Time-Varying Network Tomography: Router Link Data

Jin Cao Drew Davis Scott Vander Wiel Bin Yu

February 29, 2000

Abstract

The origin-destination (OD) traffic matrix of a computer network is useful for solving problems in design, routing, configuration debugging, monitoring, and pricing. Directly measuring this matrix is not usually feasible but less informative link measurements are easy to obtain.

This work studies the inference of OD byte counts from link byte counts measured at router interfaces under a fixed routing scheme. A basic model of the OD counts assumes that they are independent normal over OD pairs and iid over successive measurement periods. The normal means and variances are functionally related through a power law. We deal with the time-varying nature of the counts by fitting the basic iid model locally using a moving data window. Identifiability of the model is proved for router link data and maximum likelihood is used for parameter estimation. The OD counts are estimated by their conditional expectations given the link counts and estimated parameters. OD estimates are forced to be positive and to harmonize with the link count measurements and the routing scheme. Finally, ML estimation is improved by using an adaptive prior.

Proposed methods are applied to two simple networks at Lucent Technologies and found to perform well. Furthermore, the estimates are validated in a single-router network for which direct measurements of origin-destination counts are available through special software.

Keywords: EM, Filtering, Normal, Inverse Problem, MLE, Link Data, Network Traffic, Smoothing, Variance Model.

[1] Jin Cao, Drew Davis and Scott Vander Wiel are Members of Technical Staff at Bell Laboratories, Lucent Technologies. Bin Yu is Member of Technical Staff at Bell Laboratories, Lucent Technologies and Associate Professor of Statistics at University of California at Berkeley.

1 Introduction

Research on computer network monitoring and management is exploding. A statistical perspective is needed for solving many of these problems either because the desired measurements are not directly available or because one is concerned about trends that are buried in notoriously noisy data. The problem we consider in this paper has both of these aspects: indirect measurements and a weak signal buried in high variability.

In a local area network (LAN), routers and switches direct traffic by forwarding data packets between nodes according to a routing scheme. Edge nodes directly connected to routers (or switches) are called origins or destinations, and they do not usually represent single users but rather groups of users or hosts that enter a router on a common interface. An edge node is usually both an origin and a destination depending on the direction of the traffic. The set of traffic between all pairs of origins and destinations is conventionally called a traffic matrix, but in this paper we usually use the term origin-destination (OD) traffic counts to be specific. On a typical network the traffic matrix is not readily available, but aggregated link traffic measurements are.

The problem of inferring the OD byte counts from aggregated byte counts measured on links is called *network tomography* by Vardi (1996). The similarity to conventional tomography lies in the fact that the observed link counts are linear transforms of unobserved OD counts with a known transform matrix determined by the routing scheme. Vardi (1996) studies the problem for a network with a general topology and uses an iid Poisson model for the OD traffic byte counts. He gives identifiability conditions under the Poisson model and discusses using the EM algorithm on link data to estimate Poisson parameters in both deterministic and Markov routing schemes. To mitigate the difficulty in implementing the EM algorithm under the Poisson model, he proposes a moment method for estimation and briefly discusses the normal model as an approximation to the Poisson. Tebaldi and West (1998) follow up with a Bayesian perspective and an MCMC implementation, but only deal with link counts from a single measurement interval. Vanderbei and Iannone (1994) apply the EM algorithm but also use a single set of link counts.

This paper focuses on time-varying network tomography. Based on link byte counts

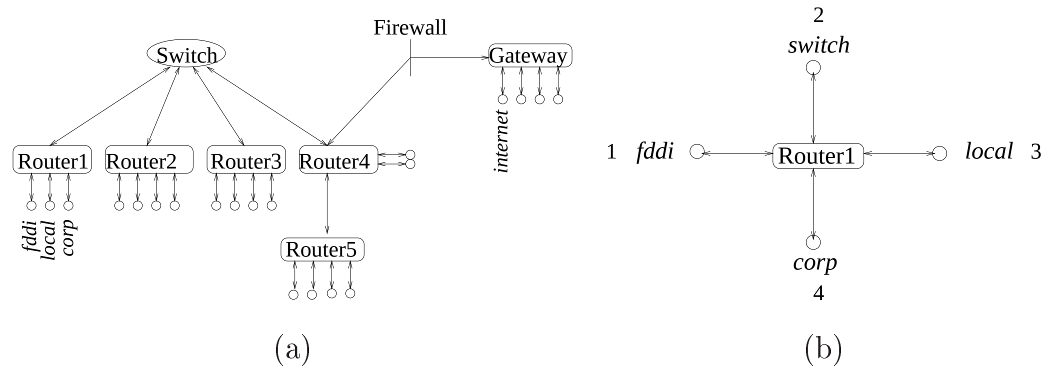

Figure 1: (a) A router network at Lucent Technologies; (b) The network around *Router1*.

measured at the router interfaces and under a fixed routing scheme, the time-varying traffic matrix is estimated. The link counts are readily available through the Simple Network Management Protocol (SNMP) which is provided by nearly all commercial routers. The traffic matrix or OD counts, however, are not collected directly by most LANs. Such measurements typically require specialized router software and hardware dedicated to data collection. This is not practical for most networks. Nevertheless, network administrators often need to make decisions that depend on the traffic matrix. For example, if a certain link in the network is overloaded, one might either increase the capacity of that link or adjust the routing tables to make better use of the existing infrastructure. The traffic matrix can help determine which approach would be more effective and locate the source of unusually high traffic volumes. An Internet Service Provider can also use the traffic matrix to determine which clients are using their network heavily and charge them accordingly rather than using the common flat rate pricing scheme.

Two standard traffic measurements from SNMP are (1) incoming byte count, the numbers of bytes received by the router from each of the interfaces connected to network links; and (2) outgoing byte count, the numbers of bytes that the router sends on each of its link interfaces. We collect these incoming and outgoing *link counts* at regular five minute intervals from each of the routers in a local network at Lucent diagrammed in Figure 1(a). The six boxes represent network routers. The oval is a switch with a similar function but no capabilities for data collection. Individual nodes hanging from the routers connect to subnetworks of users, other portions of the corporate network, shared backup systems, and so forth. A data

