z_{15} + X_6,$$

and so on for Σ_3.

Notice that the vector $\mathbf{X}$ represent a sufficient statistic for the $\mathbf{Z}$, so that the components of $\underline{\lambda}$ don't appear anywhere in the equations. Neither does ϕ, because under the assumption of a common, fixed, independent travel time distribution for each link and each message the ϕ is a common factor to both numerator and denominator and thus cancels out.

By conditioning step by step "one node upstream" it is unnecessary to convolute $F(t)$. This would not be the case in the presence of missing data; that is when the counts at some node are not available.

Now, for $P(\mathbf{X}|\mathbf{Z}, \underline{Z})$ the simplification is going to be dramatic, since

- conditioning on the fully detailed set of $\mathbf{Z}$, the components of $\mathbf{X}$ remain conditionally independent;
- for each of the **OD** traffic counts X_j only the first registered number of messages will contain relevant information: that is, only $Z_{(i)j}$ matters, where (i) indicates the minimum of the indices i for which the element Z_{ij} is non-zero.

So, by indicating with $\underline{Z}^*$ the vector of these "first counts", the distribution of interest factorizes as:

$$\begin{aligned} P(\mathbf{X}|\mathbf{Z},\underline{Z}) &= P(\mathbf{X}|\underline{Z}^*) \\ &= \prod P(X_i|Z_i^*) \\ &\propto \prod P(X_i)P(Z_i^*|X_i), \end{aligned}$$

where $P(X_i)$ is the prior Poisson distribution for the number of messages with i-th **OD** address, and

$$P(Z_i^*|X_i) = \binom{X_i}{Z_i^*} \phi^{Z_i^*}(1-\phi)^{X_i - Z_i^*}.$$

Both the assumption of a common travel time distribution for the different links and that of independence of travel times along subsequent links are relevant here.

Simulating from the Multinomial distribution $P(\mathbf{Z}|\underline{Z},\mathbf{X})$ is an easy task. Setting up a Metropolis-Hastings step for $P(\mathbf{X}|\mathbf{Z},\underline{Z})$ is also easy, using the Poisson priors as proposal and computing the ratio of the two likelihoods for the probability of accepting the candidate. The step to get the current values for the vector λ is added after each generation of the vector $\mathbf{X}$. Nothing changes in this respect, still dealing with Gamma priors and conditional posteriors.

To perform the simulation, the only input is a matrix having the same dimensions as $\mathbf{Z}$, with elements 0 or 1 if the corresponding Z_{ij} are zero or positive, the observed counts $\underline{Z}$, the value for ϕ, and a first guess for $\mathbf{X}$. The results of a simulation were checked by computing the exact posterior distributions for a simple path, with three nodes – A,B,C – two links – ab,bc – and three **OD** pairs – $X_1 = AB$, $X_2 = ABC$,$X_3 = BC$.

In this case, the counts are just two:

$$\underline{Z} = (Z_1, Z_2)$$

and the $\mathbf{Z}$ matrix looks like:

$$\begin{pmatrix} Z_{11} & Z_{12} & 0 \\ 0 & Z_{22} & Z_{23} \end{pmatrix}$$

where Z_{11} corresponds to the number of messages with address AB registered at node B, Z_{12} to those with address ABC, registered at the same node; while Z_{22} are these latter ones registered at node C, together with Z_{23} corresponding to the messages with address BC, arrived at node C by time t.

The exact distribution $P(X_1, X_2, X_3|Z_1, Z_2)$ is factorized, up to a normalizing constant as the product of three independent Poisson priors $P(X_1, X_2, X_3) = \prod_{i=1}^{3} P(X_i|\lambda_i)$ and $P(Z_1, Z_2|X_1, X_2, X_3)$. For the latter, we can write

$$P(\underline{Z}|\mathbf{X}) = P(Z_1|\mathbf{X}) \cdot P(Z_2|Z_1, \mathbf{X}).$$

Now, as a general rule, even if in this case the first factor doesn't need it, these distributions are computed by conditioning on and integrating out the knowledge of the possible values of $\mathbf{Z}$, consistent with the observed counts. In fact:

$$\begin{aligned} P(Z_1|\mathbf{X}) &= \sum_{\mathcal{Z}_{11}} P(Z_{11} = z_{11}, Z_{12} = Z_1 - z_{11}|\mathbf{X}) \\ &= \sum_{\mathcal{Z}_{11}} \binom{X_1}{z_{11}} \phi^{z_{11}} (1-\phi)^{X_1 - z_{11}} \binom{X_2}{Z_1 - z_{11}} \phi^{Z_1 - z_{11}} (1-\phi)^{X_2 - Z_1 + z_{11}} \\ &= \binom{X_1 + X_2}{Z_1} \phi^{Z_1} (1-\phi)^{X_1 + X_2 - Z_1}, \end{aligned}$$

while:

$$\begin{aligned} P(Z_2|Z_1, \mathbf{X}) &= \sum_{\mathcal{Z}_{12}} P(Z_2|Z_1, \mathbf{X}, Z_{12} = z_{12}) P(Z_{12} = z_{12}|\mathbf{X}, Z_1) \\ &= \sum_{\mathcal{Z}_{12}} \Big(\sum_{\mathcal{Z}_{22}} P(Z_2|Z_1, \mathbf{X}, Z_{12} = z_{12}, Z_{22} = z_{22}) P(Z_{22} = z_{22}|Z_1, \mathbf{X}, Z_{12} = z_{12}) \Big) \\ &\quad P(Z_{12} = z_{12}|\mathbf{X}, Z_1), \end{aligned}$$

where:

$$\begin{aligned}
P(Z_{12} = z_{12}|\mathbf{X}, Z_1) &= \binom{Z_1}{Z_{12}} (\frac{X_2}{X_1+X_2})^{Z_{12}} (\frac{X_1}{X_1+X_2})^{Z_1 - Z_{12}} \\
&= \binom{Z_1}{z_{12}} (\frac{X_2}{X_1+X_2})^{z_{12}} (\frac{X_1}{X_1+X_2})^{Z_1 - z_{12}} \\
P(Z_{22} = z_{22}|Z_1, \mathbf{X}, Z_{12} = z_{12})) &= \binom{Z_{12}}{Z_{22}}) \phi_*^{Z_{22}} (1-\phi_*)^{Z_{12}-Z_{22}} \\
&= \binom{z_{12}}{z_{22}}) \phi_*^{z_{22}} (1-\phi_*)^{z_{12}-z_{22}} \\
P(Z_2|Z_1, \mathbf{X}, Z_{12} = z_{12}, Z_{22} = z_{22}) &= P(Z_{23} = Z_2 - Z_{22}|Z_1, \mathbf{X}, Z_{12}, Z_{22}) \\
&= P(Z_{23} = z_{23}|Z_1, \mathbf{X}, z_{12}, z_{22}) \\
&= \binom{X_3}{z_{23}} \phi^{z_{23}} (1-\phi)^{X_3 - z_{23}}.
\end{aligned}$$

Note that in the previous equations a ϕ_* indicates the cumulative distribution function of travel time whose value is not in general the same as ϕ. This arises as the probability that the messages with address ABC made it by time t has to be imputed here, and so the convolution of $F(t)$ with itself (as the same F holds for all the links) needs to be computed in order to end up with the distribution of the sum of two travel times. Under the assumed independence of travel times,

$$\phi_* = F_*(t) = \int_0^t f(s) f(t-s)\, ds,$$

where f is the distribution function of travel time. Here for simplicity's sake all computations assume $\phi_* = \phi$, but strictly speaking the above integral should be computed, as all the "two-links or more" convolutions should when dealing with longer routes in general.

Notice that in the preceding series of equations, conditioning on the vector $\mathbf{X}$, the dependence of $\underline{Z}$ on λ is eliminated.

As for the ranges of the summations, a natural sequence of bounds develops as a cascade: as the unique example in this simple case shows the messages summing up to Z_{22} must have already gone through node B, and therefore must be part of Z_1 and in particular less than or equal to Z_{12}. Of course another upper bound is given, however, by the original number of messages, X_2. Thus, in its superior limit, the range has the minimum between Z_{12} and X_2. In respect to its inferior limit, the superior limit of X_3 must be taken into account, which cannot be overcome by the result of the subtraction of Z_{22} from Z_2.

The theoretical results agreed with those from the simulation. The simple example presented here does not introduce peculiarities of any sort in the structure and the computations performed for the theoretical distribution were by purpose kept under a general form. Thus, this agreement of the results from the empirical estimation procedure to the theoretical distribution can be used as a general proof of the validity of the present approach to even larger – in the present setting actually "longer" – networks.

5.2 The Data from I-5

The preceding section can be viewed as an exercise in translating from a theoretical schema developed for one-shot data into a setting which includes the dimension of time. But ultimately the new kind of measurements considered in this second part of my study were studied and modeled in an entirely different manner. In fact, the focus shifted from addressing the identification of a number of **OD** flows to modeling and predicting the intensity of an undistinguished traffic flow along several links, and its development through time during the day.

The data, part of a project of study undertaken by the National Institute of Statistical Sciences, were collected along a section of Interstate Highway 5 (I-5), north

of Seattle (just outside downtown), by a series of single-loop detectors that spanned a length of 11 miles. The detectors were operative weekdays for 4 hours in the morning (6-10am) southbound, and 6 hours in the afternoon (2-8pm) northbound, recording vehicles traveling in all the lanes except those dedicated to high occupancy vehicles ("HOV" lanes) and on- and off-ramps. Between 80 and 90 detectors were at work, at a total of 25 locations (one detector per lane, with 3 or 4 lanes monitored at each location). A set of fifty days is available, going from the end of May to the middle of August 1996. After a preliminary cleaning of the data set, only 15 locations per direction were saved, whose detectors seem to give results that are both sensible and compatible with one another.

These are single loop detectors which report counts and occupancy. Readings were taken every 20 seconds but the data in this format turned out to be too volatile and unreliable and were subsequently aggregated to 1 minute resolution and over all lanes at a location. All the detectors are presence-type detectors, i.e., sensors detecting the presence and passage of a vehicle over a short segment of roadway. When a vehicle enters the detection zone the sensor is activated and remains so until the vehicle leaves the detection zone. The absence of a vehicle over a detection zone can be thought as giving a '0' signal. The presence of a vehicle as giving a '1' signal. Each detector is scanned a fixed number of times per second, say 60. The total number of scanning intervals 'on' over a time period of T minutes is referred to as occupancy (in scans). Counts are simply the number of times over the period of observation in which the detector switches from 'off' to 'on'.

Counts per minute is the variable modeled and predicted by this study.

5.3 Purpose of the Study and Exploratory Analysis

The purpose of the present analysis is to build a model capable of explaining the development of traffic flow throughout the day at the different locations, as a function of a set of explanatory variables that includes, at least initially:

- the flow at preceding time points at upstream locations;
- the time of the day;
- the topography of the locations involved (distances between detectors, presence of on- and off-ramps between detectors);
- the day of the week;
- a.m. or p.m. time of collection.

A convenient subset of the entire data set was isolated to constitute the object of an exploratory analysis of the pattern of regularities and dependencies. The segments of highway including the HOV lane were discarded from the analysis, leaving to the unrecorded on- and off-ramps alone the special feature of introducing additional degrees of unpredictability beyond the natural uncertainty of the evolution of traffic flow. In fact a number of factors inherent to the process of traffic flow introduce uncertainty in the series of counts through time at a given location, even when the same quantity has been recorded upstream at a preceding time, and that piece of information is available. Moments of congested traffic, slowdowns, or on the contrary exceptionally light traffic with high average speed, sudden stops or oversize vehicles are just a few examples. The procedure of data collection itself introduces another important source of uncertainty because of the absence of any information about the

number of cars that leave or enter the highway along ramps located between the detectors.

The a.m. portion of data was chosen for a subset of nine days, from Monday the 17th through Thursday the 26th of June. The set of detectors saved for the analysis is composed of six locations, labeled $1, 2, ...6$ with distances within one another respectively of $1.28, 0.2, 0.58, 1.96, 0.82$ miles. Figure 5.1 of next Section 5.4 represents a sketch of the layout.

The idea is to start with a simple linear regression of counts at time t at location i on counts at time $t-1$ or earlier at location $i-1$ or farther upstream. The physical nature of the process is here taken into account, and for each detector's counts the independent variables are chosen as those that, given the distance between detectors, and an hypothetical average speed of 55 miles per hour, can contribute something to the dependent variable under exam. Here is where the variable occupancy would add perhaps some valuable piece of information, being related through counts to the actual speed of the vehicles recorded at the detector. But for the time being occupancy is not taken into account.

A set of independent univariate models, one for each detector under exam, is the short term goal. Further on, though, these cellular models need to grow into a larger, multivariate one that can account for the correlation between detectors at the same time.

The exploratory analysis moved in two directions:

1. identify, or rather confirm, the pattern of dependencies already inferred by the eye-ball assessment of mean distances travelled along the links considered (here links are to be intended as the segments between two consecutive detectors);

2. detrend the set of counts for each day at each location by estimating the parameters of a spline function, and thus a smooth curve along the daily series

is drawn, to be interpreted as the consistent regular pattern of usage of that segment of roadway throughout the day. In a way this should eliminate from the picture the unknown latent variables related to the flows from on- and off-ramps.

This latter item was actually the first to be addressed, and the findings were consistent with expectations. Different detectors showed slightly different patterns in terms of the shape of the smooth line fitting the trends, and these trends carried some diversity with respect to different days of the week. Overall, the diversities don't overwhelm the similarities, already suggesting the possibility of modeling the data by an hierarchical structure.

The number of observations for each detector each day is 228 and the exploratory analysis suggested as a good choice for the design of the spline function two interior knots, at points $t = 60$ and $t = 150$, together with the endpoints and an intercept, for a total of six parameters per spline function fitted.

Once each series of observations is detrended by subtracting the estimated smooth curve, a series of linear models were constructed and fitted to the six detectors by isolating the 6 sets of independent variables through a crude assessment of conventional indices as R^2 and normality of the residuals.

The output of this first step of analysis was a pretty clear depiction of the form of the model to be fitted to each detector (i.e. which independent variables to include in each regression). An important point was the assessment of diversities and similarities among the days of the week. As already mentioned, an hierarchical structure is suggested which would allow for the peculiarity of a particular day, without giving up the opportunity to borrow strength from the whole set of daily data for the assessment of those features that show consistency throughout the week.

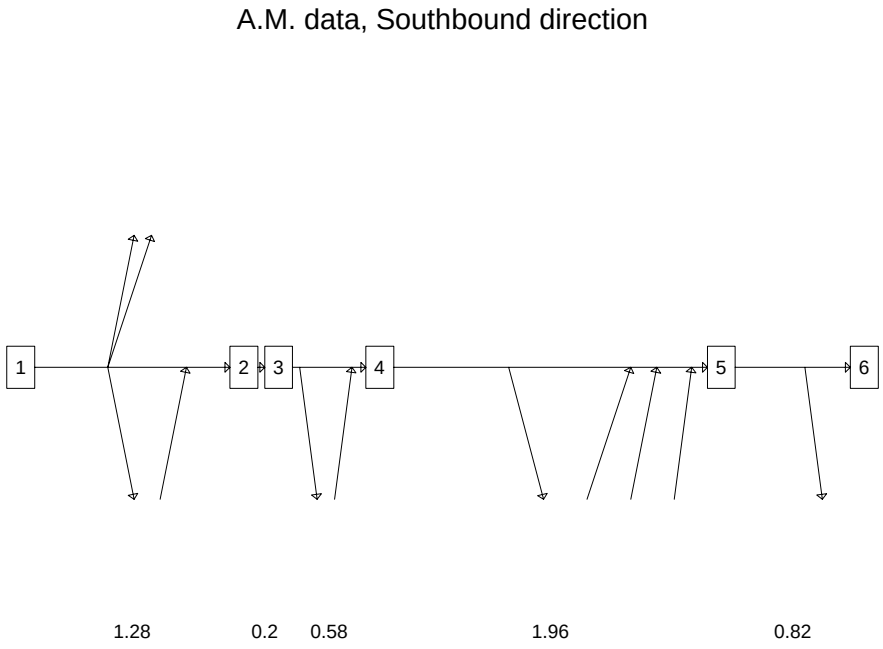


Figure 5.1: The sequence of detectors and the distances between them; the arrows joining and departing the straight line indicate on- and off-ramps

5.4 The Model

Figure 5.1 shows a sketch of the sequence of six detectors at which the counts are recorded, minute by minute, and reports the distances between each couple and the presence of on- and off-ramps. Each day has 228 minutes of recording at the six locations, so the variable "counts" indicated by Y has 3 indices to carry along:

- time of the day: $t = 1, \ldots, T = 228$;
- day of the week: $j = 1, \ldots, d = 9$, for the subset of the data used to estimate the parameter of the model;
- detector: $i = 1, \ldots, D = 6$.

The T'-dimensional vector of each detector's counts Y_{ij} is assumed to follow a Multivariate Normal distribution (MVN) whose mean is determined by the regression function and the variance-covariance matrix is taken of the form "identity times a constant", to express the belief in an uncorrelated series of errors.

The first piece of the model is therefore:

$$Y_{ij} \sim MVN(X_{ij}\theta_{ij}; \sigma^2 I).$$

The set of independent variables for each regression varies with the indices j, i apart from the portion of the design matrix that corresponds to the basis matrix for the spline function fitting. This is the part of the regression that represents the family of piecewise polynomials with the number of interior knots and degree specified in the model – the cubic spline function has been thought to fit the data well enough to justify a degree of 3 –, evaluated at the values of $t \in (1+l, T)$ where l is the maximum time lag taken into account for the explanatory variables in the regression. The basis matrices therefore each have 6 columns and numbers of rows given by the numbers of minutes in the day that the model explains, in general equal to $T' = T - l$.