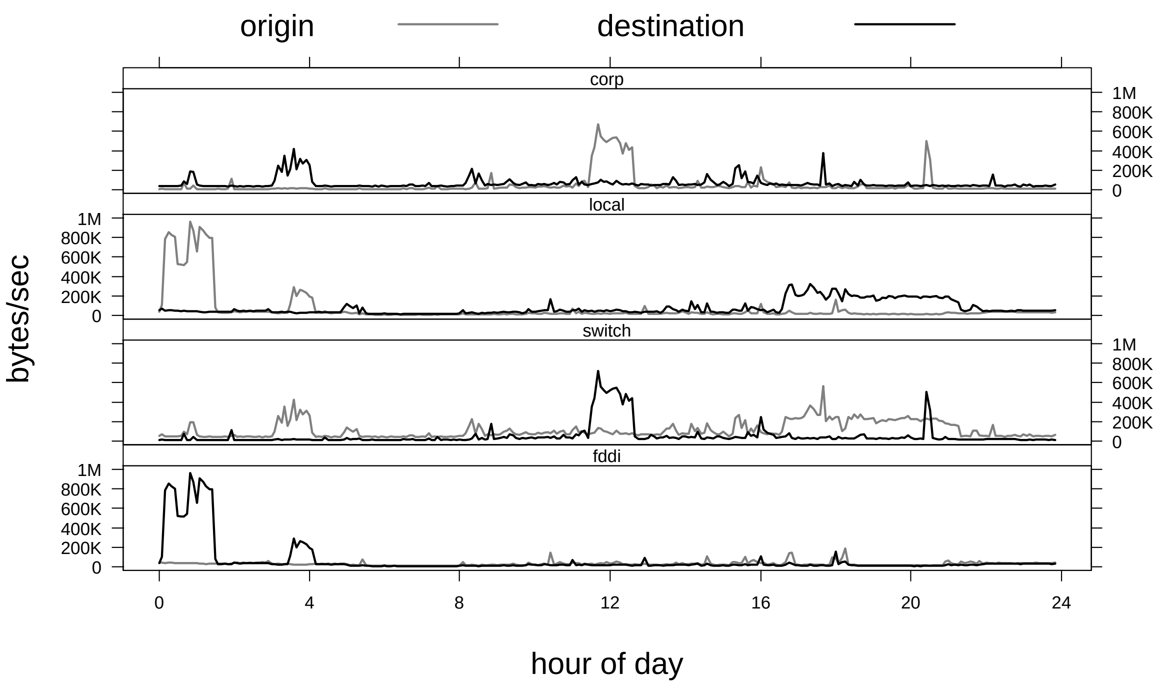

Figure 2: Link measurements for the *Router1* subnetwork illustrated in Figure 1(b). The average number of bytes per second is measured over consecutive 5 minute intervals for the 24 hour period of February 22, 1999. Each panel represents a node. Origin (gray) and destination (black) series are superposed. Matching patterns suggest traffic flows. For example, at hour 12 the origin *corp* pattern matches that for destination *switch*.

packet originating, for example, from one of the *Router1* nodes and destined for the public Internet would be routed from *Router1* to the *Switch*, to *Router4*, through the *Firewall* to *Gateway* and then to the Internet. A network routing scheme determines this path.

Figure 1(b) illustrates the subnetwork around *Router1*. For several reasons this paper uses this simple network as the main example to illustrate the methodologies. One reason is that we have access to the system management staff for gaining access to data and for understanding details of the network, and at the same time we have available validation data to check our estimates. Another reason for studying a small problem is that in larger networks, it often makes sense to group nodes and estimate traffic between node groups. In Section 6, we discuss how estimation for all OD pairs in a large network can be done via a series of smaller problems on aggregated networks. We do not mean to imply, however, that all realistic problems can be reduced to our 4×4 example. Additional work is needed to scale these methods up to work on larger networks.

The 4 links on *Router1* give rise to 8 link counts, with incoming and outgoing measurements for each interface. Time series plots of the link counts are shown in Figure 2. The byte counts are highly variable as is typical for data network traffic. The traffic levels rise and fall suddenly and in ways that do not suggest stationarity. It is striking that a traffic pattern at an origin link interface can often be matched to a similar one at a destination interface and this can give a pretty good idea of where traffic flows. For example, origin *local* matches destination *fddi* at hour 1 and origin *switch* matches destination *corp*at hour 4. Such pattern matching may seem undemanding but the $4 \times 4 = 16$ OD count time series that we want to estimate are obviously not completely determined from the 8 observed link count series. There is often a large range of feasible OD estimates consistent with the observed link data. Our model-based estimates typically have have errors that are much less than these feasible ranges. Section 5 gives a specific example. Furthermore, visual pattern matching does not work well for the two-router network discussed in Section 6.

The rest of the paper is organized as follows. Section 2 gives a basic independent normal model for OD byte counts in which means and variances are functionally related through a power law. The section also proposes to estimate these counts using approximate conditional expectations given the link measurements, the estimated ML parameters, and the fact that the OD counts are positive. An iterative refitting algorithm ensures that estimates meet the routing constraints. Section 3 deals with the time-varying nature of traffic data by using a moving data window to fit the basic model locally and presents exploratory plots to show that the model assumptions are plausible for our example data. Section 4 develops a refinement of the moving iid model by supplementing the moving window likelihood with an adaptive prior distribution which is derived from modeling the parameter changes using independent random walks. Estimates from this model are then validated in Section 5 using direct measurements of OD byte counts obtained by running specialized software on the router and dumping data to a nearby workstation. Section 6 provides a brief application of our methods to a two-router network. Section 7 concludes with directions for further work. The appendix gives proofs of the model identifiability.

2 Normal Model and Estimation Methods

2.1 Basic model

In this section we describe a model for a single vector of link counts measured at a given time; the counts at time t reflect the network traffic accruing in the unit interval (eg. a 5 minute interval) ending at t. Subsequently, in Section 2.3, we discuss fitting this model to a sample of such link count vectors measured over successive time intervals.

Let $\mathbf{x}_t = (x_{t,1}, \ldots, x_{t,I})'$ denote the vector of *unobserved* link counts (average number of bytes per second, for example) of all OD pairs in the network at a given time t. In the *Router1* network, $\mathbf{x}_t$ has $I = 16$ elements. One element, for example, corresponds to the number of bytes originating from *local* and destined for *switch*. Although we have used the customary terminology "traffic matrix," it is convenient to arrange the quantities for OD pairs in a single vector, $\mathbf{x}_t$.

Let $\mathbf{y}_t = (y_{t,1}, \ldots, y_{t,J})'$ be the vector of *observed* incoming and outgoing byte counts at each router link interface. For example, one element of $\mathbf{y}_t$ corresponds to the number of bytes originating from *local* regardless of their destination. With 4 nodes, there are 4 incoming links and 4 outgoing links. In principle, the router is neither a source nor a destination. This implies that the total count of incoming bytes should be equal to the total count of outgoing bytes and thus the 8 link counts are linearly dependent. To avoid redundancies we remove the outgoing link count for the 4th interface, leaving $J = 7$ linearly independent link measurements.

In general, each element of $\mathbf{y}_t$ is a sum of certain elements of $\mathbf{x}_t$ with the exact relationship determined by the routing scheme. In matrix notation, these relationships are expressed as

$$\mathbf{y}_t = \mathbf{A}\mathbf{x}_t,$$

where $\mathbf{A}$ is a $J \times I$ incidence matrix embodying the routing scheme. Typically J is much less than I because there are many more node pairs than links. Each column of $\mathbf{A}$ corresponds to an OD pair and indicates which links are used to carry traffic between that pair of nodes. We assume $\mathbf{A}$ to be fixed and known.

In the case of the single-router network around *Router1* as shown in Figure 1(b), the routing matrix has entries of 0 except in the places indicated below

$$\mathbf{A} \;=\; \begin{array}{l} \text{orig-1} \\ \text{orig-2} \\ \text{orig-3} \\ \text{orig-4} \\ \\ \text{dst-1} \\ \text{dst-2} \\ \text{dst-3} \end{array} \begin{pmatrix} 1 & 1 & 1 & 1 & & & & & & & & & & & & \\ & & & & 1 & 1 & 1 & 1 & & & & & & & & \\ & & & & & & & & 1 & 1 & 1 & 1 & & & & \\ & & & & & & & & & & & & 1 & 1 & 1 & 1 \\ & & & & & & & & & & & & & & & \\ 1 & & & & 1 & & & & 1 & & & & 1 & & & \\ & 1 & & & & 1 & & & & 1 & & & & 1 & & \\ & & 1 & & & & 1 & & & & 1 & & & & 1 & \end{pmatrix}$$

Columns are associated with OD pairs. The second column, for example, is for origin 1 and destination 2. The first row of the equation $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t$ corresponds to the statement that the observed byte count for traffic arriving on interface 1 is equal to the sum of the unobserved counts for traffic originating at node 1 and destined for nodes 1, 2, 3, and 4.

The unobserved OD byte counts, $\mathbf{x}_t$ at time t are modeled as a vector of independent normal random variables

$$\mathbf{x}_t \sim \text{normal}(\boldsymbol{\lambda},\ \boldsymbol{\Sigma}), \tag{1}$$

implying that the observed link byte counts, $\mathbf{y}_t$, have distribution

$$\mathbf{y}_t = \mathbf{A}\mathbf{x}_t \sim \text{normal}(\mathbf{A}\boldsymbol{\lambda},\ \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}'). \tag{2}$$

The parameters are $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_I)', \lambda_i > 0$ and

$$\boldsymbol{\Sigma} = \phi \ \text{diag}(\sigma^2(\lambda_1), \ldots, \sigma^2(\lambda_I)), \tag{3}$$

where $\phi > 0$ is a scale parameter and the function $\sigma^2(\cdot)$ describes a relationship which is assumed to be known between the mean and variance. Although $\boldsymbol{\Sigma}$ depends on $\boldsymbol{\lambda}$ and ϕ, we suppress this from the notation. Equations (1)–(3) define the general model, but this paper concentrates on a specific power law form for σ^2:

$$\sigma^2(\lambda) = \lambda^c \tag{4}$$

where c is a constant.

The normal approximation to the Poisson model mentioned by Vardi (1996) is the special case $\phi = 1$ and $c = 1$. It is necessary, however, to account for extra Poisson variability in our byte count data. Section 3.2 presents some exploratory data analysis suggesting that the power law (4) controls the relation between mean and variance. We regard ϕ primarily as a nuisance parameter to account for extra Poisson variation; but a change in ϕ can also be used to accommodate a change of the units of traffic measurements from "bytes" to "bytes/sec" which may be more intuitive to system administrators. In other words, our model is scale-invariant while the Poisson model is not. The power c is not formally estimated in our approach, but Section 3 shows how to select a reasonable value and demonstrates the differences between two choices: $c = 1$ and 2.

Normal distributions describe continuous variables. Given the high speed of today's network and 5 minute measurement intervals, the discreteness of byte counts can be ignored. However, the normal model is only an approximation if only for the reason that $\mathbf{x}_t$ is positive. But if $\mathbf{x}_t$ is Poisson distributed with a relatively large mean, then the approximation is good. Working in the normal family obviously makes the distribution of observation $\mathbf{y}_t$ easy to handle and leads to computational efficiency as we shall see below. Subsection 3.2 explores model assumptions for our data.

2.2 Identifiability

Model (2)–(4) with a fixed power c is identifiable. This is stated in the following theorem and corollary. Proofs are given in Appendix A.

Theorem 2.1 *Let* $\mathbf{B}$ *be the* $[J(J+1)/2] \times I$ *matrix whose rows are the rows of* $\mathbf{A}$ *and the component-wise products of each different pair of rows from* $\mathbf{A}$*. Model (2)–(4) is identifiable if and only if* $\mathbf{B}$ *has full column rank.*

Corollary 2.2 *For byte counts from router interface data,* $\mathbf{B}$ *has full column rank and thus model (2)–(4) is identifiable.*

An intuitive explanation of the identifiability for the router interface data follows from hints that surface in the proof of the corollary. For the ith origin-destination pair, let y_o represent the number of incoming bytes at the origin interface and let y_d represent the number of outgoing bytes at the destination interface. The only bytes that contribute to both of these counts are those from the ith OD pair, and thus $\mathrm{Cov}(y_o, y_d) = \phi\lambda_i^c$. Therefore

λ_i is determined up to the scale ϕ. Additional information from $\mathrm{E}(\mathbf{y}_t)$ identifies the scale and identifiability follows. A similar reasoning justifies Vardi's moment estimator of $\boldsymbol{\lambda}$.

2.3 Estimation of $\boldsymbol{\lambda}$ based on IID Observations

The basic model given by (2)–(4) is identifiable, but reasonable estimates of the parameters require information to be accumulated over a series of measurements. We now describe a maximum likelihood analysis based on T iid link measurement vectors $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$ to infer the set of OD byte count vectors $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$. This simple iid model forms the basis for time-varying estimates in Section 3 where we apply it locally to byte count vectors over a short windows of successive time points.

Under assumption (2) and for $\boldsymbol{\theta} = (\boldsymbol{\lambda}, \phi)$, the log likelihood is

$$l(\boldsymbol{\theta} \mid \mathbf{Y}) = -\frac{T}{2} \log |\mathbf{A\Sigma A}'| - \frac{1}{2}\sum_{t=1}^{T}(\mathbf{y}_t - \mathbf{A}\boldsymbol{\lambda})'(\mathbf{A\Sigma A}')^{-1}(\mathbf{y}_t - \mathbf{A}\boldsymbol{\lambda}).$$

Let $\mathbf{W} = \mathbf{A}'(\mathbf{A\Sigma A}')^{-1}\mathbf{A}$ with the ijth element w_{ij}, and $\dot{\sigma}_i^2$and $\ddot{\sigma}_i^2$ respectively denote the first and second derivatives of $\sigma^2(\lambda)$ evaluated at λ_i. Under assumption (4),

$$\dot{\sigma}_i^2 = c\lambda_i^{c-1}, \qquad \ddot{\sigma}_i^2 = c(c-1)\lambda_i^{c-2}.$$

The $I \times I$ Fisher information matrix $-\mathrm{E}(\partial^2 l/\partial \boldsymbol{\lambda}^2)$ for $\boldsymbol{\lambda}$ has entries:

$$-\mathrm{E}\left(\frac{\partial^2 l}{\partial\lambda_i \partial\lambda_j}\right) = T(w_{ij} + \frac{1}{2}\phi^2 \dot{\sigma}_i^2 \dot{\sigma}_j^2 w_{ij}^2), \tag{5}$$

which provides insight into the source of information in estimating $\boldsymbol{\lambda}$. The first term is the information about $\boldsymbol{\lambda}$ that would come from a model in which $\boldsymbol{\Sigma}$ had no relation to $\boldsymbol{\lambda}$. The second term brings in the additional information about $\boldsymbol{\lambda}$ that is provided by the covariance of $\mathbf{y}_t$. Because $\mathrm{rank}(\mathbf{W}) = \mathrm{rank}(\mathbf{A})$ and $\mathbf{A}$ is generally far from full column rank, the Fisher information matrix about $\boldsymbol{\lambda}$ would be singular without the second term of (5). Thus, it is the covariance of $\mathbf{y}_t$ that provides the crucial information needed in estimating $\boldsymbol{\lambda}$. This is reminiscent of the role that the covariance played in proving identifiability. In fact, it can be proved that the matrix with entries w_{ij}^2 is positive definite as long as $\boldsymbol{\lambda} > \mathbf{0}$ and this implies the Fisher information matrix is non-singular.