The hierarchical structure of the model consists of two more stages: both the vector of parameters θ and the parameter σ^2 have a distribution a priori specified as:

$$\begin{aligned} \theta_{ij} &\sim MVN(\mu_i; \Sigma_i) \quad for\ j = 1, ...d; \\ \sigma^2 &\sim IG(\frac{\nu_0}{2}; \frac{\nu_0 \tau_0^2}{2}); \end{aligned}$$

where IG indicates the Inverse Gamma distribution. The hyperparameters μ_i and Σ_i have distributions a priori specified as:

$$\mu_i \sim MVN(\eta_i; C_i);$$

$$\Sigma_i^{-1} \sim W((\rho R)^{-1}; \rho);$$

where W indicates the Wishart distribution. The seminal paper by Lindley and Smith [21], the follow-up by Smith [27], and two other papers by Gelfand and Smith [15] and Gelfand, Hills, Racine-Poon and Smith [16] provide useful reference for the Normal data hierarchical model.

The following gives in detail the set of dependent variables for each detector's regression model:

1. Detector 1 counts represent a peculiar variable, because no counts at preceding locations are available to "explain" them. The set of X_{ij} regressors is simply the basis matrix for the cubic spline function that is to be fitted to the series of data.

2. Detector 2 counts can take advantage of the records at detector 1 at time $t-1$ and $t-2$. There is a distance of 1.28 miles between the two detectors, and so, supposing normal conditions of traffic, a proportion of cars recorded 1 and 2 minutes before at detector 1 can be expected to be recorded at detector 2. The other 6 columns of the design matrix are related to the basis matrix for the cubic spline.

3. Detector 3 counts use the same regressor information as detector 2, as the distance between the two is too short to allow for any correlation between the traffic at detector 2 one minute before and the traffic at detector 3 at present.

4. Detector 4 counts use information from all three of the upstream detectors, and in particular the counts at detector 2 and 3 at time $t-1$, and the counts at detector 1 at time $t-2$ and $t-3$.

5. Detector 5 counts use the counts recorded at detectors 4, 2 and 3 minutes before, the distance between the two detectors being almost 2 miles.

6. Detector 6 counts use the number of cars recorded at detector 5 one minute before.

Even when not explicitly stated, the design matrix X_i includes the basis matrix for the smooth spline curve that fits the daily trend at each location.

5.5 Markov Chain Monte Carlo Development

Since one goal of the Bayesian analysis is obtaining the joint posterior distribution for the set of parameters of interest, the preferred method of estimation is not through an analytical development of the formulas, but through a Markov chain simulation.

A straight Gibbs sampler for the problem at hand is available as the prior distributions are conditionally conjugate. As is usually the case using the Gibbs solution, things are significantly simplified by conditioning each component of the vector of random parameters to the remaining ones. In fact, starting from the prior specifications and assumptions listed in Section 5.4, the full conditional distributions that play a role at each step of the simulation algorithm can be easily derived. In the following development I drop the index referring to the detector that is under analysis as the dependent variable of the regression, since the forms of the distributions are common to all the six detectors, and they are conditionally independent.

Write N for the total number of observations in the data set, i.e. $N = d \cdot T'$. The distributions for the parameters of interest are then updated at each step of the iterative algorithm:

$$[\sigma^2|Y, \theta_1, \theta_2, \ldots, \theta_d, \mu, \Sigma] \equiv IG(\frac{N+\nu_0}{2}; \frac{\nu_0\tau_0^2 + \sum_{i=1}^{d}(Y_i - X_i\theta_i)'(Y_i - X_i\theta_i)}{2}),$$

$$[\Sigma^{-1}|Y, \theta_1, \theta_2, \ldots, \theta_d, \mu, \sigma^2] \equiv W([\sum_{i=1}^{d}(\theta_i - \mu)(\theta_i - \mu)' + \rho R]^{-1}; d + \rho),$$

$$[\mu|Y,\Sigma^{-1},\theta_1,\theta_2,\ldots,\theta_d,\sigma^2] \equiv MVN([d\Sigma^{-1}+C^{-1}]^{-1}\cdot[C^{-1}\eta+d\Sigma^{-1}\bar{\theta}];[d\Sigma^{-1}+C^{-1}]^{-1}),$$

$$[\theta_i|Y,\Sigma^{-1},\mu,\sigma^2] \equiv MVN([\Sigma^{-1}+\frac{X_i'X_i}{\sigma^2}]^{-1}\cdot[\Sigma^{-1}\mu+\frac{X_i'Y_i}{\sigma^2}];[\Sigma^{-1}+\frac{X_i'X_i}{\sigma^2}]^{-1}),$$

where d is the number of days in the data set, $\bar{\theta}$ is the vector whose components are the means of the corresponding components of the θ_i taken over i, and $(\nu_0,\tau_0^2,\eta,C,\rho,R)$ are hyperparameters fixed a-priori.

The Gibbs sampler cycles through the different subsets of random parameters, generating new values for each in turn by conditioning to the present values of the others. The components of θ_i are sampled together, and so are the components of μ and the entries of the matrix Σ, by generating values from the previously defined multivariate distributions.

The result of the iterative sampling, after eliminating a number of initial iterations devoted to the burn-in process, are values for the variables drawn from their full joint distribution.

5.6 Analysis of Each Detector, Common Features, Model Fitting and Predictions

The described analysis has been run for each of the six detectors separately by estimating the posterior distributions of the parameters through the 9 days' data chosen as a representative sample. The results are depicted by plots of the posterior distributions for the parameters μ and θ_i, and by time series plots of the actual data and the fitted curves.

First, in Figure 5.2, part of the results of the analysis run for Detector 2 are shown and are meant to be representative of the general behavior of the estimation process. The focus here is on how much the posterior distribution of the parameter μ changes with respect to the vague prior information. Namely the prior mean was

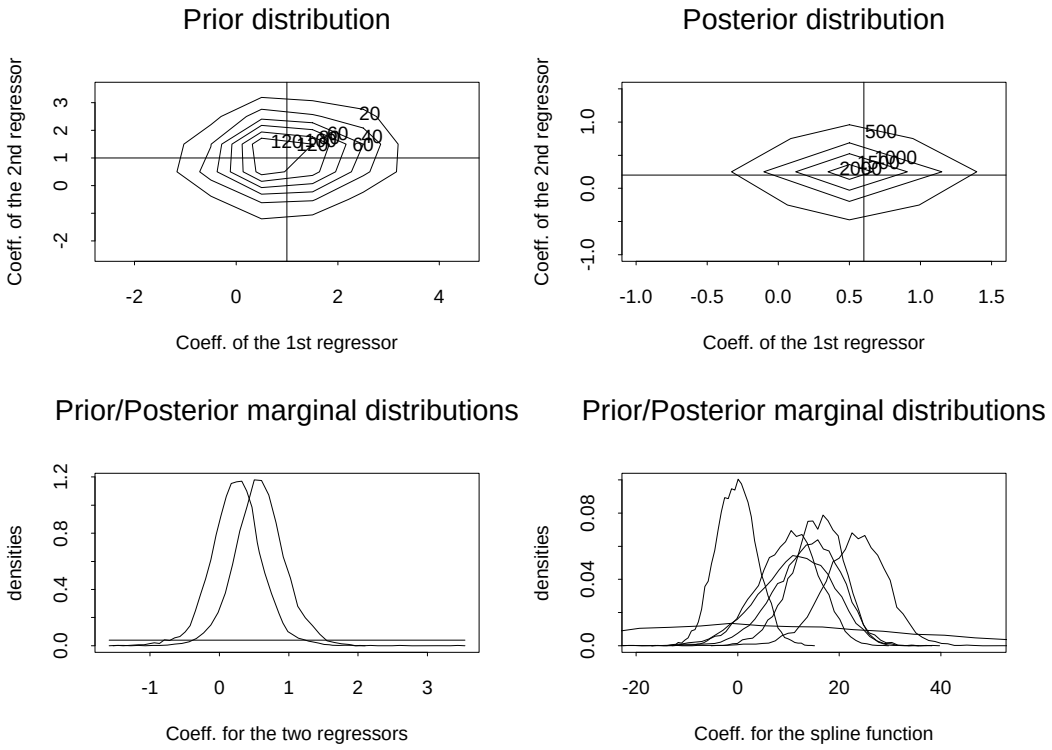


Figure 5.2: The prior and posterior distributions for the components of the vector μ of detector 2. This is meant to be just a qualitative display of the degree of change between the a-priori information and the posterior result.

taken as $\eta = (1, 1, 1, 1, 1, 1, 1, 1)$ and the prior variance/covariance matrix was taken as two independent blocks, the portion $C_\beta = 100 \cdot I_2$ related to the two coefficients of the regression on the counts at Detector 1 (at time $t-1$ and $t-2$) and the portion $C_\gamma = 1000 \cdot I_6$ related to the six coefficients of the spline function.

Then a sequence of pictures, from Figure 5.3 to Figure 5.8 display summaries of the posteriors for regression and spline parameters on each detector.

The series of pictures deserves some comments.

- For what regards the shape of the spline functions, it is easy to verify both the diversity among detectors and the similarities among days at the same detector, which is a corroboration of the assumptions of our analysis. We in

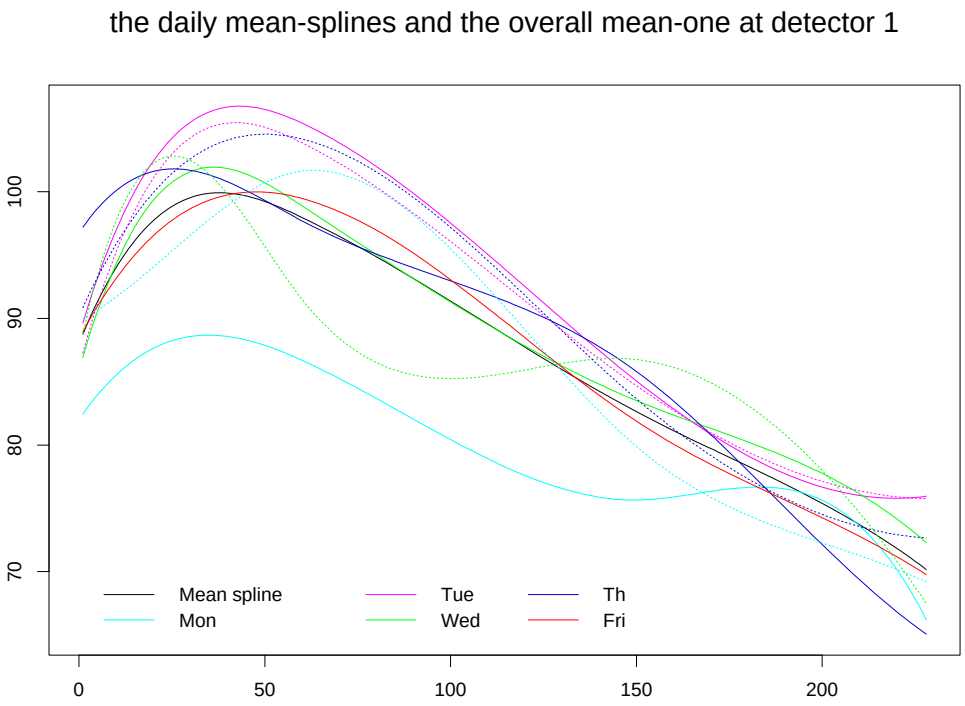


Figure 5.3: Detector 1: the posterior means of the spline functions for the 9 days and the overall mean

fact assumed, in principle, exchangeability of weeks (Mondays should present a similar behavior, as Tuesdays should, and so on). The shapes of the smooth curves are actually similar when a couple of same days of the week are compared. Furthermore, apart from the peculiar behavior of one day out of nine, each of the six detectors' set of curves show consistent patterns among the different days, hinting to a possible, more radical assumption of exchangeability of days. The different shapes among detectors are to be such: the presence of the regression over counts at preceding location should, and does account for the similarity in space, leaving to the spline function the task of explaining the peculiarities of each detector.

- For what regards the coefficients of the regression on the preceding – in time and space – counts, their significance is in almost all the instances, well stated by the bounds of the boxplots above 0. It is the case that in the presence of several regressor variables the explanatory power is "absorbed" by a single one. See in particular the plot relative to Detector 4. Again the closeness of the distributions for a specific coefficient throughout the set of days – except for the above mentioned odd day – suggests exchangeability of days, not only weeks. In these series of plots the different quality of the first variable is evident. The first variable is in fact the hyperparameter of the distribution of the remaining ones. As it is peculiar of a hierarchical model, the hyperparameter allows, by a greater dispersion, for the child parameters to account for individual features, but can be viewed as a good representation of the general behavior of the family of variables.

The real data and the series of values fitted by the models, for two in-sample days (the first Tuesday and the second Thursday of the two weeks-data set used for estimation) are shown in Figure 5.9 and Figure 5.10. The real data are plotted as

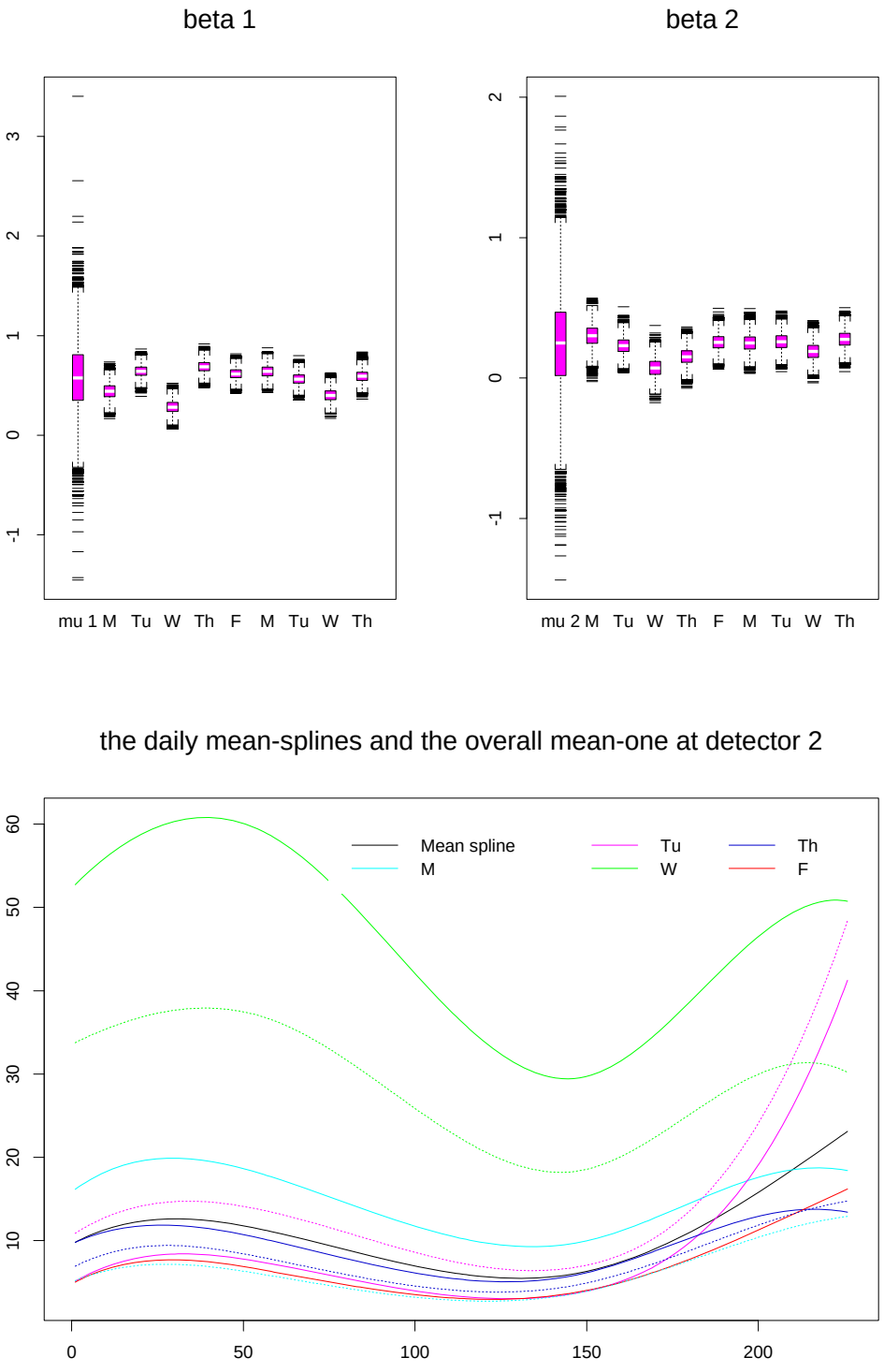


Figure 5.4: Detector 2: the distribution of the two parameters of the regression on the counts at detector 1 in the boxplots, and the posterior means of the spline functions.