Because $\boldsymbol{\Sigma}$ is functionally related to $\boldsymbol{\lambda}$ there are no analytic solutions to the likelihood equations. We turn to derivative-based algorithms to find the maximum likelihood estimate (MLE) of the parameters ϕ and $\boldsymbol{\lambda}$ subject to the constraints $\phi > 0$ and $\boldsymbol{\lambda} > \mathbf{0}$. We find it useful to start the numerical search using the EM algorithm (Dempster *et al.*, 1977), with the complete data defined as the T unobserved byte count vectors $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ of the OD pairs. Note that the complete data log-likelihood is the familiar normal form

$$l(\boldsymbol{\theta} \mid \mathbf{X}) = -\frac{T}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{t=1}^{T}(\mathbf{x}_t - \boldsymbol{\lambda})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}_t - \boldsymbol{\lambda}).$$

Let $\boldsymbol{\theta}^{(k)}$ be the current estimate of the parameter $\boldsymbol{\theta}$. Quantities that depend on $\boldsymbol{\theta}^{(k)}$ are denoted with a superscript (k). The usual EM conditional expectation function Q is

$$\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) &= \mathrm{E}(l(\boldsymbol{\theta} \mid \mathbf{X}) \mid \mathbf{Y}, \boldsymbol{\theta}^{(k)}) \\
&= -\frac{T}{2}(\log|\boldsymbol{\Sigma}| + \mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{R}^{(k)})) - \frac{1}{2}\sum_{t=1}^{T}(\mathbf{m}_t^{(k)} - \boldsymbol{\lambda})'\boldsymbol{\Sigma}^{-1}(\mathbf{m}_t^{(k)} - \boldsymbol{\lambda}),
\end{aligned}$$

where
$$\begin{aligned}
\mathbf{m}_t^{(k)} &= \mathrm{E}(\mathbf{x}_t \mid \mathbf{y}_t, \boldsymbol{\theta}^{(k)}) = \boldsymbol{\lambda}^{(k)} + \boldsymbol{\Sigma}^{(k)}\mathbf{A}'(\mathbf{A}\boldsymbol{\Sigma}^{(k)}\mathbf{A}')^{-1}(\mathbf{y}_t - \mathbf{A}\boldsymbol{\lambda}^{(k)}), \\
\mathbf{R}^{(k)} &= \mathrm{Var}(\mathbf{x}_t \mid \mathbf{y}_t, \boldsymbol{\theta}^{(k)}) = \boldsymbol{\Sigma}^{(k)} - \boldsymbol{\Sigma}^{(k)}\mathbf{A}'(\mathbf{A}\boldsymbol{\Sigma}^{(k)}\mathbf{A}')^{-1}\mathbf{A}\boldsymbol{\Sigma}^{(k)},
\end{aligned} \tag{6}$$

are the conditional mean and variance of $\mathbf{x}_t$ given both $\mathbf{y}_t$ and the current estimate $\boldsymbol{\theta}^{(k)}$. To complete the M step, we need to maximize the Q function with respect to $\boldsymbol{\theta}$. Let

$$a_i^{(k)} = r_{ii}^{(k)} + \frac{1}{T}\sum_{t=1}^{T}(m_{t,i}^{(k)})^2, \qquad b_i^{(k)} = \frac{1}{T}\sum_{t=1}^{T}m_{t,i}^{(k)}.$$

It can be shown that equations $\partial Q/\partial\boldsymbol{\theta} = \mathbf{0}$ are equivalent to

$$\begin{cases} 0 = c\phi\lambda_i^c + (2-c)\lambda_i^2 - 2(1-c)\lambda_i b_i^{(k)} - c a_i^{(k)}, \; i = 1, \ldots, I \\ 0 = \sum_{i=1}^{I}\lambda_i^{-c+1}(\lambda_i - b_i^{(k)}) \end{cases}. \tag{7}$$

The quantities $a_i^{(k)}$ are non-negative by definition and, with this in mind, it is straightforward to show that nonnegative solutions $\boldsymbol{\lambda}$ and ϕ to equation (7) always exist, even though they must generally be found numerically. Let $\mathbf{f}(\boldsymbol{\theta}) = (f_1(\boldsymbol{\theta}), \ldots, f_{I+1}(\boldsymbol{\theta}))$ be the right-hand sides of the above equations. We shall use the one-step Newton-Raphson algorithm to update $\boldsymbol{\theta}^{(k)}$ (Lange, 1995):

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - [\dot{\mathbf{F}}(\boldsymbol{\theta}^{(k)})]^{-1}\mathbf{f}(\boldsymbol{\theta}^{(k)}),$$

where $\dot{\mathbf{F}}$ is the Jacobian of $\mathbf{f}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$:

$$\frac{\partial f_i}{\partial \lambda_j} = \delta_{ij}(\phi c^2 \lambda_i^{c-1} + 2(2-c)\lambda_i - 2(1-c)b_i^{(k)}), \qquad \frac{\partial f_i}{\partial \phi} = c\lambda_i^c, \qquad i = 1, \ldots, I$$
$$\frac{\partial f_{T+1}}{\partial \lambda_j} = (2-c)\lambda_i^{1-c} - (1-c)\lambda_i^{-c} b_i^{(k)}, \qquad \frac{\partial f_{T+1}}{\partial \phi} = 0.$$

In general the Newton-Raphson steps do not guarantee $\boldsymbol{\theta}^{(k)} > \mathbf{0}$ but in the special case of $c = 1$ or $c = 2$, given ϕ, we can explicitly find a positive solution $\boldsymbol{\lambda}$ to (7) and use fractional Newton-Raphson steps on ϕ when necessary to prevent negative solutions.

Convergence of the above modified EM algorithm has been proved (Lange, 1995). However, it is usually quite slow in practice due to the sub-linear convergence of the EM algorithm (Dempster *et al.*, 1977). Second-order methods based on quadratic approximations of the likelihood surface have faster convergence rates. There are many such algorithms and we use the one in the S function `ms` (Chambers and Hastie, 1993) which is based on the published algorithm of Gay (1983). Our implementation uses analytical derivatives of the likelihood surface up to second order. Based on this information, the algorithm derives a quadratic approximation to the likelihood surface and uses a model trust region approach in which the quadratic approximation is only trusted in a small region around the current search point. With this algorithm, as with the EM algorithm, the likelihood function increases at each subsequent iteration. Since this algorithm is designed for unconstrained optimization, we reparameterize the likelihood function using $\boldsymbol{\eta} = (\log(\boldsymbol{\lambda}), \log(\phi))$ and supply to `ms` the first derivatives and second derivatives in terms of $\boldsymbol{\eta}$. We summarize the numerical algorithm as follows: (i) Initialize $\boldsymbol{\theta} = \boldsymbol{\theta}_0$; (ii) Update $\boldsymbol{\theta}$ using Newton Raphson EM steps until the change in $l(\boldsymbol{\theta})$ is small; (iii) Update $\boldsymbol{\theta}$ using a second order method (like `ms` in S) until convergence is declared.

The choice of starting point is fairly arbitrary for the EM iterations. For $\boldsymbol{\lambda}$, we use a constant vector in the center of the region; that is, $\boldsymbol{\lambda}_0 = a_0\mathbf{1}$ where a_0 solves $\mathbf{1}' A \boldsymbol{\lambda}_0 = \mathbf{1}' \sum_1^T \mathbf{y}_t / T$. In the case of $c = 1$, $\phi = \mathrm{Var}(y_{t,j})/\mathrm{E}(y_{t,j})$ for any $j = 1, \ldots, J$. A moment estimator based on this is used for the starting value of ϕ; similar ideas are used to give starting values of ϕ in general cases of c. Our experience is that this easily computed starting point gives stable performance. A more complex choice is to use a moment estimator like that proposed by Vardi (1996), but for the mean variance relation (4). Computations take

6 seconds per MLE window on the *Router1* network with 16 OD pairs using Splus 3.4 on a shared SGI Origin 2000 with 200 MHZ processors. The computations to produce Figure 5, for example, take 30 minutes. This could be reduced considerably by computing in a compiled language such as C.