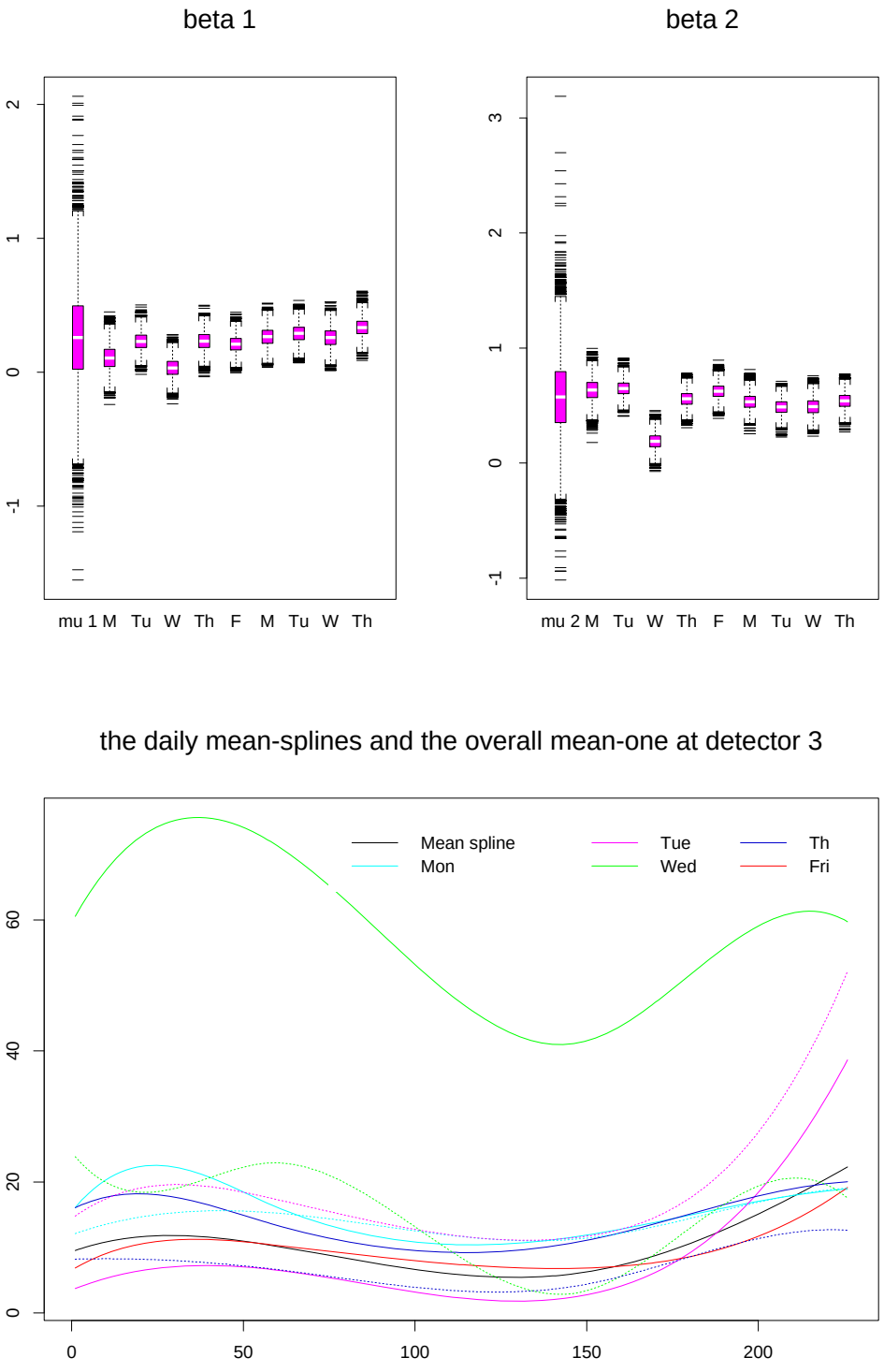


Figure 5.5: Detector 3: the distribution of the two parameters of the regression on the counts at detector 1 in the boxplots, and the posterior means of the spline functions.

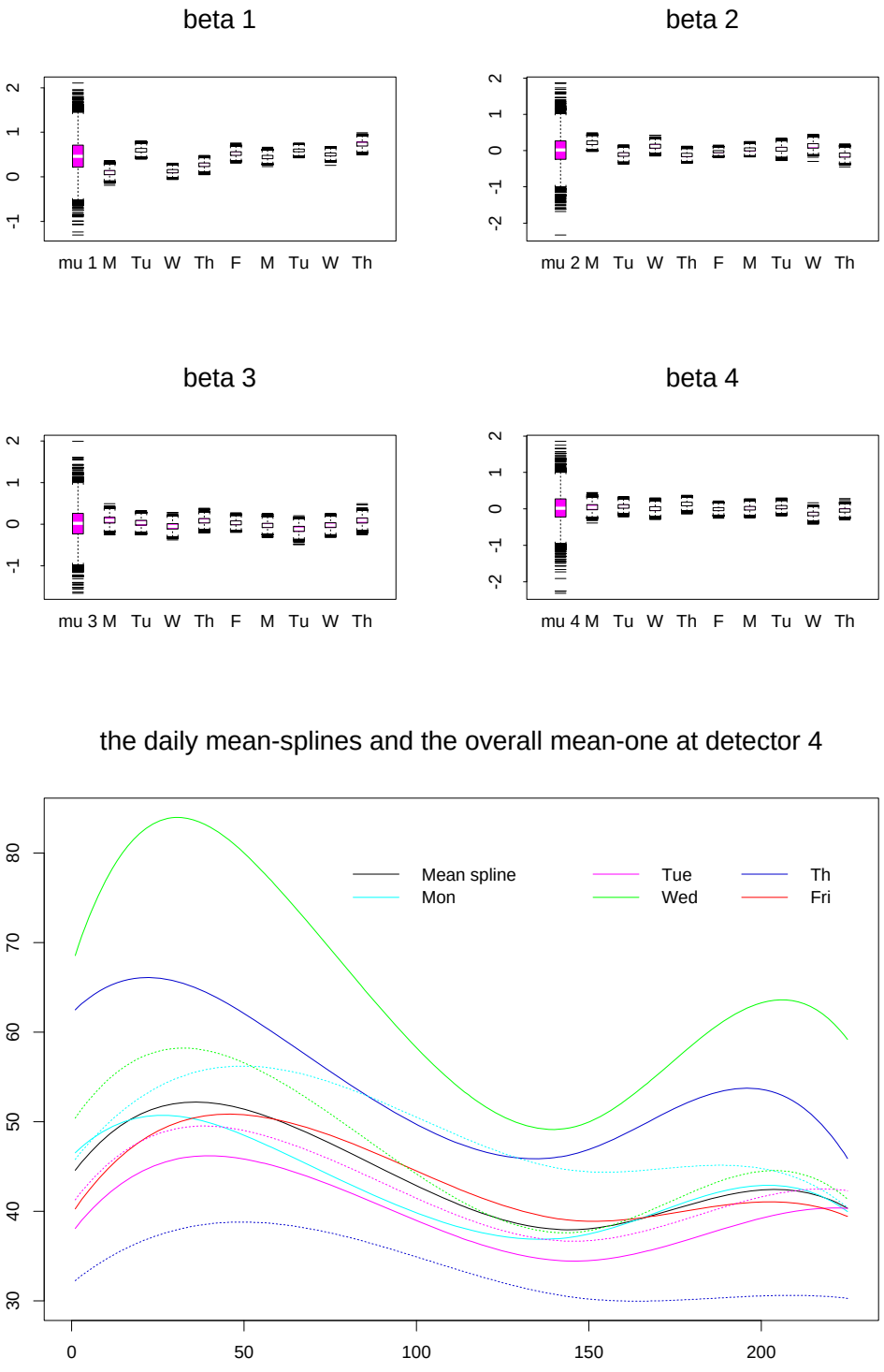


Figure 5.6: Detector 4: the distribution of the four parameters of the regression on the counts at detectors 1, 2 and 3 in the boxplots, and the posterior means of the spline functions.

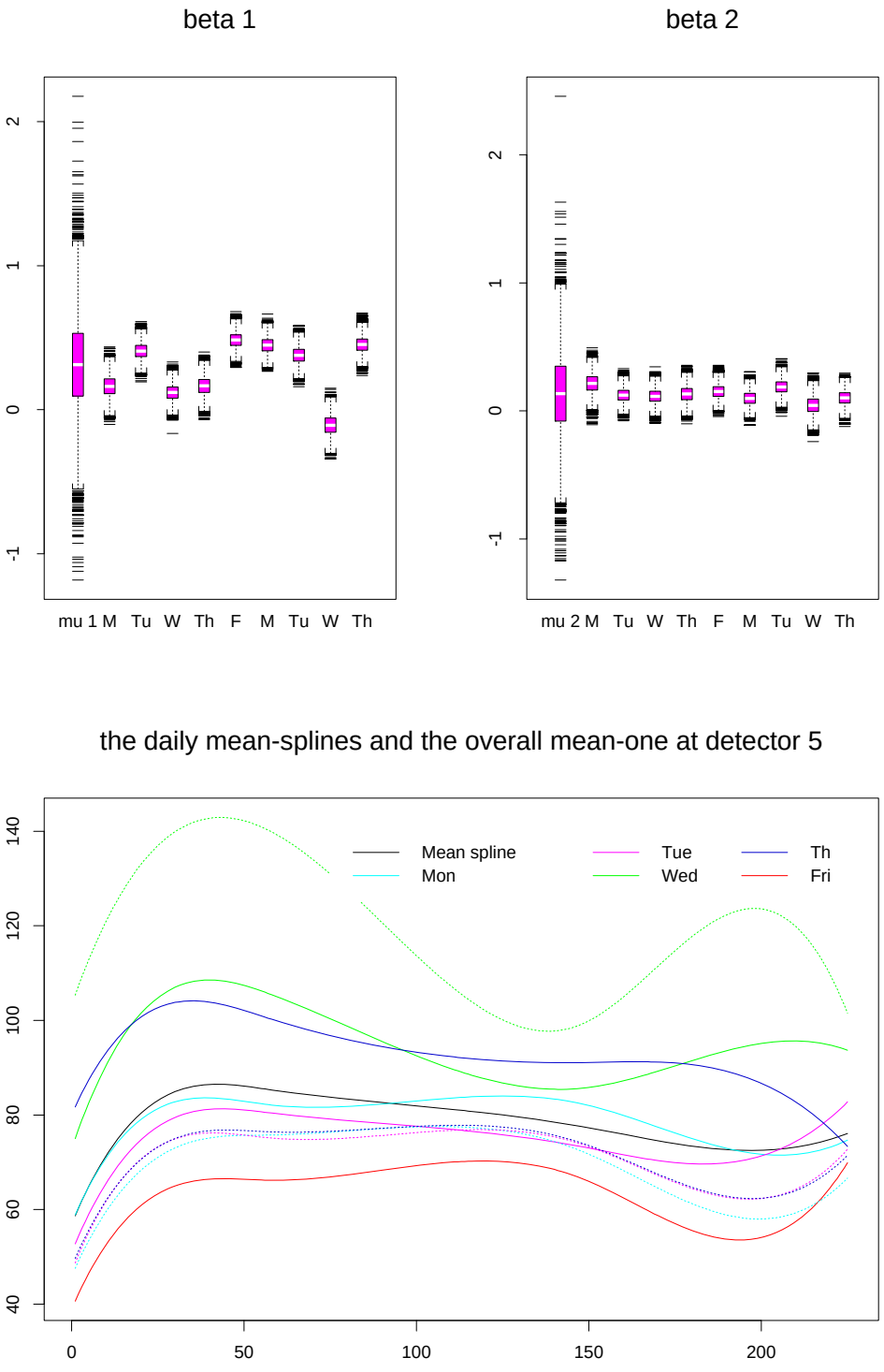
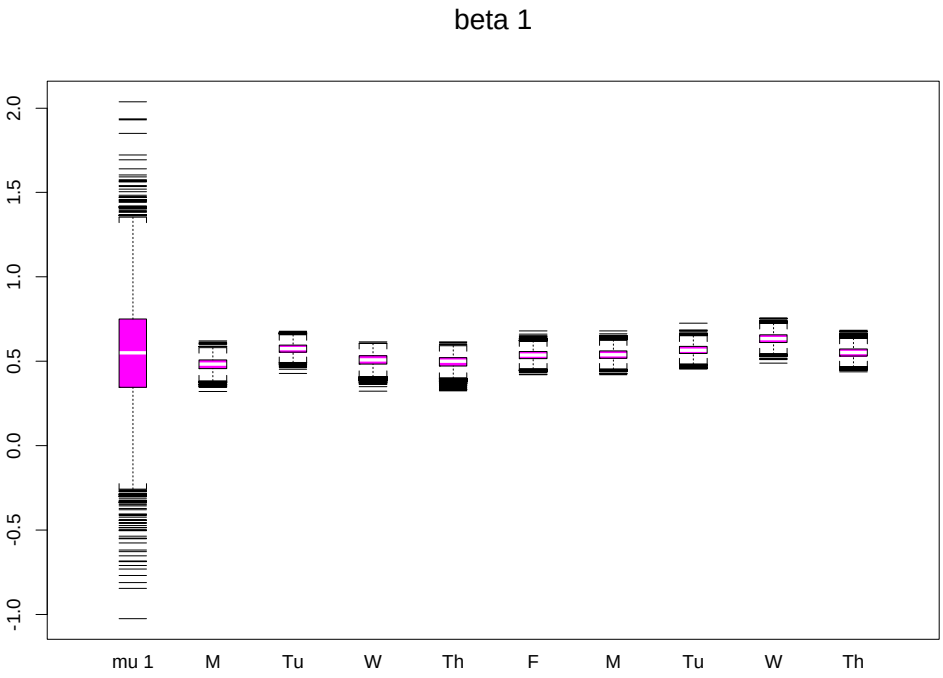


Figure 5.7: Detector 5: the distribution of the two parameters of the regression on the counts at detector 4 in the boxplots, and the posterior means of the spline functions.



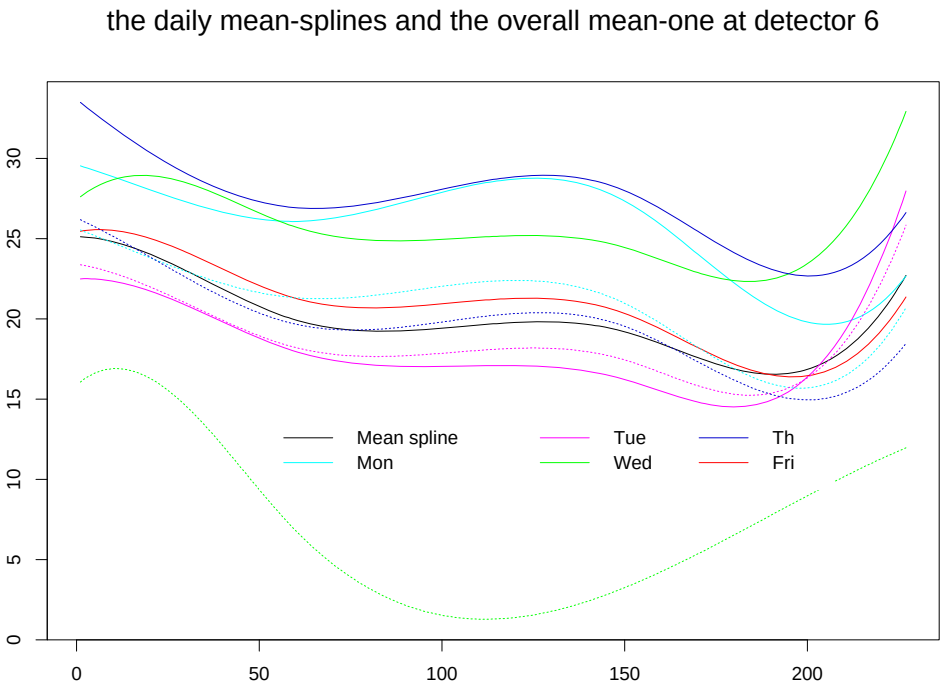


Figure 5.8: Detector 6: the distribution of the parameter of the regression on the counts at detector 5 in the boxplots, and the posterior means of the spline functions.

dotted lines, the fitted series are plotted as solid lines. It is nicely evident how the overall trend of each series is well picked up by the estimated spline function, and the regression is able to adapt to the oddities in the series when enough time and space occurs between the observations that constitute the regressor and the variable that is fitted by the model. Notice in particular the deep "hole" in the data at Detector 5 on Tuesday (Figure 5.9), that the model observes and uses to predict the correspondent dip in the series at Detector 6.

5.7 Predicting New Days

The good performance of the model for data "in sample" is encouraging, but it has still to be tested with respect to its ability to explain new, out-of-sample data. Data of the same format recorded over a different forty day period are available. The estimated parameters $\mu, \Sigma, \sigma^2, \theta_i \;\; i = 1, \ldots, 9$ can be used to model all the new sets of observations. Figure 5.11 and Figure 5.12 are plots of the data from two out-of-sample days, together with the fitted lines, in the same form of display used for in-sample data.

The explanatory power of the parameters estimated through the 9 sampled days is verified by the new data, which supports the hypothesis of the exchangeability of weeks – as the posterior means of θ_i have been used, in particular the one estimated for the first Tuesday in-sample and the ones estimated for the second Thursday in-sample. The fitting procedure would nevertheless improve from tuning the parameters after having observed a chunk of data from the detectors, and the very nature of the process induces the adoption of this perspective. In the near future cars may be equipped with computers able to process real-time data about the present conditions of the traffic along the links of interest to the driver, predict travel times, congestion episodes, intensities of vehicles on the road, and suggest the best itinerary to the

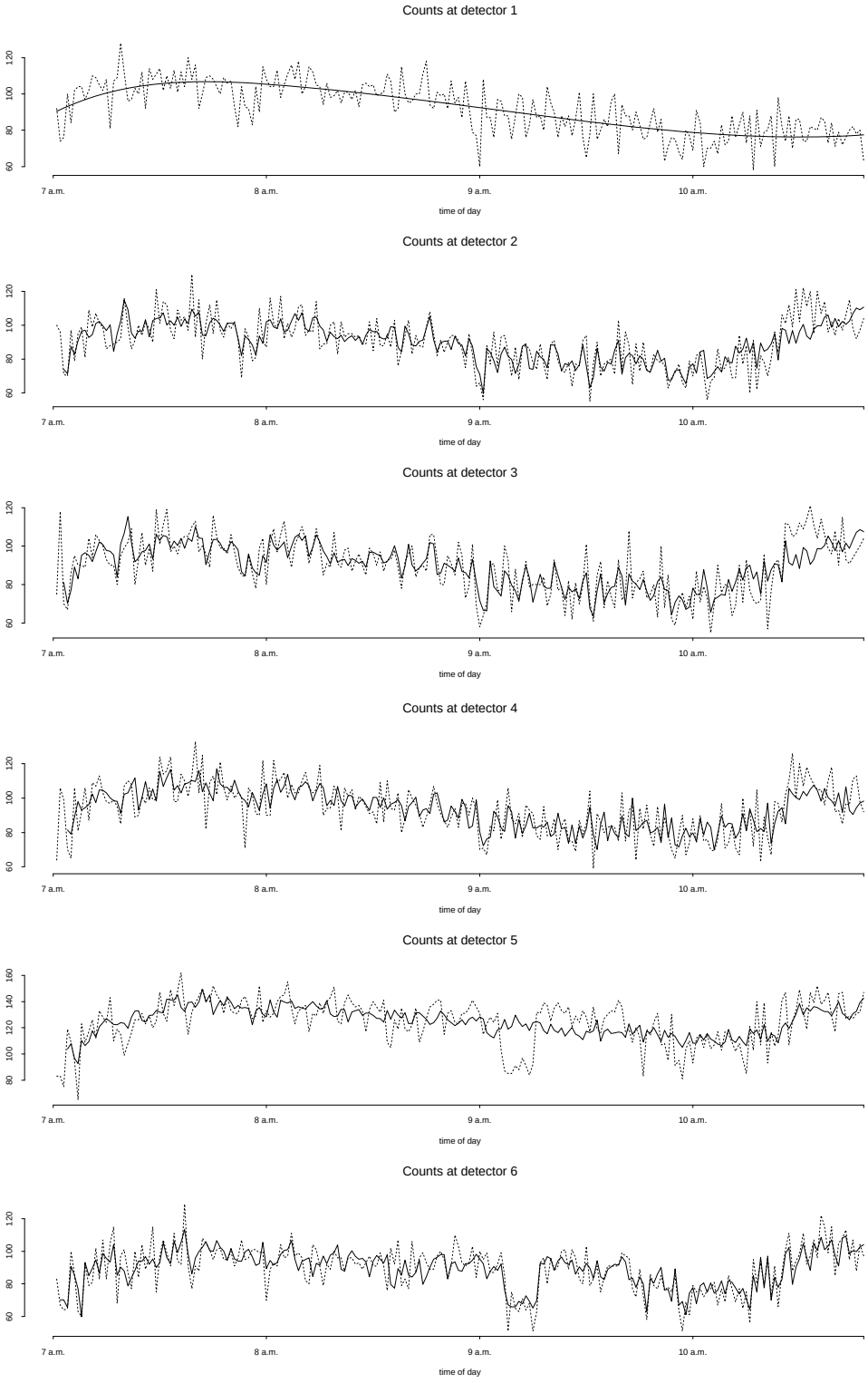


Figure 5.9: Tuesday, the 18th of June, in-sample: Real data and fitted values for the entire sequence of detectors.

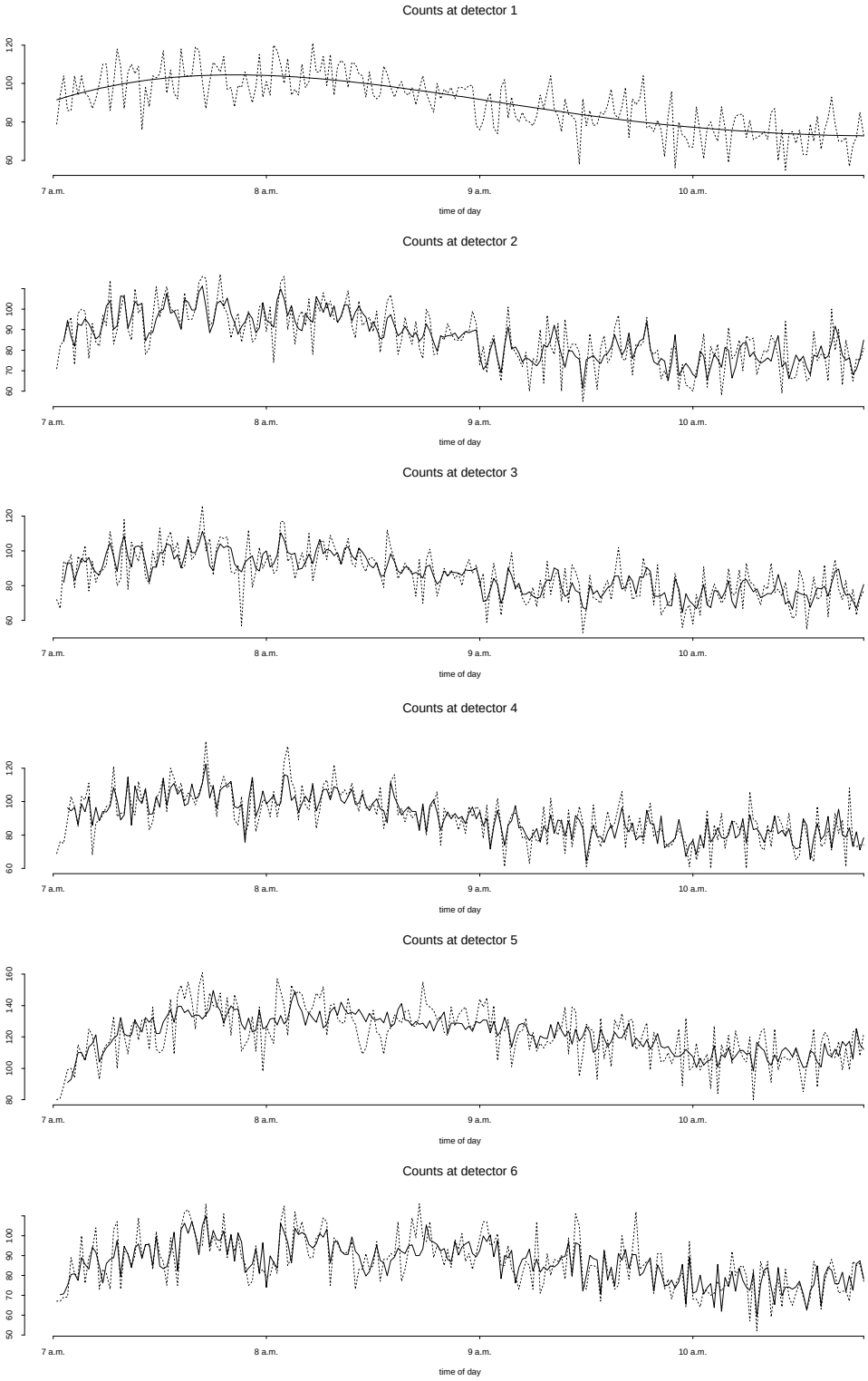


Figure 5.10: Thursday, the 27th of June, in-sample: Real data and fitted values for the entire sequence of detectors.

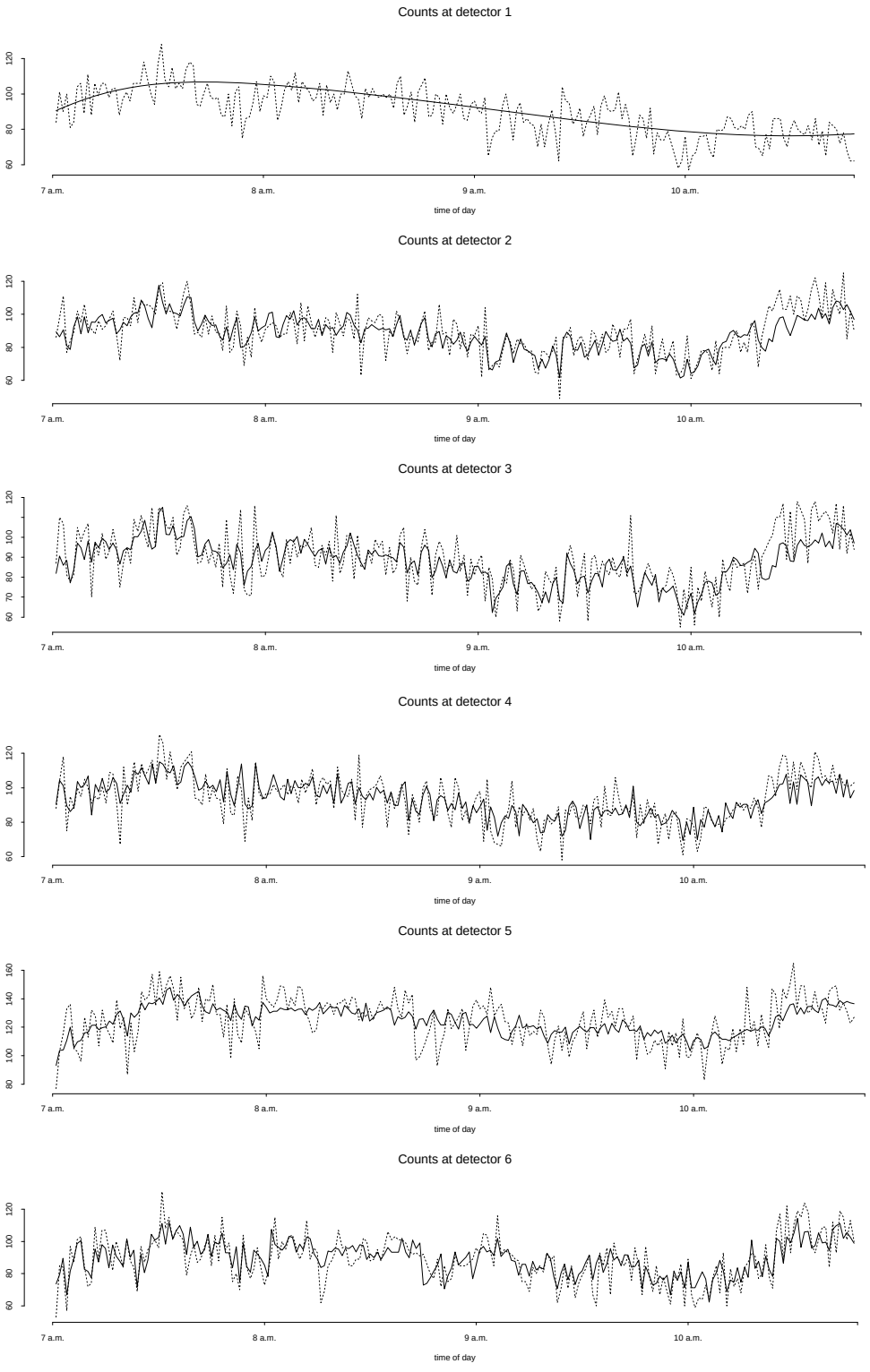


Figure 5.11: Tuesday, the 2nd of July, out-of-sample: Real data and fitted values for the entire sequence of detectors.

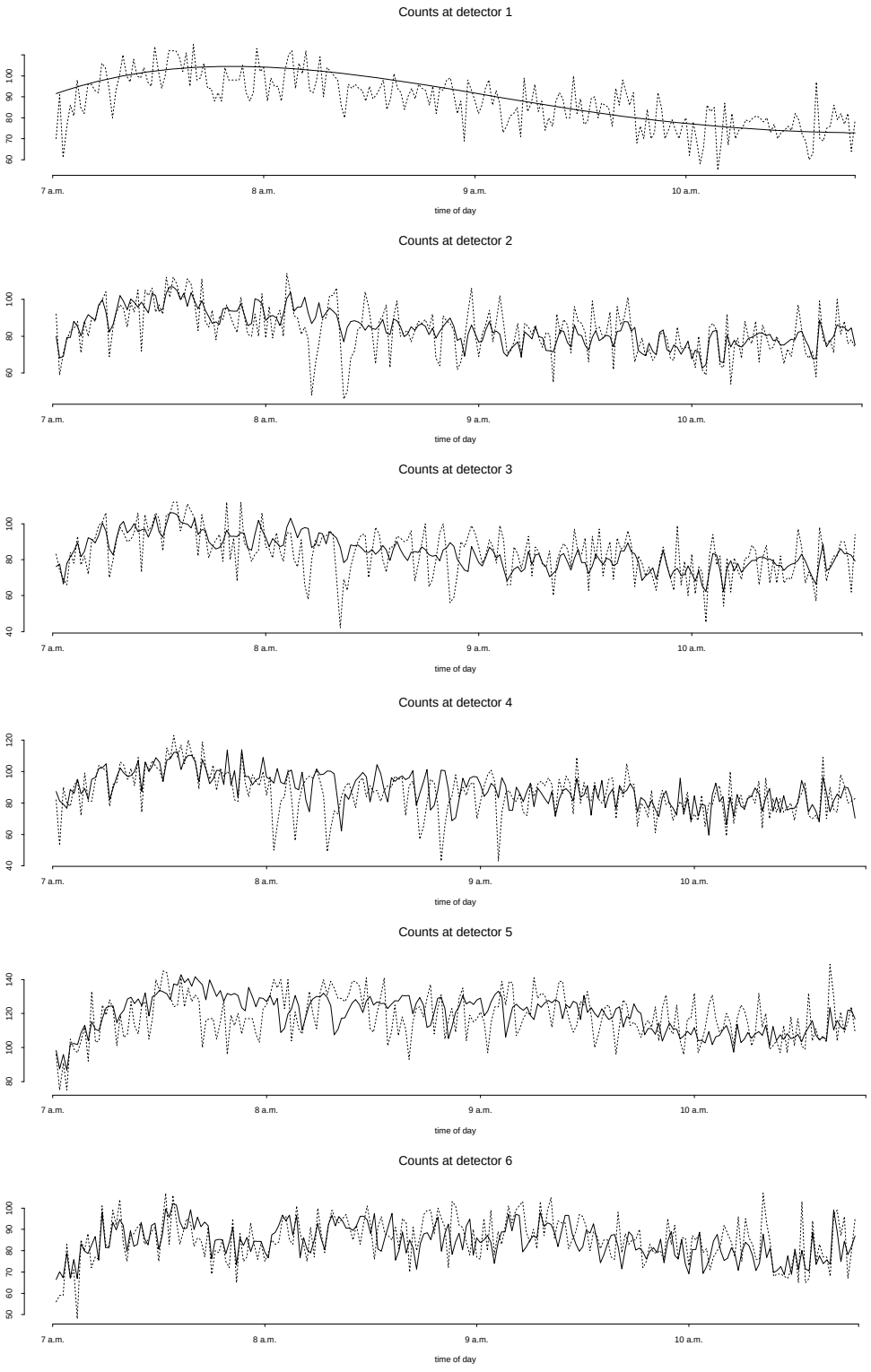


Figure 5.12: Thursday, the 8th of August, out-of-sample: Real data and fitted values for the entire sequence of detectors.

destination. The flow of data from the road detectors would be continuous, thus it is sensible to imagine the possibility of adapting the current estimations by this on-line tuning.

Thus, a dynamic update of the same parameters is of interest, to allow for revision of the estimated values of the quantities as soon as part of the new data becomes observable, and so to contribute to revised predictions about the development during the later part of the day. The approaches followed below with respect to this issue are:

- the simple computation, at any given time t, of the new posterior distribution for the parameters given the new set of information up to this time;
- the modeling of a dynamic behavior by assuming a general univariate dynamic linear model (DLM) as in West and Harrison [42].

Actually there is no need to present these two approaches separately. The reason will be clear after a brief sketch of the relevant theory.

5.8 Dynamic Linear Models or Simple Prior/Posterior Updating

The general univariate DLM is defined by the observation equation:

$$Y_t = X_t\theta_t + v_t,$$

where, assuming constant variance, $v_t \sim N(0, V)$, and X_t is a known matrix of regression variables. This is coupled with the system equations:

$$\theta_t = \theta_{t-1} + w_t,$$

where, in the scaled version, $w_t \sim N(0, W_t = VW_t^*)$; and, working in terms of the precision parameter $\phi = V^{-1}$, with the initial information:

$$\begin{aligned} \theta_0|D_0, \phi_0 &\sim N(m_0, VC_0^*), \\ \phi_0|D_0 &\sim G(\frac{n_0}{2}, \frac{n_0 S_0}{2}). \end{aligned}$$

See West and Harrison [42] for a detailed and broad treatment of these models.

In considering t as a "new day", the parameters of the last distributions are taken from the posterior mean estimates that resulted from the analysis detailed in Section 5.7:

- For m_0, the posterior mean of θ_j is used according to which day of the week the new data belong to, in this way assuming exchangeability of the weeks. If exchangeability of days were to hold, the posterior mean for μ would be used instead.

- For C_0, the posterior mean of the Σ matrix is used.

- The parameters for the Gamma distribution are chosen so that S_0 equals the posterior mean for σ^2, and n_0 states a rather strong degree of "certainty" in this prior estimate, as both the exploratory analysis and the Gibbs sampling's result reconfirm the stability of this value from day to day.

The two error sequences $\{v_t\}$ and $\{w_t\}$ are assumed to be independent, and mutually independent.

Defining as D_0 the value of the parameters $(m_0, C_0^*, n_0, S_0, \{W_0^*, W_1^*, \ldots, W_T^*\})$, for this DLM closed to external information, the information at time t is recursively defined: $D_t = \{Y_t, D_{t-1}\}$.

Standard results (again West and Harrison [42]) give the formulas for one-step forecast and posterior distributions:

- at time $t-1$ the process is defined for some mean m_{t-1} and variance matrix C_{t-1} to have the posterior distribution:

$$(\theta_{t-1}|D_{t-1}) \sim MVT_{n_{t-1}}(m_{t-1}, C_{t-1}),$$

$$(\phi|D_{t-1}) \sim G(\frac{n_{t-1}}{2}, \frac{n_{t-1}S_{t-1}}{2});$$

- at time t the prior is:

$$(\theta_t|D_{t-1}) \sim MVT_{n_{t-1}}(a_t, R_t),$$

with $a_t = m_{t-1}$ and $R_t = C_{t-1} + W_t$;

- the forecast distribution for one-step ahead values is :

$$(Y_t|D_{t-1}) \sim T_{n_{t-1}}(f_t, Q_t),$$

with $f_t = X_t a_t$ and $Q_t = X_t' R_t X_t + S_{t-1}$;

- the posterior distribution at time t for the process is:

$$(\theta_t|D_t) \sim MVT_{n_t}(m_t, C_t),$$

$$(\phi|D_t) \sim G(\frac{n_t}{2}, \frac{n_t S_t}{2});$$

with $m_t = a_t + A_t e_t$, $C_t = \frac{S_t}{S_{t-1}}(R_t - A_t Q_t A_t')$, $n_t = n_{t-1} + 1$, $S_t = S_{t-1} + \frac{S_{t-1}}{n_t}(\frac{e_t^2}{Q_t} - 1)$, where $A_t = R_t X_t Q_t^{-1}$ and $e_t = Y_t - f_t$.