2.4 Estimation of X based on IID Observations

If $\boldsymbol{\theta}$ is known, then $\mathrm{E}(\mathbf{X} \mid \mathbf{Y}, \boldsymbol{\theta}, \mathbf{X} > \mathbf{0})$ has minimum mean square prediction error for estimating $\mathbf{X}$. Conditioning on $\mathbf{X} > \mathbf{0}$ safeguards against negative estimates. When $\boldsymbol{\theta}$ is unknown, a natural alternative is

$$\hat{\mathbf{X}} = \mathrm{E}(\mathbf{X} \mid \mathbf{Y}, \boldsymbol{\theta} = \hat{\boldsymbol{\theta}}, \mathbf{X} > \mathbf{0}),$$

where $\hat{\boldsymbol{\theta}}$ is the M.L.E. of $\boldsymbol{\theta}$ based on $\mathbf{Y}$. Because the samples are independent the nth column of $\hat{\mathbf{X}}$ is equal to

$$\hat{\mathbf{x}}_t = \mathrm{E}(\mathbf{x}_t \mid \mathbf{y}_t, \hat{\boldsymbol{\theta}}, \mathbf{x}_t > \mathbf{0}). \tag{8}$$

But $[\mathbf{x}_t \mid \mathbf{y}_t, \hat{\boldsymbol{\theta}}]$ is multivariate normal, and thus computing $\hat{\mathbf{x}}_t$ generally requires multidimensional integration over the positive quadrant. To avoid this, we reason as follows: if the normal approximation for $\mathbf{x}_t$ is appropriate, the positive quadrant will contain nearly all the mass of the distribution of $[\mathbf{x}_t \mid \mathbf{y}_t, \hat{\boldsymbol{\theta}}]$ and thus conditioning on $\mathbf{x}_t > \mathbf{0}$ has little effect. The more crucial matter is to satisfy the constraint $\mathbf{A}\hat{\mathbf{x}}_t = \mathbf{y}_t$, for which an iterative proportional fitting procedure is well suited. It is natural to ask here whether this summation constraint together with the positivity constraint would be adequate to determine $\hat{\mathbf{x}}_t$ without our proposed statistical model. The answer is in general no and we will come back to this point in the validation section 5.

Iterative proportional fitting is an algorithm widely used in the analysis of contingency tables. It adjusts the table to match the known marginal totals. The algorithm and its convergence have been the subject of extensive study (Ireland and Kullback, 1968; Ku and Kullback, 1968; Csiszár, 1975). For a one-router network, the linear constraints $\mathbf{A}\mathbf{x}_t = \mathbf{y}_t$ with positive $\mathbf{x}_t$ can be re-expressed in a contingency table form if we label the table in both directions by the nodes in the network, with the rows as the origin nodes and the columns as destinations. Each entry in the table represents the byte count for an OD pair. In this case, the constraint translates exactly into column and row margin summation constraints. For general networks, $\mathbf{A}\mathbf{x}_t = \mathbf{y}_t$ corresponds to further summation constraints beyond the marginal constraints, but the iterative proportional fitting algorithm can easily deal with these additional constraints in the same way as the marginal constraints. As long as the space $\{\mathbf{x}_t : \mathbf{y}_t = \mathbf{A}\mathbf{x}_t, \mathbf{x}_t \geq \mathbf{0}\}$ is not empty, positivity of the starting point is a sufficient condition for convergence.

To give a positive starting value, we use a component-wise version of (8)

$$\hat{x}_{t,i}^{(0)} = \mathrm{E}(x_{t,i} \mid \mathbf{y}_t, \hat{\boldsymbol{\theta}}, x_{t,i} > 0), \qquad i = 1, \ldots, I \tag{9}$$

and then we adjust the resulting vector $\hat{\mathbf{x}}_t^{(0)} = (\hat{x}_{t,1}^{(0)}, \ldots, \hat{x}_{t,I}^{(0)})$ using an iterative proportional fitting procedure to meet the constraint $\mathbf{A}\hat{\mathbf{x}}_t = \mathbf{y}_t$. The quantities $\hat{x}_{t,i}^{(0)}$ can be computed based on the following easily derived formula for a Gaussian variate $Z \sim \mathrm{normal}(\mu,\ \sigma^2)$: $\mathrm{E}(Z \mid Z > 0) = \mu + \sigma/\sqrt{2\pi}\exp(-\mu^2/(2\sigma^2))\Phi^{-1}(\mu/\sigma)$, where $\Phi(\cdot)$ is the standard normal cumulative distribution. The conditional mean and variance of $[x_{t,i} \mid \mathbf{y}_t, \hat{\boldsymbol{\theta}}]$ can be found using (6).

Our iterative proportional fitting procedure starts with $\hat{\mathbf{x}}_t^{(0)}$ from (9) and then sweeps cyclically through the constraint equations as follows. For each row a_j of $\mathbf{A}$ obtain $\hat{\mathbf{x}}_t^{(j)}$ by multiplying the components of $\hat{\mathbf{x}}_t^{(j)}$ corresponding to nonzero elements of a_j by the factor $y_{t,j}/\mathbf{a}_j\hat{\mathbf{x}}_t^{(j-1)}$. Here superscripts of $\hat{\mathbf{x}}_t$ are interpreted modulo J. Convergence is declared when all constraints are met to a given numerical accuracy.

3 A Moving IID Model for Time Series Data

Recall from Figure 2 that the links exhibit outbursts of traffic interleaved with low activity intervals. Changes in the local average byte counts are obvious and a time-varying approach is needed to model this data.

3.1 A Local iid Model

To allow for parameters that depend on t, the basic iid model is extended to a local iid model using a moving window of a fixed size $w = 2h + 1$, where h is the half-width. For estimation at time t, the window of observations centered at t are treated as iid:

$$\mathbf{y}_{t-h}, \ldots, \mathbf{y}_{t+h} \sim \text{iid normal}(\mathbf{A}\boldsymbol{\lambda}_t,\ \mathbf{A}\boldsymbol{\Sigma}_t\mathbf{A}') \tag{10}$$

where $\boldsymbol{\Sigma}_t = \phi_t \text{diag}(\boldsymbol{\lambda}_t{}^c)$. This is the moving iid version of model (2)–(4) and the methods of the previous section are used to obtain estimates $\boldsymbol{\lambda}_t$ and $\mathbf{x}_t$. Because consecutive windows are overlapping, estimates $\hat{\boldsymbol{\lambda}}_t$ and $\hat{\mathbf{x}}_t$ are implicitly smoothed.

The iid assumption for consecutive $\mathbf{x}_t$ vectors within a window is approximate with respect to both independence and identical distributions. The assumption makes our method simple. By choosing a relatively small window size, the method can effectively adapt to the time-varying nature of the data. A moving window approach formally falls under local likelihood analysis (Hastie and Tibshirani, 1990; Loader, 1999), with a rectangular weight function. Within that framework, equation (10) can be justified as a local likelihood approximation to a time-varying model with observations that are independent over time.

3.2 Exploratory Analysis and Model Checking

Before estimating parameters in the *Router1* network, we do some exploratory analysis to check appropriateness of the various assumptions made about $\mathbf{x}_t$. Since $\mathbf{x}_t$ is not observed, however, these checks must be based only on $\mathbf{y}_t$. In what follows, $\mathbf{m}_t$ and $\mathbf{s}_t$ denote the expected value and *component-wise* standard deviation of $\mathbf{y}_t$.

IID Normality. Using the *switch* link as a representative interface, Figure 3 shows normal probability plot (left) and time series plot (right) for standardized residuals of byte counts originating at *switch*. The vector of residuals for all links at time t is defined as $\hat{\mathbf{e}}_t = (\mathbf{y}_t - \hat{\mathbf{m}}_t)/\hat{\mathbf{s}}_t$ where $\hat{\mathbf{m}}_t$ and $\hat{\mathbf{s}}_t$ are the sample mean and standard deviation over a window of size 11 centered at t. Each element of $\hat{\mathbf{e}}_t$ should be approximately iid normal over time. The probability plot is somewhat concave and this is typical of the other links as well, implying that the actual count distributions have heavier upper tails than the normal. The agreement is sufficient, however, for a normal-based modeling approach to be meaningful.