The use of a discount factor is chosen as the best way to define the sequence of variance/covariance matrices for the error terms in the system evolution: the matrix W_t is defined as

$$W_t = C_{t-1}(1-\delta)/\delta$$

which implies

$$R_t = C_{t-1}/\delta.$$

The value of this factor has to be tuned in the process of deciding the degree of "dynamicity" of the state vector. Always $0 < \delta \leq 1$ holds; the smaller the parameter, the larger the degree of adaptation to the behavior of the last data seen. This of course is a critical point. The adaptation doesn't have to be too quick in front of a highly volatile series, which is often the case for this kind of data. Because the estimated value of σ^2 is around 100 units, for all six detectors, it would be dangerous to allow for a quick adaptation of the parameters to each sudden jump in the series of observations; in fact even an eye-ball inspection of a few sample series shows how "sudden jumps" and intervals of higher volatility are soon wiped off by a sequence of observations more in agreement with the expected trend.

Furthermore, the possibility of block-discounting ([42]) may play an important role in this case, since the vector of regression parameters actually consists, in all but the first detector's model, of two subvectors: one containing the coefficients of the regression over the counts at the preceding detectors, at earlier time points, and one containing the coefficients for the spline function. It is sensible to think that the latter need not to adapt to aberrancies of a particular day, while the first can "learn" from the peculiar character of the oscillations that something is interfering with the "general pattern of dependencies" and can improve by a tuning action.

The limiting case in the degree of adaptation is $\delta = 1$. In this case the dynamic linear model reduces to the constant one, stating that

$$\theta_{t+1} = \theta_t.$$

Now it is clear why there is no need to consider the two approaches radically different, but rather as two instances of a single model. An algorithm for the dynamic update

is set up, and, in one of the analyses, the discount factor is set to 1, to obtain the simple prior/posterior updating in the presence of new data.

The one-step-ahead predictive distribution has already been described in terms of distribution for the vector of parameters in the regression and for the actual value of the series. When predicting k steps ahead, with $k \geq 2$, covariances between the values of the series of observations enter the picture. By easy computations, the predictive distribution is:

$$Y_{t+k}|D_t \sim T_{n_t}(f_t(k), Q_t(k)),$$

with mean and variance recursively defined by the following set of equations:

$$\begin{aligned} f_t(k) &= x'_{t+k} a_t(k), \\ Q_t(k) &= x'_{t+k} R_t(k) x_{t+k} + S_{t+k}, \\ a_t(k) &= a_t(k-1), \\ R_t(k) &= R_t(k-1) + W_{t+k}, \\ a_t(0) &= m_t, \\ R_t(0) &= C_t. \end{aligned}$$

In our model:

$$\begin{aligned} a_t(k) &= m_t, \\ f_t(k) &= x'_{t+k} m_t, \\ R_t(k) &= C_t + \sum_{i=0}^{k-1} W_{t+k-i}, \end{aligned}$$

and, since we are using the discount factor technique,

$$W_t = \frac{1-\delta}{\delta} C_{t-1},$$

so the expression for $R_t(k)$ becomes:

$$R_t(k) = C_t(1 + k\frac{1-\delta}{\delta}).$$

When considering more than one value to predict, the distribution becomes a multivariate T:

$$\mathbf{Y}_t^k \sim MVT_{n_t}(f_t^k; \mathbf{Q}_t),$$

where the notation means

$$\begin{aligned} \mathbf{Y}_t^k &= (Y_{t+1}, Y_{t+2}, \ldots, Y_{t+k})', \\ f_t^k &= (x_{t+1}'m_t, x_{t+2}'m_t, \ldots, x_{t+k}'m_t)', \end{aligned}$$

and

$$\mathbf{Q}_t = \begin{pmatrix} x_{t+1}'R_t(1)x_{t+1} & x_{t+1}'R_t(1)x_{t+2} & \ldots & x_{t+1}'R_t(1)x_{t+k} \\ x_{t+2}'R_t(1)x_{t+1} & x_{t+2}'R_t(2)x_{t+2} & \ldots & x_{t+2}'R_t(2)x_{t+k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{t+k}'R_t(1)x_{t+1} & x_{t+k}'R_t(2)x_{t+2} & \ldots & x_{t+k}'R_t(k)x_{t+k} \end{pmatrix},$$

with $R_t(k)$ defined as above.

A simple implementation of the sequential updating as it is described above give updated posterior estimates for the parameters that govern the series behavior, up to time t. Then from time $t+1$, using the updated posterior estimates for the component of the parameters in the regression vector, forecast values can be generated from the distribution written above, and confidence intervals for these future values drawn, by iterating the generation of future "scenaries" and computing quantiles from the set of values generated.

5.9 Detector 1: Expected Spline Functions

The first analysis takes one of the above mentioned "new days", Thursday the 8th of August, and performs the updating of the parameters for the spline function that fits its daily trend, at 6 different points in time throughout the day:

$$t = 50, 90, 120, 150, 180, 200,$$

and with three different values for the discount factor:

$$\delta = 1, 0.99, 0.95.$$

Figure 5.13 shows the results. For $\delta = 1$ and $\delta = .99$ the results are indistinguishable. The quick and effective adaptation takes place immediately at $t = 50$ and after the second update at $t = 90$ the learning process can be considered over, since subsequent tuning does not change the shape and position of the spline function. The smaller discount factor of $\delta = .95$ clearly introduces too much variability in the updated versions of the curve. It exaggerates the adaptation to the local behavior of the observations by giving too much weight to the last observations to the left of the point where the updating takes place. These results were corroborated by analyses of a number of new days. The lesson seems to be to avoid applying a discount factor to the variance-covariance matrix of the vector of the spline function parameters.

5.10 Other Detectors: Block Discount for the Parameter Vector

It needs to be checked if the other portion of the parameter vector, whose components are related to the regression on the counts at preceding detectors, could benefit from assuming a dynamic nature. To answer this question, the same kind of analysis must

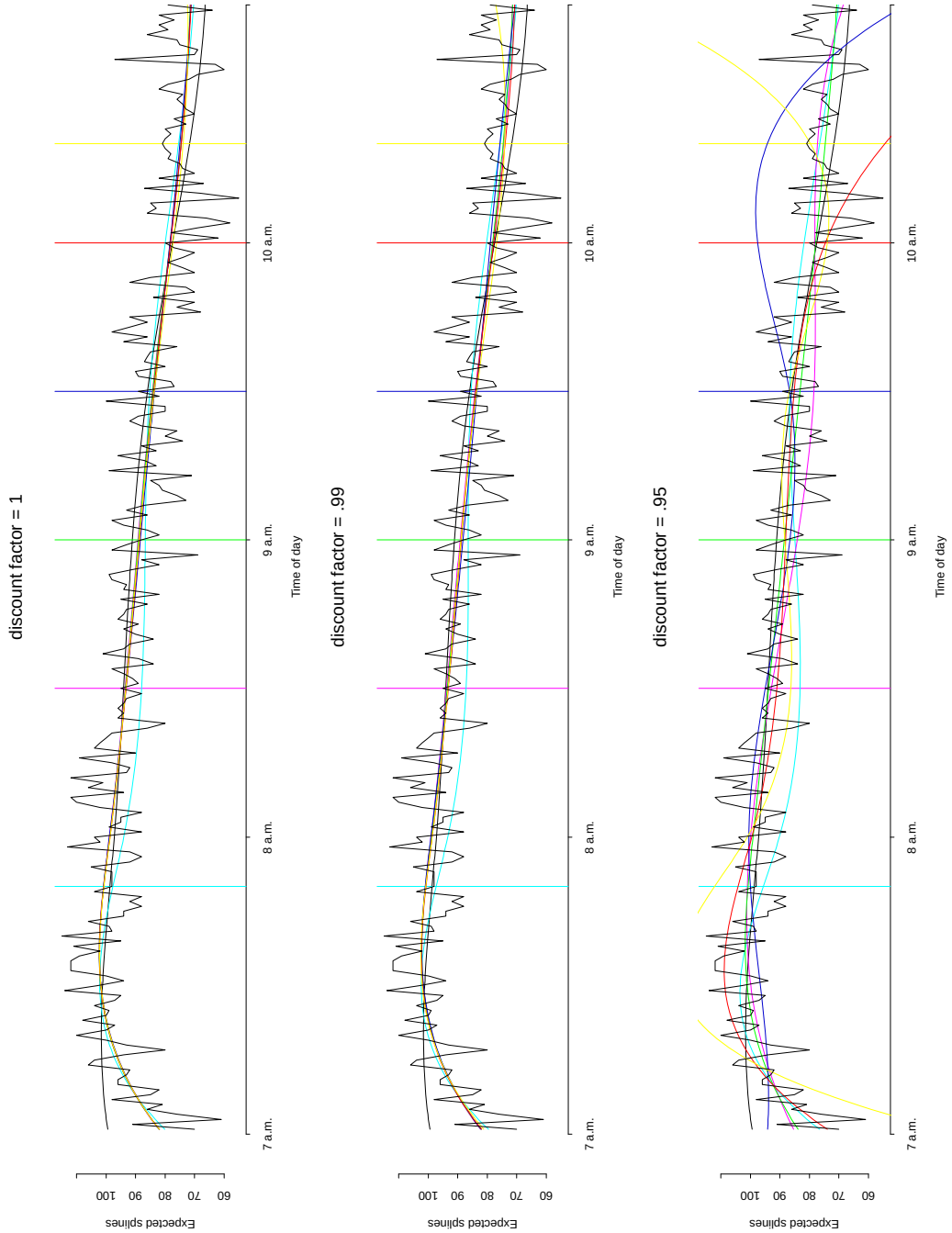


Figure 5.13: Thursday, the 8th of August; detector 1: Real data (dotted lines) and fitted spline functions, from the posterior mean of the parameter θ obtained by the Gibbs sampling analysis (black solid line) and from a number of updated posterior estimates of θ. The different colors of the curves correspond to the different time points where updating takes place, as highlighted by the vertical lines. In the three different plots, three different values of the factor δ are at work, as indicated in the titles.

be applied to detectors other than the first. For these detectors, the vector θ can be thought of as the result of the juxtaposition of two subvectors:

$$\theta' = (\beta', \gamma')$$

where the first component corresponds to the regression of Y_i $i = 2, \ldots, 6$ on the counts at detectors $i-1, i-2, \ldots$ along the specifications of the individual models. It is already clear that the γ part of the vector doesn't need discounting in the updating process, but still the other portion of the vector needs to be free of assuming a true dynamic behavior.

This issue is easily addressed by the feature of component (or block) discounting ([42]). By thinking of the evolutions of the two portions of the vector θ as independent, the matrix W_t can be designed as block diagonal. Indicate by k the number of components of β and l the number of components of γ. Call δ_β the discount factor for the β portion and δ_γ the discount factor for the γ portion. The evolution of the R_t matrix in the updating process can be seen as $R_t = C_t + W_t$ from which it is clear that the covariances between the β and γ parts of θ don't change, while the variance/covariance diagonal blocks of C_t are updated by the correspondent blocks in W_t. Translating this in terms of discount factors is easy, as the following equations show:

$$\begin{aligned} R_t &= \begin{pmatrix} C_t^\beta & C_t^{\beta\gamma} \\ C_t^{\beta\gamma'} & C_t^\gamma \end{pmatrix} + \begin{pmatrix} W_t^\beta & O \\ O' & W_t^\gamma \end{pmatrix} \\ R_t &= \begin{pmatrix} \frac{C_t^\beta}{\delta_\beta} & C_t^{\beta\gamma} \\ C_t^{\beta\gamma'} & \frac{C_t^\gamma}{\delta_\gamma} \end{pmatrix}, \end{aligned}$$

where C_t^β and W_t^β are submatrices of $(k \times k)$ dimensions, C_t^γ and W_t^γ are submatrices $(l \times l)$, $C_t^{\beta\gamma}$ is $(k \times l)$, $C_t^{\beta\gamma'}$ is its transpose, and the O, O' are to be considered matrices with all 0 components, of the corresponding dimensions.

Once this form of the model has been implemented and the analysis conducted for a number of different out-of-sample days, by performing the updating with a range of δ_β between .90 and 1 the results do not seem to differ from the ones relative to the $\underline{\gamma}$ portion of the vector. No improvement seems to be gained by allowing the parameters to take a dynamic nature, at least by an eye-ball inspection. Figure 5.14 shows how the results of different updating through the day with different discount factors do not seem to improve the fitted values beyond what is achieved by choosing $\delta_\beta = 1$.

But a rehearsal of the analysis in which the one-step ahead forecast values and residuals are computed assuming the value $\delta_\beta = 1$ first and $\delta_\beta = .95$, does render them in greater detail. Plots of the one-step ahead forecast values and real data are shown for the two days. Quantile-quantile plots that compare the distribution of the residuals with a Normal distribution, and plots of the autocorrelation functions are shown as well, to verify that nothing is left to be explained in the residuals. Only the plots related to the analysis with $\delta_\beta = 1$ are shown, nothing qualitatively different appeared in the analysis run assuming $\delta_\beta = .95$.

A further analysis of the residuals was conducted by standardizing them and computing the 95% confidence intervals; the log-likelihood functions at the entire set of time points along the daily series were summed and the results compared. Figure 5.21 and Figure 5.22 show the time series of the standardized residuals and the confidence intervals for Tuesday the 2nd of July, and Figure 5.23 together with Figure 5.24 do the same for Thursday the 8th of August. The tables compare the values of the log-likelihood function for the two days, in the six detectors, for the two values of δ_β.

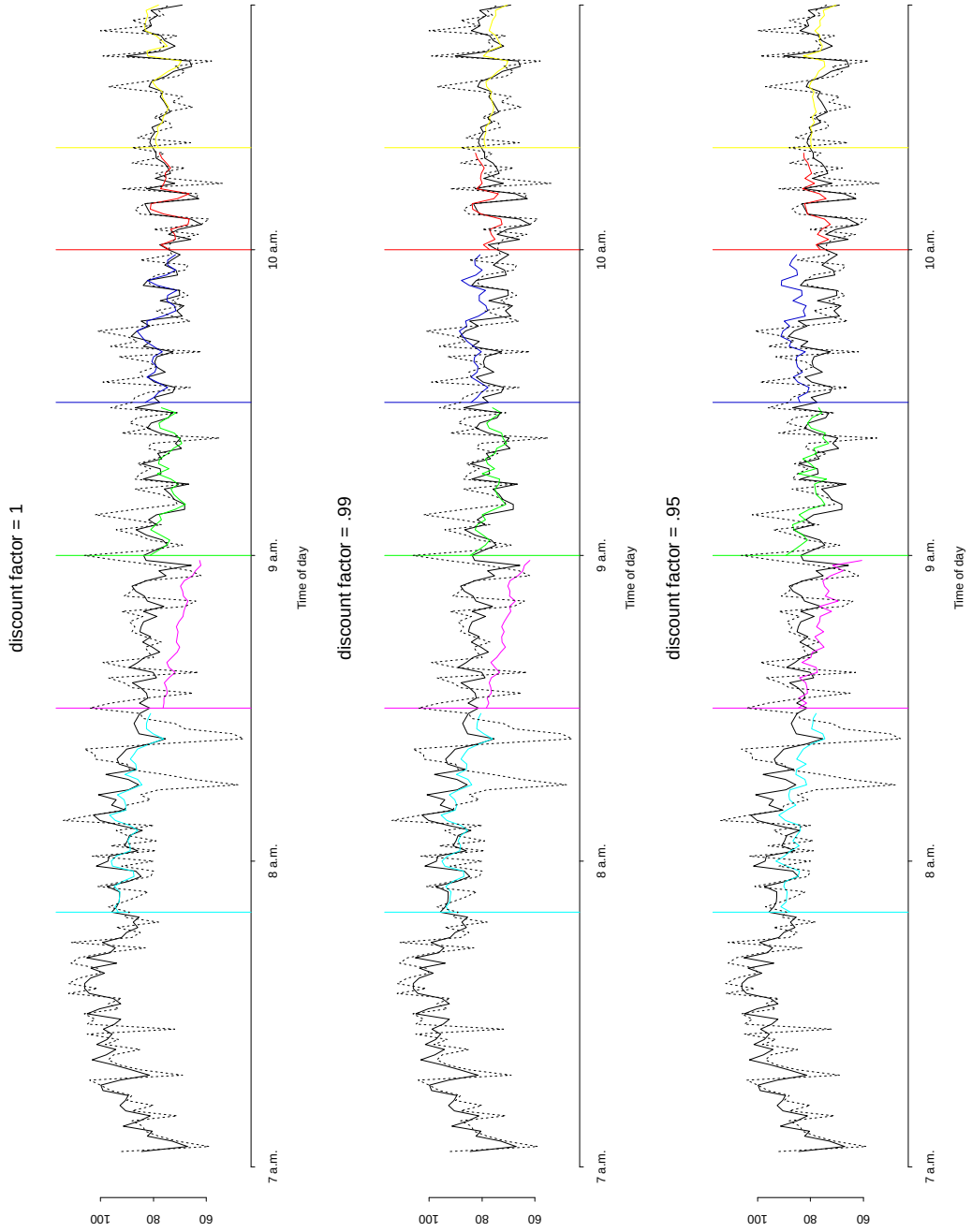


Figure 5.14: Thursday, the 8th of August; detector 2: Real data (dotted lines) and fitted values, from the posterior mean of the θ parameter obtained by the Gibbs sampling analysis (black solid line) and from a number of updated posterior estimates of θ. The different colors of the curves correspond to the different time points where updating takes place, as highlighted by the vertical lines. In the three different plots, three different values of the factor δ_β are at work, as indicated in the titles.

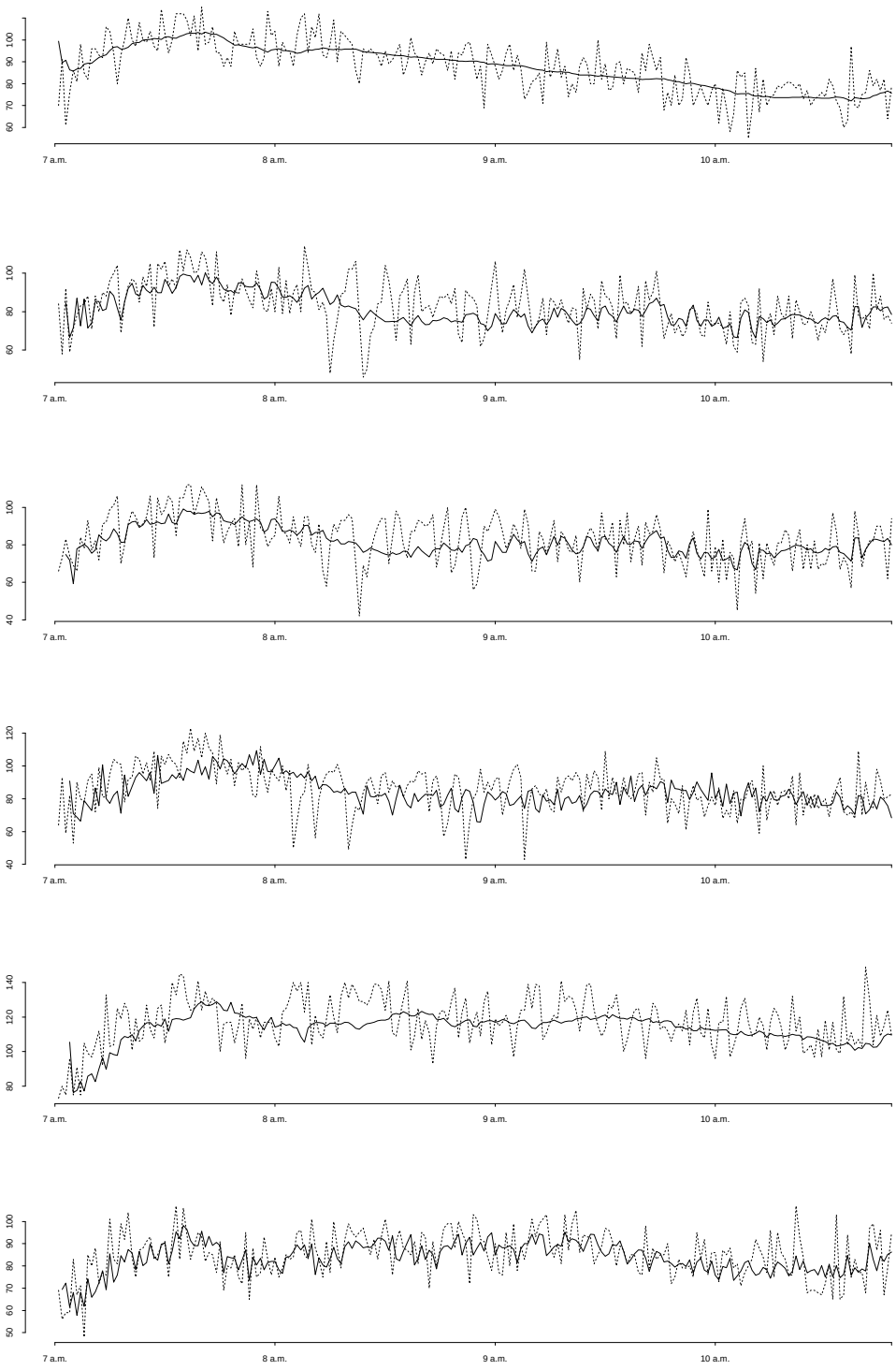


Figure 5.15: Thursday, the 8th of August. Real data (dotted lines), and one-step-ahead forecast values for the 6 detectors.

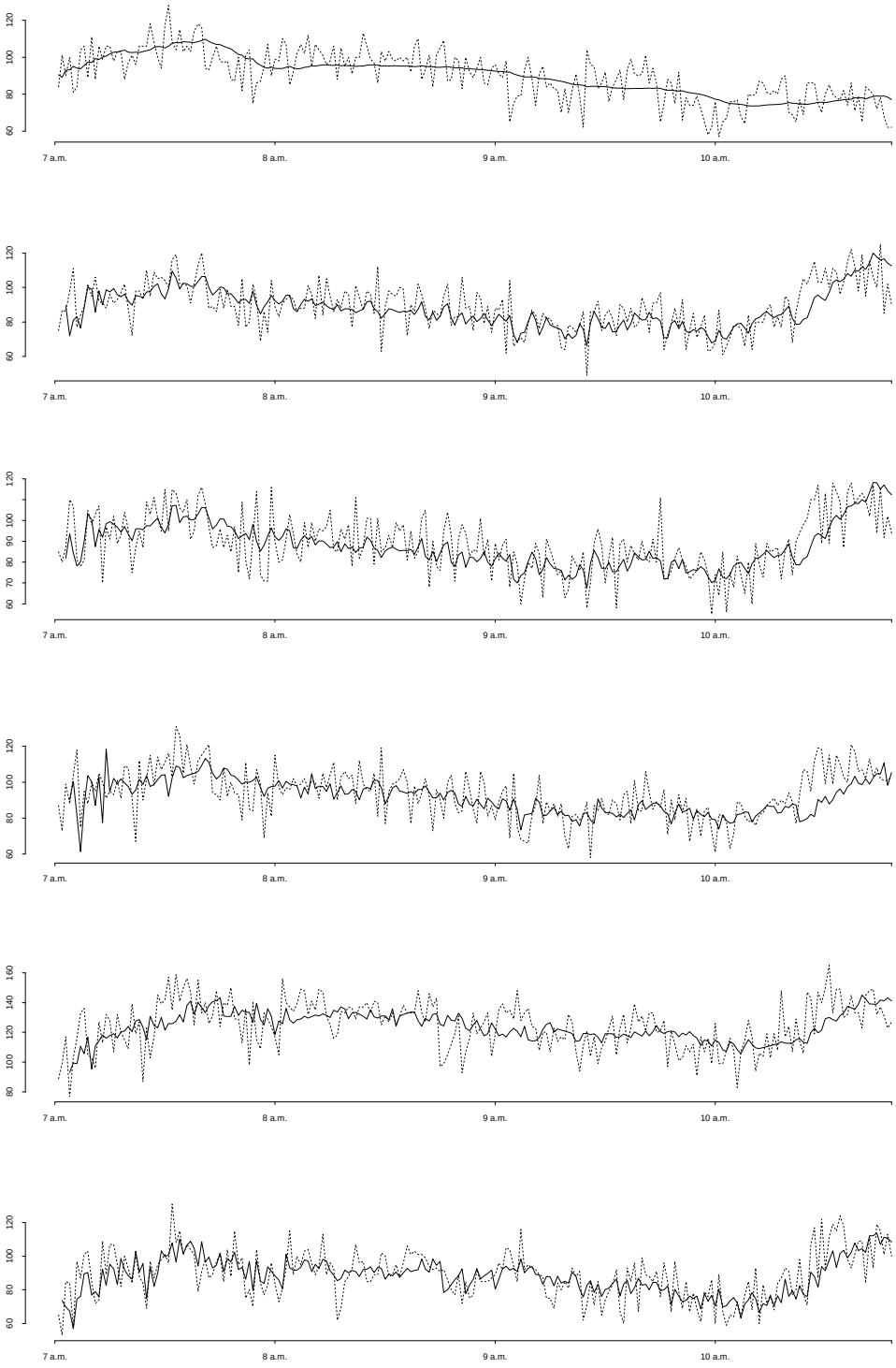


Figure 5.16: Tuesday, the 2nd of July. Real data (dotted lines), and one-step-ahead forecast values for the 6 detectors.

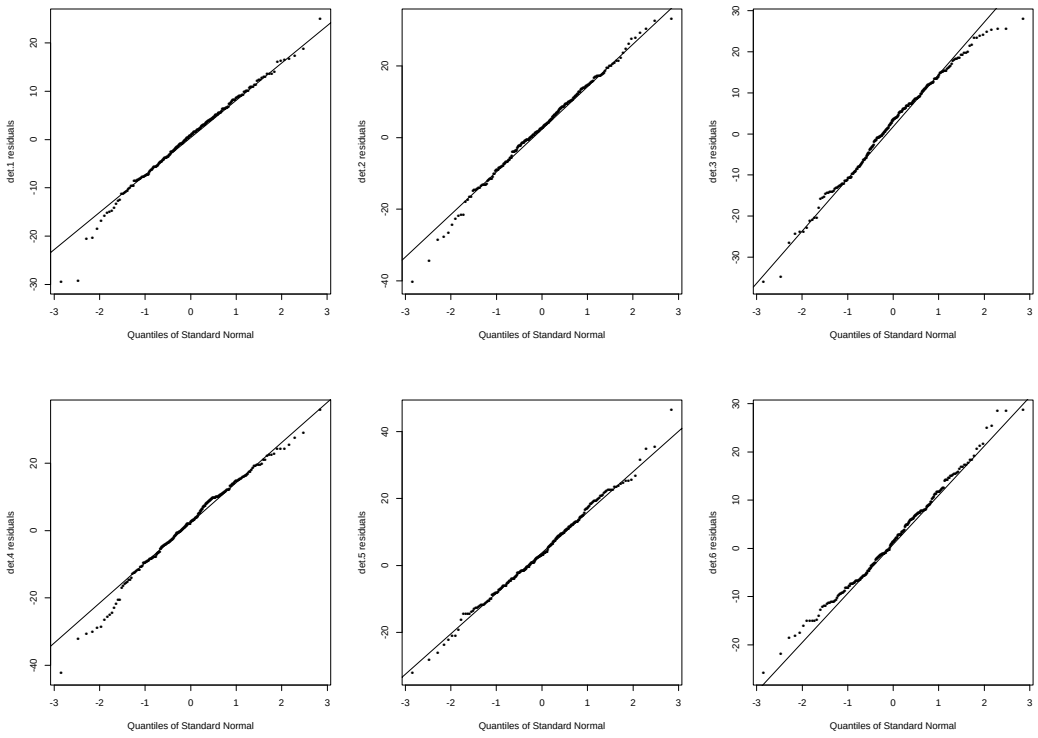


Figure 5.17: Thursday, the 8th of August. Quantile-quantile plots of the residuals at the six detectors.

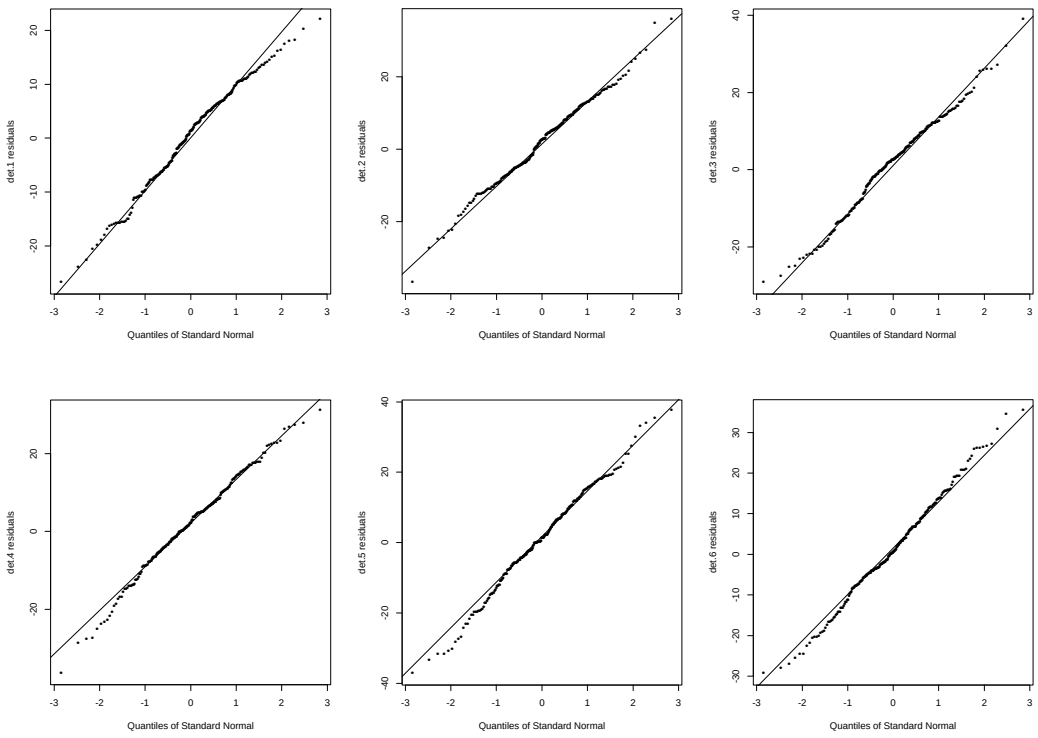


Figure 5.18: Tuesday, the 2nd of July. Quantile-quantile plots of the residuals at the six detectors.

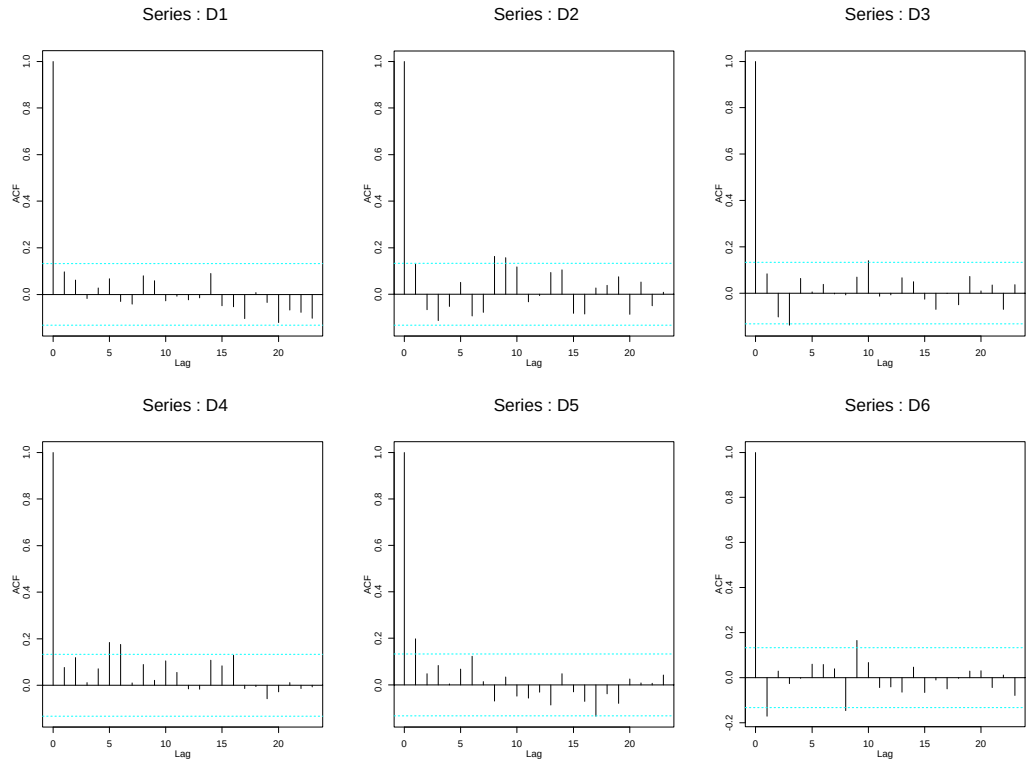


Figure 5.19: Thursday, the 8th of August. Autocorrelation functions of the series of residuals.

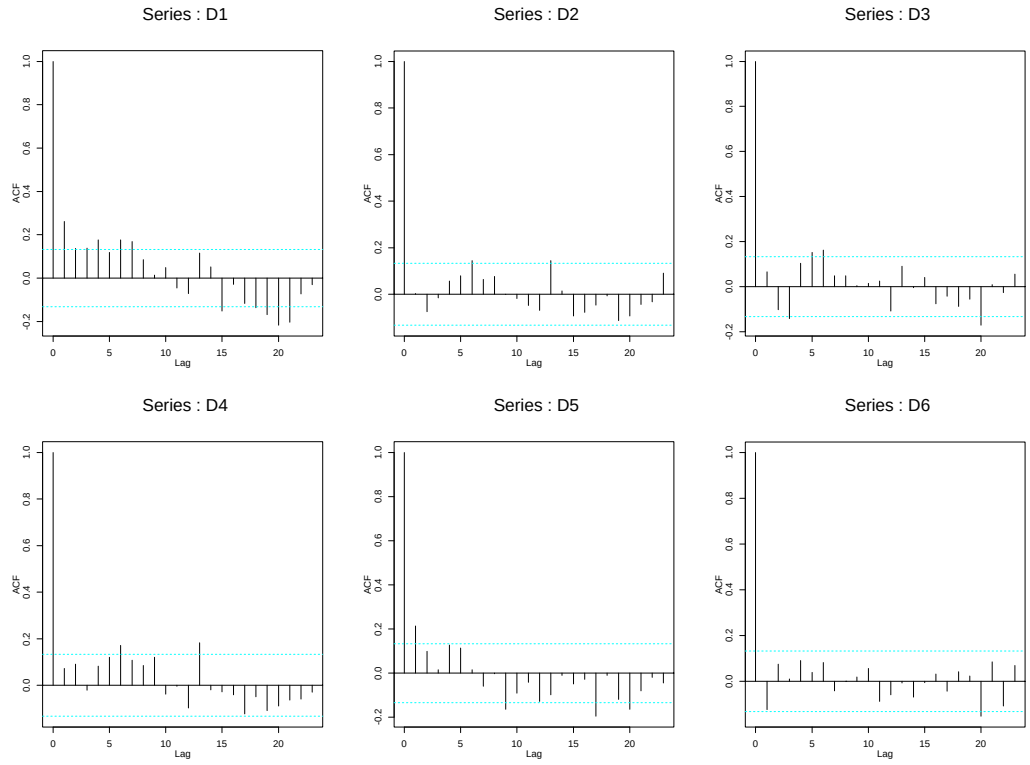


Figure 5.20: Tuesday, the 2nd of July. Autocorrelation functions of the series of residuals.