Independence over time is regarded as an effective assumption and the time-series plot of residuals in the right-hand panel is meant only to check for gross violations. For a small

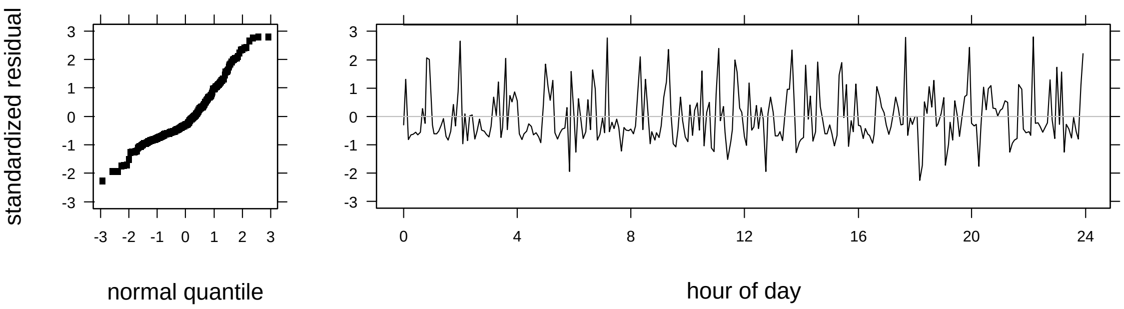

Figure 3: Left: normal probability plots of locally standardized residuals from the *origin switch* link measurements shown in Figure 2. The distribution has a somewhat heavier tail than the normal. Right: time series plots of locally standardized residuals for the same *origin switch* link.

window size, the link counts have more or less constant mean and variance unless the window happens to include an abrupt change for one or more links and in this case we have to live with the consequences of our model being wrong. This point is addressed further in Section 3.3. Large windows including several days, for example, might contain diurnal patterns that would render the iid assumption unreasonable. With $w = 11$ corresponding to a 55 minute window, the local dependence is negligible.

Window size. In Section 3.3 to follow, the window choice is $w = 11$. Windows of 5 and 21 yielded similar final results. It is possible to follow the guidelines in Loader (1999) to choose the window size via cross validation (CV) to minimize the mean square estimation error for $\mathbf{x}_t$ but we have not done this.

Variance-mean relationship. Identifiability requires a relation between the mean and variance of OD byte counts. If the relation is unknown, however, uncovering it remains difficult. In Section 2, we proposed a power relation as a simple extension to the Poisson count model that allows over-dispersion and a super-linear increase in the variance with the mean. We now attempt to check this assumption.

Figure 4 plots $\log \hat{s}^2_{t,j}$ against $\log \hat{m}_{t,j}$ ($j = 1, \ldots, 8$) using the local averages and standard-

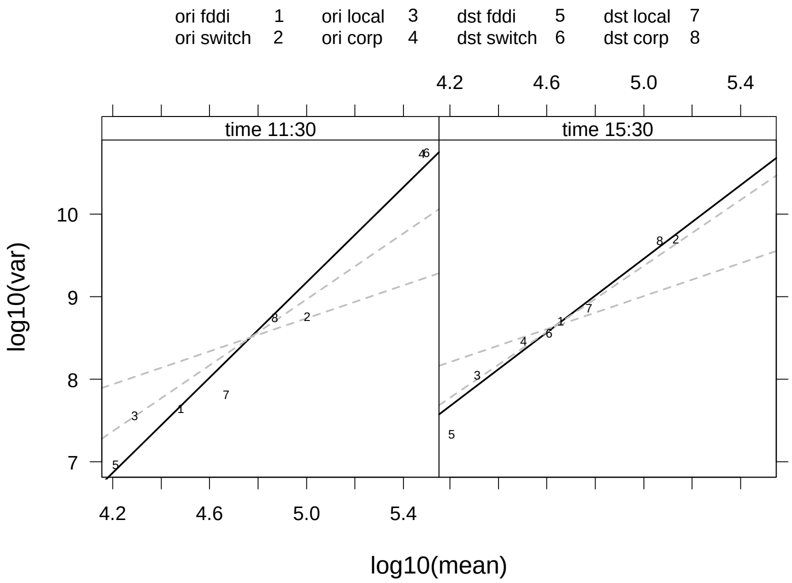

Figure 4: Local variances, $\hat{s}_t^2$, versus local means, $\hat{m}_t$, for *Router1* link measurements. Two representative times are shown. Each panel has 8 points, one for each source and destination link. The solid line is a linear fit to the points while the dashed lines have slopes $c = 1$ and $c = 2$ corresponding to different power law relations. The quadratic power law, $c = 2$, is a better fit.

deviations described above for checking the normal assumption. The two panels represent windows centered at different time periods: t = 11:30 AM and 3:30 PM. Each point in the plot represents a given network link. According to the arguments that follow, a power relation $\boldsymbol{\Sigma}_t = \phi_t \mathrm{diag}(\boldsymbol{\lambda}_t{}^c)$ will tend to produce a linear plot with slope equal to c. Rough linearity is evident in the figure and slopes of the solid fitted lines indicate that a quadratic power law ($c = 2$) is more reasonable than a linear law ($c = 1$). The two panels shown are typical of most cases.

Suppose that only the first K OD pairs contribute significantly to the byte counts on the jth link at time t. Then for $c \geq 1$ and $K \geq 1$,

$$K^{1-c}(\lambda_1 + \ldots + \lambda_K)^c \leq \lambda_1^c + \ldots + \lambda_K^c \leq (\lambda_1 + \ldots + \lambda_K)^c.$$

Thus, the variance $s_{t,j}^2$ and mean $m_{t,j}$ of the byte counts for link j satisfy

$$\log s_{t,j}^2 = \log \phi + c \log m_{t,j} + a_{t,j}$$

where $a_{t,j}$ is bounded between zero and $(1-c)\log K$. If byte counts on most of the link interfaces are dominated by a few OD pairs, then $(1-c)\log K$ is likely to be small in comparison to the variation of $c \log m_{t,j}$ and thus $\log s_{t,j}^2$ will be approximately linear in $\log m_{t,j}$ with a slope c. When $c = 1$, $a_t \equiv 0$ and this linearity holds exactly.

It may be possible to estimate c directly from the data but identifiability for this general case model is not obvious. Therefore we select a value for c from from a limited number of candidates. Since these models all have the same complexity, our selection procedure compares the maximum log-likelihood for each window and selects the one that on average gives the largest maximum log-likelihood.

3.3 Local Fits to *Router1* Data

Estimates of $\boldsymbol{\lambda}$. We fit models with $c = 1$ and $c = 2$ using a moving window of width $w = 11$. Comparing the fits, 98% of windows give larger likelihood for $c = 2$ even though the difference in fit is typically small.

Figure 5 plots estimates of $\boldsymbol{\lambda}$ (gray) for the *Router1* network. For comparison, the top and right marginal panels also show 11-point moving averages of the observed byte counts $\bar{\mathbf{y}}_t$ in black. Obviously, these moving averages do not rely on a relation between mean and variance, but they are, nevertheless, in general agreement with the model-based estimates. If the match was poor, we would question the power-law specification. The model-based margins are more variable the moving averages. This is most obvious at the high peaks in Figure 5 but magnifying the vertical scale by a factor of 20 (not shown) would demonstrate that the difference in smoothness holds at smaller scales as well. Discussion of estimates $\hat{\mathbf{x}}_t$ is deferred to Section 5 after the modeling refinements of Section 4.