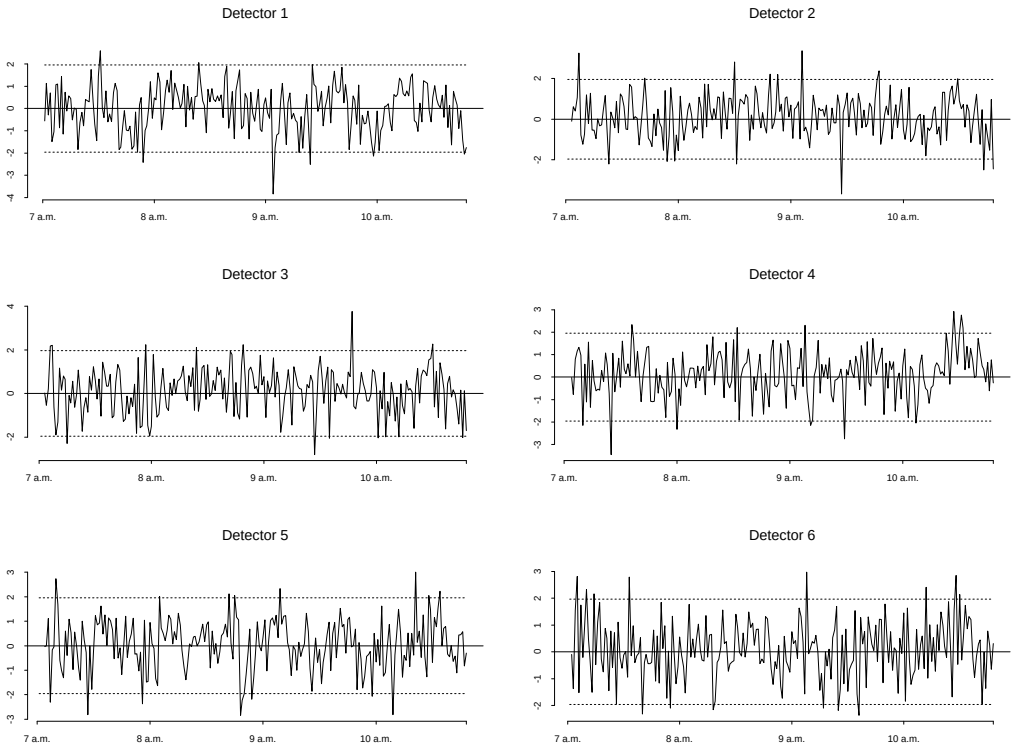


Figure 5.21: Tuesday, the 2nd of July; all the detectors: Standardized residuals from the one-step-ahead forecast values; the dotted lines show the 95% confidence intervals. In this plot $\delta_\beta = 1$.

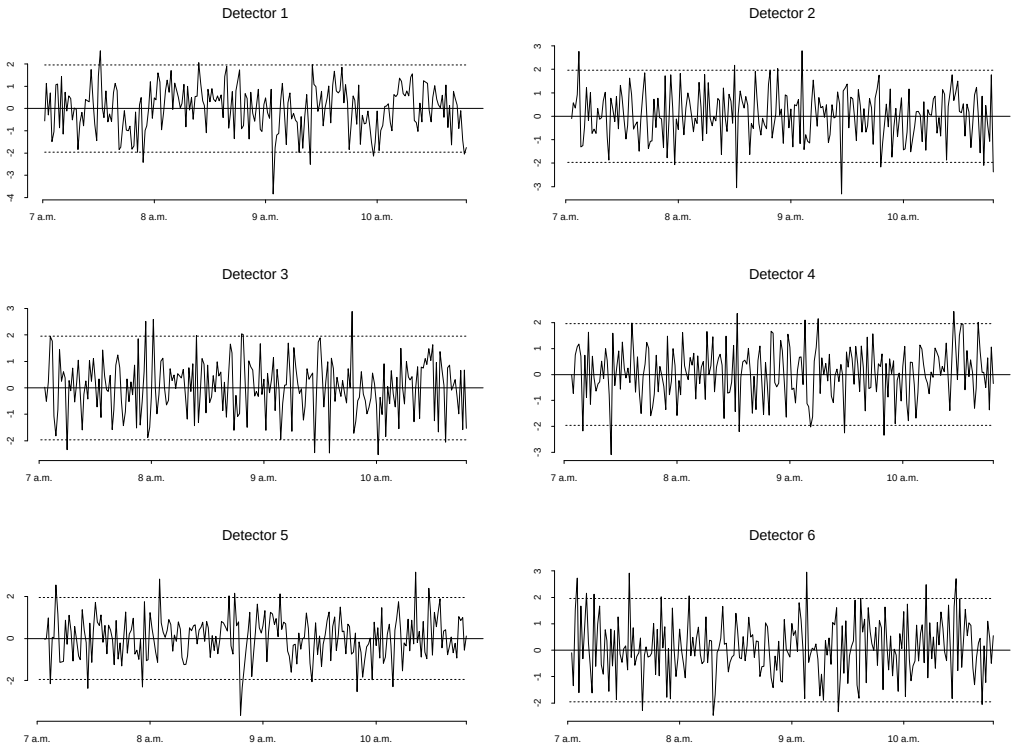


Figure 5.22: Tuesday, the 2nd of July; all the detectors: Standardized residuals from the one-step-ahead forecast values; the dotted lines show the 95% confidence intervals. In this plot $\delta_\beta = .95$.

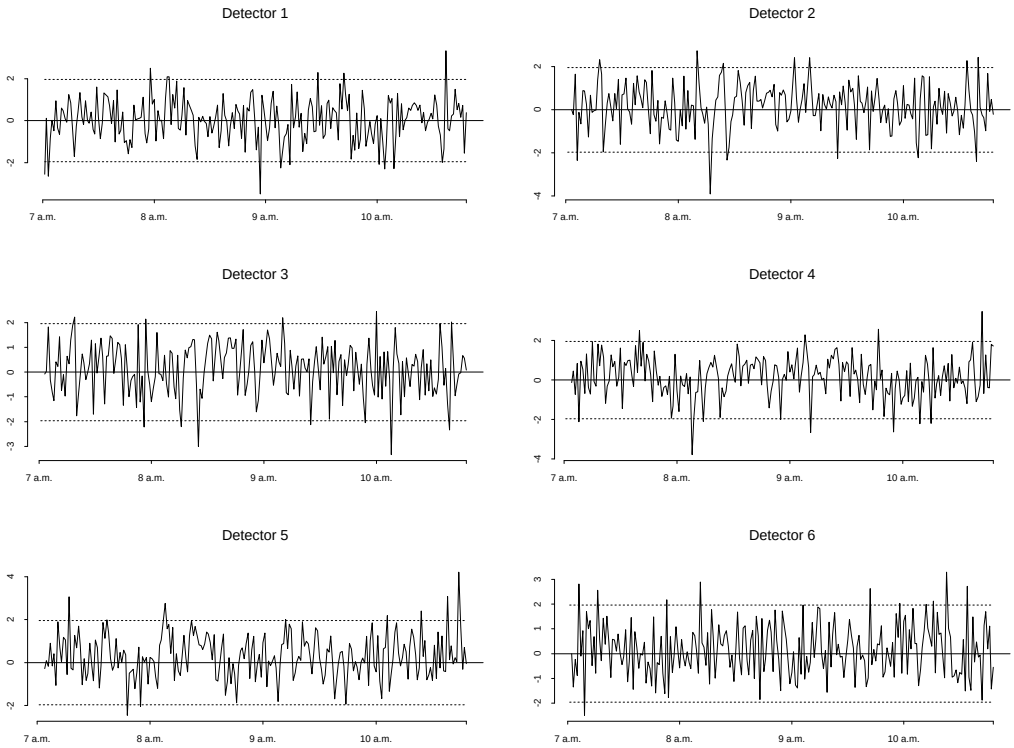


Figure 5.23: Thursday, the 8th of August; all the detectors: Standardized residuals from the one-step-ahead forecast values; the dotted lines show the 95% confidence intervals. In this plot $\delta_\beta = 1$.

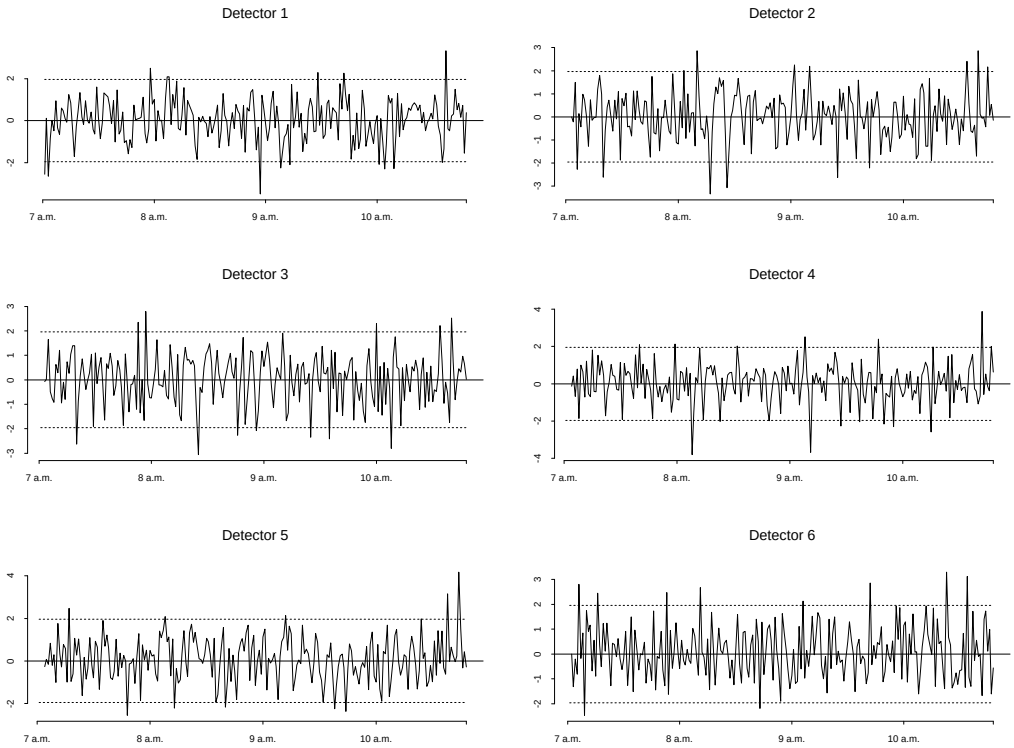


Figure 5.24: Thursday, the 8th of August; all the detectors: Standardized residuals from the one-step-ahead forecast values; the dotted lines show the 95% confidence intervals. In this plot $\delta_\beta = .95$.

Detector 07.02	log-likelihood $\delta_\beta = 1$	log-likelihood $\delta_\beta = .95$
1	-327.6243	...
2	-331.5013	-326.8692
3	-328.969	-324.5909
4	-327.421	-318.7826
5	-326.4524	-325.0183
6	-338.4178	-336.851

Detector 08.08	log-likelihood $\delta_\beta = 1$	log-likelihood $\delta_\beta = .95$
1	-327.1235	...
2	-328.866	-327.552
3	-326.866	-325.924
4	-330.878	-329.327
5	-330.07	-327.597
6	-335.7754	-335.5245

Now it can be better assessed that there actually exists a slight improvement in assuming a discount factor strictly less than unity.

In the end the conclusion seems to indicate that in order to better predict new out-of-sample days, the adaptation of the vector of parameters through the day can be better performed assuming a constant vector for the daily trend modeled through the spline function, but allowing the set of parameters of the true regression part to dynamically adapt at each instant, with a discount factor in the neighborhood of .95.

5.11 Summary

After a brief section presenting the translation of the old "one-shot" setting into the new one in which time plays the leading role, I develop an analysis for a different kind of data to address the issue of modeling and predicting the intensity of an undistinguished traffic flow along several links, and its development through time during the day. The data collected by a series of single-loop detectors consists of

counts of cars per minute along a section of Interstate Highway 5 (I-5), north of Seattle. The evolution of this quantity throughout 4 hours in the morning, each day of the week, is what the model attempts to explain.

An hierarchical model for the parameters of a regression composed in part by counts at preceding points in space and time and in part by a basis for a spline function to account for a daily trend is fitted, its parameters estimated and the dynamic updating procedure elaborated. Thus, the estimation procedure can adapt to new out-of-sample data, using the actual counts as they become observable to modify the prior estimate of trend and regression coefficients. The model performs well in both the fitting and the predicting arenas. Consistent patterns throughout the different days of the week are assessed, while the regression on the counts recorded at previous time/space points accounts for the variability in the flows throughout the day.

Chapter 6

Future Directions

6.1 About Seattle's Data Model

Although their full exploration is beyond the scope of the present work, I would like to cursorily indicate some possible extensions and developments of the part elaborated in the final chapter.

I already mentioned the idea of connecting the models for the single detectors into a complex multidimensional model able to account for correlation between detectors across space, besides the simple link through time that the regressions provide. Related to this issue, an interesting idea comes to mind in observing the quality of the process of traffic flow. Until now we regressed one detector on the counts of previous ones, but nothing precludes linking detectors even in reverse temporal order. What happens at an earlier time downstream can, and in the presence of congestion often does, influence a detector's counts, even more than what has been recorded upstream. Thus an analysis which allows this kind of reversed regression can be as meaningful as the more traditional one. As a first approximation, a simple change in the set of regressors for each detector's model, to include downstream counts at preceding time points would do, in this respect.

Another way of bracketing off the physical phenomenon itself is fitting an auto-regressive model. In the present analysis I try to "drive" with the cars recorded on the road: that is, I try to follow them through the network. It is conceivable however that a model which simply describes, without actually translating the material process, could be as effective as the present one.

I didn't study what happens in the afternoon. The issue of checking the validity of this model for the other portion of the traffic, in the reverse sense of flow, is still open.

An important piece of information available in the data set, the records of occupancy, has still to be considered. As I have already explained, occupancy is related to the average speed through the value of the counts. A better assessment of the actual speed in each of the time instants could be of assistance in respect to tracing back cars in previous recordings, and predicting what will occur in the immediately following time intervals.

6.2 Networks...

Possibly the main direction of further refinement in the original, fixed routing network area as I have approached it in Chapter 4 concerns the prior assessment of the λ components. Chapter 4 clearly highlighted how this issue becomes crucial in cases where problems of identification arise, often in a subtle, not immediately evident manner. In more general terms, priors that allow for dependencies among the different components of λ, or subsets of it, can improve the estimation results. In all but the most simple hypothetical scenarios the quality of the prior information appears decisive.

Aside from this, however, I would like to open here an entirely new and broad perspective, to which the network estimation approach can be applied and prove

valuable.

6.3 ...and Contingency Tables

Given a multi-way contingency table with known margins, the objective is filling each cell with counts consistent with the totals. This objective arises when trying to gain a multidimensional perspective of the status of a population by combining the results of a variety of studies each contributing a "side of the table". Even when drawing only upon disparate surveys, historical data and other sources of less than entirely precise information, it is often the case that a rough idea of individuals' probabilities of belonging to a specific cell can be gleaned.

As usual, the simple example of a two way contingency table is addressed first, as the basic block from which to start building. Thus, consider a m-by-n table that represents the two dimensional distribution of Y^{tot} individuals with respect to two variables A and B, whose realizations will be indicated by $(A_1, A_2, \ldots, A_m)$ and $(B_1, B_2, \ldots, B_n)$. Define the row and column marginal totals by

$$Y_1^A, Y_2^A, \ldots, Y_m^A$$

and

$$Y_1^B, Y_2^B, \ldots, Y_n^B.$$

So

$$\sum_{i=1}^{m} Y_i^A = \sum_{j=1}^{n} Y_j^B = Y^{tot}$$

and the table is usually represented by a $m \times n$ matrix:

$$\begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_m \end{array} \begin{array}{c} \begin{array}{cccc} B_1 & B_2 & \dots & B_n \end{array} \\ \begin{pmatrix} X_{1,1} & X_{1,2} & \dots & X_{1,n} \\ X_{2,1} & X_{2,2} & \dots & X_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{m,1} & X_{m,2} & \dots & X_{m,n} \end{pmatrix} \end{array}$$

There is another way of representing the same table, which immediately recalls the problem of **OD** flows estimation from link counts: As a first step, the entries of the table are lined up in a $(m \cdot n)$-dimensional vector

$$\mathbf{X} = (X_{1,1}, X_{1,2}, \dots, X_{m,n-1}, X_{m,n}),$$

and the margins of the two distributions in a $(m + n - 1)$-dimensional vector

$$\mathbf{Y} = (Y_1^A, \dots, Y_m^A, Y_1^B, \dots, Y_{n-1}^B)$$

(notice that the value of Y_n^B is not included, being redundant in a sense that will soon become clearer). Now, the $\mathbf{Y}$ vector can be viewed as the result of a number of linear combinations of the components of the $\mathbf{X}$ vector, i.e.

$$\mathbf{Y} = \mathbf{A}\mathbf{X},$$

with $\mathbf{A}$ easily written down as the following $0/1$ matrix of dimensions $(m + n - 1) \times (m \cdot n)$:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \dots & \mathbf{A}_m \\ \mathbf{I}|\mathbf{0} & \mathbf{I}|\mathbf{0} & \dots & \mathbf{I}|\mathbf{0} \end{pmatrix}.$$

The entries of this matrix are matrices themselves, and need some further explanations. For $i = 1, \dots, m$ the matrix $\mathbf{A}_i$ is a $m \times n$ matrix of 0's except the ith row, which is filled up with 1's. The matrices $\mathbf{I}$ of the bottom row are m identity matrices

of dimension $(n-1) \times (n-1)$ while the $\mathbf{0}$'s must be intended as columns of 0's, to be juxtapposed to the previous $\mathbf{I}$'s. Now it is clear why one of the marginal totals has been deleted: the matrix $(m+n) \times (m \cdot n)$ would be singular, being one of the counts in $\mathbf{Y}$ redundant, since the sums of the two marginals must both equal Y^{tot}; this way the matrix $\mathbf{A}$ is ensured to be full rank.