Difficulties with strictly local fitting. An intrinsic problem with fitting a large number of parameters using a small moving window is that the likelihood surface can have multiple local maxima and can be poorly conditioned in directions where the data contains little information on the parameters. This can lead to estimation inaccuracies and numerical difficulties. Some of the smaller scale variations of $\hat{\boldsymbol{\lambda}}_t$ in Figure 5 are likely due to poor conditioning.

An example of a numerical problem comes when the iterative procedure for estimating $\boldsymbol{\lambda}_t$ converges to a boundary point in which some components are zero. Since estimation is

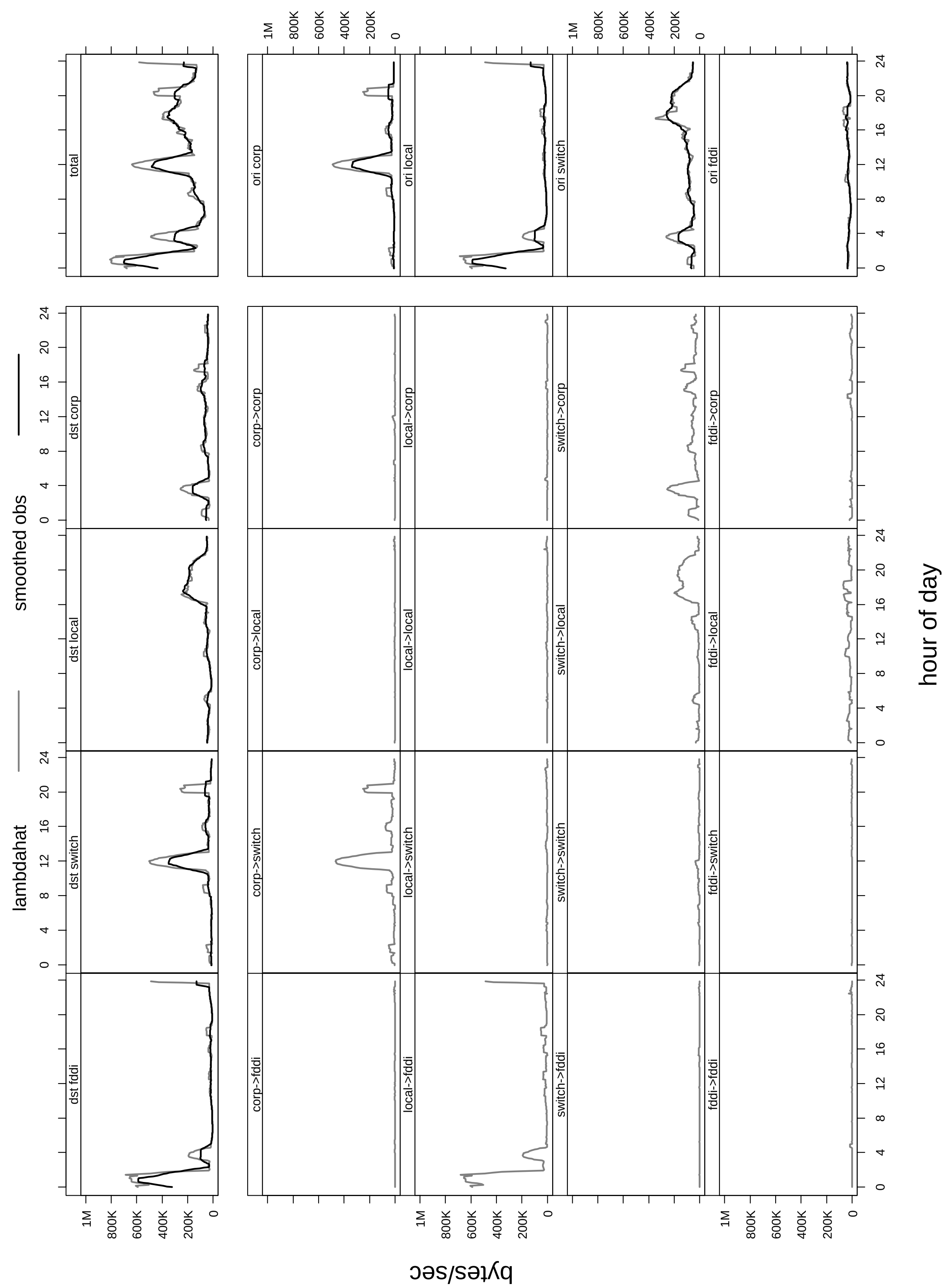

Figure 5: Mean traffic estimates, $\hat{\boldsymbol{\lambda}}_t$, for all OD pairs are plotted against time in the large 4x4 block of panels. Marginal panels on the top and right compare model-based estimates of the mean link traffic $\hat{E}(\mathbf{x}_t) = \mathbf{A}\hat{\boldsymbol{\lambda}}_t$ (gray) with moving averages of the observed link measurements (black).

done in terms of $\log \boldsymbol{\lambda}_t$, the likelihood surface is flat in directions corresponding to zeros. Additional singularity arises when the number of nonzero components becomes smaller than the number of independent link interfaces.

The following section presents a refinement to the local model with the intent of overcoming numerical problems and, more importantly, improving estimation accuracy in parameter directions that are poorly determined from small data windows. Our approach encourages smoother parameter estimates by borrowing information from neighboring data windows through the use of a prior derived from previous estimates. The refinement is especially useful for large networks with many more parameters than observed link interfaces.

4 Additional Smoothing by Modeling $\boldsymbol{\lambda}_t$

Let $\boldsymbol{\eta}_t = (\log(\boldsymbol{\lambda}_t), \log(\phi_t))$ be the log of the parameter time series. We model $\boldsymbol{\eta}_t$ as a multi-dimensional random walk:

$$\boldsymbol{\eta}_t = \boldsymbol{\eta}_{t-1} + \mathbf{v}_t, \qquad \mathbf{v}_t \sim \text{normal}(\mathbf{0},\ \mathbf{V}), \tag{11}$$

where $\mathbf{V}$ is a fixed variance matrix chosen beforehand. Equations (11) and (10) compose a state-space model for $\mathbf{Y}_t = (\mathbf{y}_{t-h}, \ldots, \mathbf{y}_t, \ldots, \mathbf{y}_{t+h})$. The choice of $\mathbf{V}$ is important because it determines how much information from previous observations carries over to time t. Let $\tilde{\mathbf{Y}}_t = (\mathbf{y}_1, \ldots, \mathbf{y}_{t+h})$ be all the observations up to time $t+h$, then

$$p(\boldsymbol{\eta}_t \mid \tilde{\mathbf{Y}}_t) = p(\boldsymbol{\eta}_t \mid \tilde{\mathbf{Y}}_{t-1}, \mathbf{Y}_t) \ \propto\ p(\boldsymbol{\eta}_t \mid \tilde{\mathbf{Y}}_{t-1}) p(\mathbf{Y}_t \mid \boldsymbol{\eta}_t).$$

Thus, maximizing the posterior $p(\boldsymbol{\eta}_t \mid \tilde{\mathbf{Y}}_t)$ is equivalent to maximizing log-likelihood with an additive penalty corresponding to the adaptive log-prior on $\boldsymbol{\eta}_t$ conditioned on past data. The penalty term conditions nearly flat directions on the original log-likelihood surface which can otherwise produce poor estimates of $\boldsymbol{\lambda}$.