The direction in which this is tending should by now be obvious: an **OD** estimation problem has been reconstructed from a totally different setting. Moreover, it is easily shown how an initial admissible solution $\mathbf{X}^0 \geq 0$ to the underdetermined system is obtainable. Once this is given, the algorithm used so far to estimate flows along links can be applied to the estimation of counts in the cells of a multi-way array. There is in fact a perfect parallel in terms of the distributional assumption as well. The assumption of a Multinomial distribution for the entries of a table, with a fixed total, and the assumption of Poisson distribution for the same counts, with fixed marginals, are equivalent.

Finding an initial point for the iterative simulation to begin is particularly easy in the present setting. Here are the details. Still working on a two-dimensional table, re-arrange rows and columns so that the marginals are both in decreasing order. Now start filling up the last element, $X_{m,n}$ by the minimum between $(Y^A_m/n, Y^B_n/m)$ rounded up to the largest integer less than or equal to it. Once the cell is filled, decrease the totals of the appropriate row and column by it:

$$Y^A_m \to Y^A_m - X_{m,n},$$

$$Y^B_n \to Y^B_n - X_{m,n}.$$

and reapply the procedure to the cells along the border of $X_{m,n}$, leaving the diagonal element last. The denominators of the fractions correspond always to the number of cells in the row/column still to fill, the numerators to the "current totals" by row and by column. The procedure is iterated until the corner of the largest square subsection

of the matrix is reached and filled. The remaining cells are then filled up in sequence by columns (or rows in case of a matrix with $m > n$) starting from the closest to the section already filled up, and from the bottom (rightmost) element up to the top (leftmost) one.

This procedure is equivalent to filling up the rearranged matrix by repeatedly adding a unity to each cell, iteratively scanning all the cells from the top left corner to the bottom right corner, until the additions to the cell contents are consistent with the row/column totals. the bottom right part will always contain lower counts than the top left one, being always possible to divide the cells of the matrix in subsets of equal counts by non intersecting, monotonically non decreasing diagonal lines starting from the left or bottom side of the table and ending up in the right or top ones: given an entry $X_{i,j}$ of the table, the entries at its right and at its bottom are always less than or equal to it. This admissible solution can be used as a starting point where to initialize the chain of iterative simulations.

Extending these results from a two-way to a multi-way table is straightforward: the block-structure of the matrix $\mathbf{A}$ will be essentially the same, with a number of 1's in each column equal to the "ways" of the table, each cell now contributing to each of the dimensions of the multivariate distribution. Also the procedure for obtaining an initial set of counts is recyclable in a multidimensional setting.

In general any k-dimensional contingency table can be seen as a representation of a set of linear equations, with a number of constraints equal to the sum of the dimensions of the marginals, and reduced to linearly independent system of equations by deleting a number of constraints equal to $k - 1$.

It is worth pointing out that a more general setting is addressed by the algorithm, where some entries for the marginals are possibly not known, or less disaggregated ones are. It all boils down to writing the correct linear representation of the problem,

identifying a square subset of the full-rank matrix $\mathbf{A}$ and iteratively sampling from the conditionals of each of the random components of $\mathbf{X}$.

The presence of "prior information" on the cell probabilities can prove valuable in terms of binding the prior distribution of the Poisson rates to particular regions of the λ space, thus avoiding the problems of identification which sparse matrices, typicalin this kind of setting, may introduce.

As this brief sketch of the problem makes evident, this represents an entirely new field of application for the estimation procedure developed in the first part of this work, and the broad perspective it promises a range of interesting new research possibilities.

6.3.1 An example

From an article by Smith, Forster and McDonald [29], the table:

Wife's ethnicity: / Husband's ethnicity:	GB	IR	Scandi navian	D	I	P	Eur. Jew. Central	Eur. Jew. Eastern
British	314	63	10	15	0	1	1	0
Irish	27	625	2	5	0	0	0	0
Scandinavian	4	9	835	20	1	0	0	0
German	26	26	10	1096	0	4	0	0
Italian	3	6	0	4	477	1	0	0
Polish	1	0	0	7	0	421	0	0
Jewish (C.E.)	1	0	0	1	0	1	112	11
Jewish (E.E.)	1	0	0	1	0	1	30	347

represents the distribution of marriages by ethnicity of each spouse (men by row, women by column) among first-generation European immigrants to the USA at the turn of the century, taken from a 1/250 sample of the 1910 census manuscripts. Eight important immigrant groups were used because of their size and substantive interest.

I applied the network analysis to the table which I consider interesting because of its sparseness (endogamous marriages were the rule, therefore the observations are concentrated along the main diagonal), and because the nature of the issue under study suggests that informed prior opinion may contribute to the analysis.

The exploratory analysis, designed just to open up this problem area for demonstration here, began with a first study using Uniform priors for the rates λ and the data provided by the row and column totals of the full table.

The results substantiated the tendency of the likelihood function to favour a uniform distribution of small counts rather than peaks of counts in few cells and small or zero counts in others (See Chapter 4 for an extended discussion of this issue).

The tables below are the approximate posterior means of the Poisson rates for each cell and of the cell counts themselves, clearly illustrating the "uniform" feature:

.

$$\begin{pmatrix} 37.0 & 72.7 & 99.1 & 61.1 & 37.4 & 42.5 & 20.2 & 43.7 \\ 92.3 & 195.7 & 135.8 & 46.2 & 90.7 & 37.8 & 17.3 & 48.1 \\ 50.3 & 257.4 & 222.6 & 61.1 & 51.4 & 124.9 & 24.6 & 84.8 \\ 72.4 & 76.0 & 116.2 & 756.5 & 38.5 & 57.0 & 12.2 & 38.3 \\ 44.3 & 37.3 & 79.0 & 117.4 & 106.5 & 32.1 & 35.0 & 44.2 \\ 44.9 & 34.8 & 68.8 & 73.0 & 68.1 & 88.3 & 12.8 & 45.1 \\ 12.8 & 13.4 & 22.9 & 11.9 & 25.7 & 19.5 & 5.8 & 21.9 \\ 28.2 & 47.9 & 118.4 & 29.2 & 66.6 & 33.0 & 22.5 & 40.5 \end{pmatrix}$$

$$\begin{pmatrix} 14 & 81 & 77 & 46 & 62 & 53 & 9 & 62 \\ 114 & 141 & 173 & 101 & 19 & 33 & 21 & 57 \\ 50 & 303 & 165 & 118 & 38 & 65 & 41 & 89 \\ 130 & 69 & 123 & 640 & 9 & 154 & 4 & 33 \\ 13 & 47 & 85 & 130 & 172 & 18 & 21 & 5 \\ 33 & 9 & 130 & 78 & 56 & 55 & 14 & 54 \\ 11 & 18 & 13 & 32 & 14 & 8 & 8 & 22 \\ 12 & 61 & 91 & 4 & 108 & 43 & 25 & 36 \end{pmatrix}$$

These results come from a run of 1,000,000 iterations, of which the last 500 values of each batch of 10,000 were saved. As explained in Chapter 4, Section 4.1, this choice was made in order to break the high degree of correlation between subsequent steps of the chain, and yet gaining the full picture of the posterior distributions. The initial value for the chain was found by the method detailed in the preceeding section.

After this first reference analysis, a second exploratory study was based on Gamma priors for each Poisson rate. These take into account the nature of the problem, favoring high counts in the cells along the main diagonal and low or zero counts elsewhere, and allowing a low range of variation (Both versions of the algorithm have been described in Chapter 3 and Chapter 4). The following table gives details of the means for the single values of λ, arranged in the same format of the table of the observed counts. The value of the variance was chosen as 100 for the diagonal entries and 10 for the off-diagonal entries.

Wife's ethnicity: Husband's ethnicity:	GB	IR	Scandi navian	D	I	P	Eur. Jew. Central	Eur. Jew. Eastern
British	313	64	8	16	1	1	1	1
Irish	28	626	1	4	1	1	1	1
Scandinavian	4	8	839	20	1	1	1	1
German	24	24	8	1100	1	4	1	1
Italian	4	8	1	4	478	1	1	1
Polish	1	1	1	8	1	421	1	1
Jewish (C.E.)	1	1	1	1	1	1	112	12
Jewish (E.E.)	1	1	1	1	1	1	32	349

The results are in this case more in accordance with the real data. The following matrix shows the posterior means for the Poisson rates, this time showing the kind of behavior to be expected: high rates along the diagonal and lower rates in the cells elsewhere. Clearly, however, the sparsity of the off-diagonal entries of the data table

are not yet adequately reflected here.

$$\begin{pmatrix} 298.4 & 65.3 & 17.7 & 16.4 & 45.6 & 24.3 & 9.5 & 17.5 \\ 29.3 & 611.5 & 124.8 & 4.5 & 72.5 & 40.6 & 14.8 & 48.6 \\ 6.3 & 39.0 & 825.7 & 20.4 & 86.3 & 117.5 & 13.2 & 52.8 \\ 24.5 & 24.3 & 8.7 & 1099.6 & 1.5 & 4.3 & 1.4 & 1.6 \\ 6.2 & 22.1 & 115.6 & 4.5 & 462.5 & 23.2 & 11.2 & 50.5 \\ 96.9 & 40.2 & 41.6 & 8.2 & 24.5 & 405.6 & 8.4 & 13.5 \\ 19.1 & 12.6 & 12.5 & 1.4 & 10.4 & 8.7 & 97.3 & 12.7 \\ 21.3 & 75.6 & 54.1 & 1.1 & 16.6 & 13.8 & 32.7 & 333.6 \end{pmatrix}$$

Hence there is still room for improvement. For instance, it seems valuable to consider allowing for dependencies among the rates of the Poisson distributions, since the nature of the problem calls for it. Subgroups of ethnicities that can be considered closer, two groups of Jews, and possible insight in religious commonalities or diversities are pieces of information which may be useful in this framework, for building a more structured prior joint distribution. Partial exchangeability of the vector of λ rather than a more complex covariance structure, could be another path to follow. Nevertheless, the algorithm is of real benefit in this problem area. When combined as a component of a real model, prior structure in an application of partially observed contingency tables, has real utility and promise. I leave these issues open here, new and important topics for future development.

Bibliography

[1] Bell,M. (1991), *The estimation of origin-destination matrices by constrained generalized least squares.* Transportation Research, 25B, 13-22.

[2] Berka, S., and Boyce, D.E. (1994), *Implementation and solution of a large asymmetric network equilibrium model.* Working Paper, Urban Transportation Center, University of Illinois at Chicago.

[3] Best, N.G., Cowles, M.K., and Vines, S.K. (1995), *CODA: Convergence diagnosis and output analysis software for Gibbs sampler output,*(Version 0.3). Medical Research Council Biostatistics Unit, Cambridge.

[4] Bierlaire, M. & Toint, P.L. (1994), *Meuse, an origin-destination matrix estimator that exploits structure.* Transportation Research, 29B, 47-60

[5] Cascetta, E. (1984), *Estimation of trip matrices from traffic counts and survey data, a Generalized Least Squares estimator.* Transportation Research, 18B, 289-299.

[6] Cascetta, E. & Nguyen, S. (1988), *A unified framework for estimating or updating origin/destination matrices from traffic counts.* Transportation Research, 22B, 437-455.

[7] Chen, Y. (1994), *Bilevel programming problems: analysis, algorithms and applications.* Ph.D. thesis, report CRT-984, Centre de récherche sur les transports (CRT), University of Montreal, Montreal, Québec, Canada.

[8] Cumbus, C., Damien, P. & Walker, S. G. *Sampling truncated Poisson and Multivariate Normals via a Gibbs sampler.* University of Michigan Business School Report n.9316-34.

[9] Drissi-Kaitouni, O. & Lundgren, J. (1992), *Bilevel origin-destination matrix estimation using a descent approach.* Technical Report LiTH-MAT-R-92-49, Department of Mathematics, Linkoping University, Sweden.

[10] Erlander, S., Jornsten, K.O. & Lundgren, J.T. (1984), *On the estimation of trip matrices in the case of missing and uncertain data.* Technical report LiTH-MAT-R-84-20, Department of Mathematics, Linkoping University, Sweden.

[11] Erlander, S., Nguyen, S. & Stewart, N. (1979), *On the calibration of the combined distribution/assignment model.* Transportation Research, 13B, 259-267.

[12] Fisk, C.S. (1988), *On combining maximum entropy trip matrix estimation with user-optimal assignment.* Transportation Research, 22B, 69-79.

[13] Fisk, C.S. & Boyce, D.E. (1983), *A note on trip matrix estimation from link traffic count data.* Transportation Research, 17B, 245-250.

[14] Florian, M. & Chen, Y. (1993), *A coordinate descent method for the bilevel OD matrix adjustment problem.* Presented at the IFORS Conference in Lisbon, Portugal. Available as an earlier version as Publication CRT-750 at the CRT, University of Montreal, Montreal, Quebec, Canada.

[15] Gelfand, A. E., and Smith, A. F. M. (1990), *Sampling-based approaches to calculating marginal densities.* Journal of the American Statistical Association, 85, 398-409.

[16] Gelfand, A. E., Hills, S. E., Racine-Poon A., and Smith, A. F. M. (1990), *Illustration of Bayesian inference in Normal data models using Gibbs sampling.* Journal of the American Statistical Association, 85, 972-985.

[17] Gilks, W.R., Richardson, S., and Spiegelhalter, D.J. (1996), *Markov Chain Monte Carlo in Practice.* Chapman & Hall, London.

[18] Golub, G.H., and Van Loan, C.F. (1983), *Matrix Computations,* Johns Hopkins University Press, Baltimore.

[19] Jornsten, K. & Nguyen, S. (1979), *On the estimation of a trip matrix from network data.* Technical report LiTH-MAT-R-79-36, Department of Mathematics, Linkoping University, Sweden, (revised April 1983).

[20] Leblanc, L.J. & Farhangian, K. (1982), *Selection of a trip table which reproduces observed link flows.* Transportation Research, 16B, 83-88.

[21] Lindley, D. V., and Smith, A. F. M. (1972), *Bayes estimates for the linear model,* (with discussion), J. R. Statist. Soc. B, 34, 1-41.

[22] Maher, M.J. (1983), *Inferences on trip matrices from observations on link volumes: a Bayesian approach.* Transportation Research, 20B, 435-447.

[23] McNeil, S., and Hendrickson, C. (1985), *A regression formulation of the matrix estimation problem.* Transportation Science, 19, 278-292.

[24] Nguyen, S. (1977), *Estimation of an OD matrix from network data: A network*

equilibrium approach. Publication no. 60, Centre de récherche sur les transports, Université de Montréal, Québec, Canada.

[25] Sen, A. (1986), *Maximum likelihood estimation of gravity model parameters.* Journal of Regional Science, 26, 461-474.

[26] Sheffi, Y. (1985), *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods.* Prentice-Hall: New Jersey.

[27] Smith, A. F. M. (1973), *A general Bayesian linear model.* J. R. Statist. Soc. B, 35, 1-41.

[28] Smith, T.E. (1987), *Poisson gravity models of spatial flows.* Journal of Regional Science, 27, 315-340.

[29] Smith,P.W.F, Forster, J.J., and McDonald, J.W. (1996), *Monte Carlo exact tests for square contingency tables.* J. R. Statist. Soc. A, 159, 309-321.

[30] Spiess, H. (1987), *A maximum-likelihood model for estimating origin-destination matrices.* Transportation Research, 21B, 395-412.

[31] Spiess, H. (1990), *A descent based approach for the origin-destination matrix adjustment problem.* Publication CRT-693 at the CRT, University of Montreal, Montreal, Quebec, Canada.

[32] S-Plus (1993), S-PLUS Guide to Statistical and Mathematical Analysis (Version 3.2), StatSci: MathSoft, Inc., Seattle.

[33] Tanner, M.A. (1993), *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions.* 2nd Edition, New York: Springer-Verlag

[34] Tierney, L. (1994), *Markov chains for exploring posterior distributions,* (with discussion), Ann. Statist., 22, 1701-1762.

[35] Van Aerde, M., Hellinga, B., Baker, M., and Rakha, H. (1996), *INTEGRATION:* An overview of traffic simulation features. Discussion paper, Department of Civil Engineering, Queen's University, Kingston, Canada.

[36] Van Zuylen, H.J. & Willumsen, L.G. (1980), *The most likely trip matrix estimated from traffic counts.* Transportation Research, 14B, 281-293.

[37] Vardi, Y. (1996), *Network tomography: Estimating source-destination traffic intensities from link data.* Journal of the American Statistical Association, 91, 365-377.

[38] Vardi, Y. & Lee, D. (1993), *From image deblurring to Optimal Investments: Maximum Likelihood Solutions for Positive Linear Inverse Problems,* (with discussion), Journal of the Royal Statistical Society, Ser. B, 55, 569-612.

[39] Yang, H., Iida, Y. & Sasaki, T. (1994), *The equilibrium based Origin-Destination matrix estimation problem.* Transportation Research, 28B, 23-33.

[40] Yang, H., Sasaki, T., Iida, Y. & Asakura,Y. (1992), *Estimation of origin-destination matrices from link traffic counts on congested networks.* Transportation Research, 26B, 417-434.

[41] West, M. (1994), *Statistical inference for gravity models in transportation flow forecasting.* Discussion paper 94-40, ISDS, Duke University.

[42] West, M. & Harrison, J. (1997), *Bayesian Forecasting and Dynamic Models.* 2nd Edition, New York: Springer-Verlag.

[43] Wolfram, S. (1988), *Mathematica. A System for Doing Mathematics by Computer,* New York: Addison-Wesley.

Biography

Born in Sermide (Italy) the 10th of February 1966.
Graduated from Bocconi University, Milan (Italy) in 1992.
From 1993 to present at ISDS.