For the prior on $\boldsymbol{\eta}_t$ we have

$$p(\boldsymbol{\eta}_t \mid \tilde{\mathbf{Y}}_{t-1}) \ \propto\ \int p(\boldsymbol{\eta}_{t-1} \mid \tilde{\mathbf{Y}}_{t-1}) p(\boldsymbol{\eta}_t \mid \boldsymbol{\eta}_{t-1}) d\boldsymbol{\eta}_{t-1}.$$

but to relieve computational burden, we approximate the posterior $p(\boldsymbol{\eta}_{t-1} \mid \tilde{\mathbf{Y}}_{t-1})$ at $t-1$ by $\text{normal}(\hat{\boldsymbol{\eta}}_{t-1},\ \hat{\boldsymbol{\Sigma}}_{t-1})$, where $\hat{\boldsymbol{\eta}}_{t-1}$ is the posterior mode and $\hat{\boldsymbol{\Sigma}}_{t-1}$ is the inverse of the curvature of the log posterior density at the mode (Gelman *et al.*, 1995). Hence, $p(\boldsymbol{\eta}_t \mid \tilde{\mathbf{Y}}_{t-1})$ can be approximated by normal$(\hat{\boldsymbol{\eta}}_{t-1},\ \hat{\boldsymbol{\Sigma}}_{t-1} + \mathbf{V})$. With this prior, optimization of $p(\boldsymbol{\eta}_t \mid \tilde{\mathbf{Y}}_t)$ can be handled in a manner similar to that outlined in Section 2.3. For the EM algorithm, the Q function is modified by including an extra constant term from the prior and hence only the M step has to be changed. If the EM algorithm is used only to provide a good starting point for a second order method, it may not even be necessary to modify the M step. To use a second order optimization routine (like `ms` in S), all the derivative calculations are modified by including the derivatives of the log-prior, $\log(p(\boldsymbol{\eta}_t \mid \tilde{\mathbf{Y}}_{t-1}))$, and hence can be carried out just as easily as before. As an end result, using a normal approximation to the adaptive prior adds almost no additional computational burden for parameter estimation. The final algorithm is as follows.

1. Initialize $t = h + 1, \hat{\boldsymbol{\eta}}_{t-1} = \boldsymbol{\eta}_0$ and $\hat{\boldsymbol{\Sigma}}_{t-1} = \boldsymbol{\Sigma}_0$.
2. Let $\hat{\boldsymbol{\Sigma}}_{t|t-1} = \hat{\boldsymbol{\Sigma}}_{t-1} + \mathbf{V}$ and $\pi(\boldsymbol{\eta}_t) = p(\boldsymbol{\eta}_t \mid \tilde{\mathbf{Y}}_{t-1}) \sim$ normal$(\hat{\boldsymbol{\eta}}_{t-1},\ \hat{\boldsymbol{\Sigma}}_{t|t-1})$ be the prior of $\boldsymbol{\eta}_t$ given observations $\tilde{\mathbf{Y}}_{t-1}$.
3. Let $g(\boldsymbol{\eta}_t) = \log \pi(\boldsymbol{\eta}_t) + \log p(\mathbf{Y}_t \mid \boldsymbol{\eta}_t)$ be the log of the posterior distribution of $\boldsymbol{\eta}_t$. Find the mode $\hat{\boldsymbol{\eta}}_t = \text{argmax } g(\boldsymbol{\eta}_t)$ using optimization method. Note that derivatives of g are $\dot{\mathbf{g}}(\boldsymbol{\eta}_t) = -\hat{\boldsymbol{\Sigma}}_{t|t-1}^{-1}(\boldsymbol{\eta}_t - \hat{\boldsymbol{\eta}}_{t-1}) + \partial \log p / \partial \boldsymbol{\eta}_t$, and $\ddot{\mathbf{g}}(\boldsymbol{\eta}_t) = -\hat{\boldsymbol{\Sigma}}_{t|t-1}^{-1} + \partial^2 \log p / \partial {\boldsymbol{\eta}_t}^2$.
4. Let $\hat{\boldsymbol{\Sigma}}_t = \ddot{\mathbf{g}}(\hat{\boldsymbol{\eta}}_t)^{-1}, t = t + 1$ and return to Step 2.

The proposed model and algorithm are similar to Kalman Filtering (Anderson and Moore, 1979) and Bayesian dynamic models (West and Harrison, 1997). The effect of the adaptive prior $\pi(\boldsymbol{\eta}_t)$ on estimation and hence the smoothness is controlled by the size of $\mathbf{V}$. If a relatively large $\mathbf{V}$ is chosen, $\pi(\boldsymbol{\eta}_t)$ only plays a secondary role. In comparison, the choice of $\boldsymbol{\eta}_0$, $\boldsymbol{\Sigma}_0$ is less important in the sense that their effect dies out with time. In our implementation, both $\mathbf{V}$ and $\boldsymbol{\eta}_0$ where set empirically from preliminary parameter estimates obtained in Section 2, and a large $\boldsymbol{\Sigma}_0$ is chosen to reflect our poor prior knowledge at the start of estimation.

Figure 6 shows estimates of $\boldsymbol{\lambda}_t$ using the adaptive prior (black) and compares them to previous estimates (gray) with no prior. The vertical scale is magnified 20 times over that of Figure 5. As desired, the prior-based estimates are clearly smoother than those from strict local fitting. Moreover, in the upper and right marginal panels, the marginal sums of the new estimates (black) are less variable than those of the old (gray).

5 Validation Against Actual OD Counts

The ultimate goal is to estimate the actual time-varying traffic, $\mathbf{x}_t$, and assess the accuracy of these estimates as well as the fit of the model. But standard residual analyses are not available because $\mathbf{x}_t$ is unobservable and the fitting procedure provides an exact fit to the link observations: $\mathbf{y}_t - \mathbf{A}\hat{\mathbf{x}}_t = \mathbf{0}$. Thus, to validate the modeling approach, we have instrumented the *Router1* network to directly measure complete OD byte counts $\mathbf{x}_t$ and not merely link measurements $\mathbf{y}_t$. This section compares estimates $\hat{\mathbf{x}}_t$ with actual OD traffic.

Measuring OD traffic on a LAN generally requires special hardware and software. *Router1* is a Cisco 7500 router capable of generating data records of IP (internet protocol) flows using an export format called *netflow*. These records were sent to a nearby workstation running *cflowd* software (CAIDA, 1999) that builds a database of summary information in real time. Finally, we ran aggregation queries on the flow database to calculate OD traffic matrices for the *Router1* network. Queries were run automatically at approximate 5 minute intervals in an effort to match with the routine link measurements studied in previous sections.

In principle, marginal sums from the *netflow* data should match the link measurements over identical time intervals. Actual measurements inevitably have discrepancies due to timing variations and slight differences in how bytes are accumulated by the two measurement methods. Let $\mathbf{x}_t$ represent OD traffic measured from *netflow* records for the 5 minute interval ending at time t. Link measurements corresponding to $\mathbf{x}_t$ are calculated as $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t$ and then these are used to form estimates, $\hat{\mathbf{x}}_t$, of the OD traffic. Fitted values in this section are based on the adaptive prior model of Section 4.

Full scale plots (not shown) show excellent agreement between actual $\mathbf{x}_t$ and predicted $\hat{\mathbf{x}}_t$ OD traffic. Large OD flows, corresponding to periods of high usage, are estimated with very small relative error. A more critical view of the estimates is seen in Figure 7 where the vertical axis is magnified by a factor of 20 so that the large well-estimated peaks are off-scale. The figure focuses on estimation errors for the smaller-scale features. In particular,

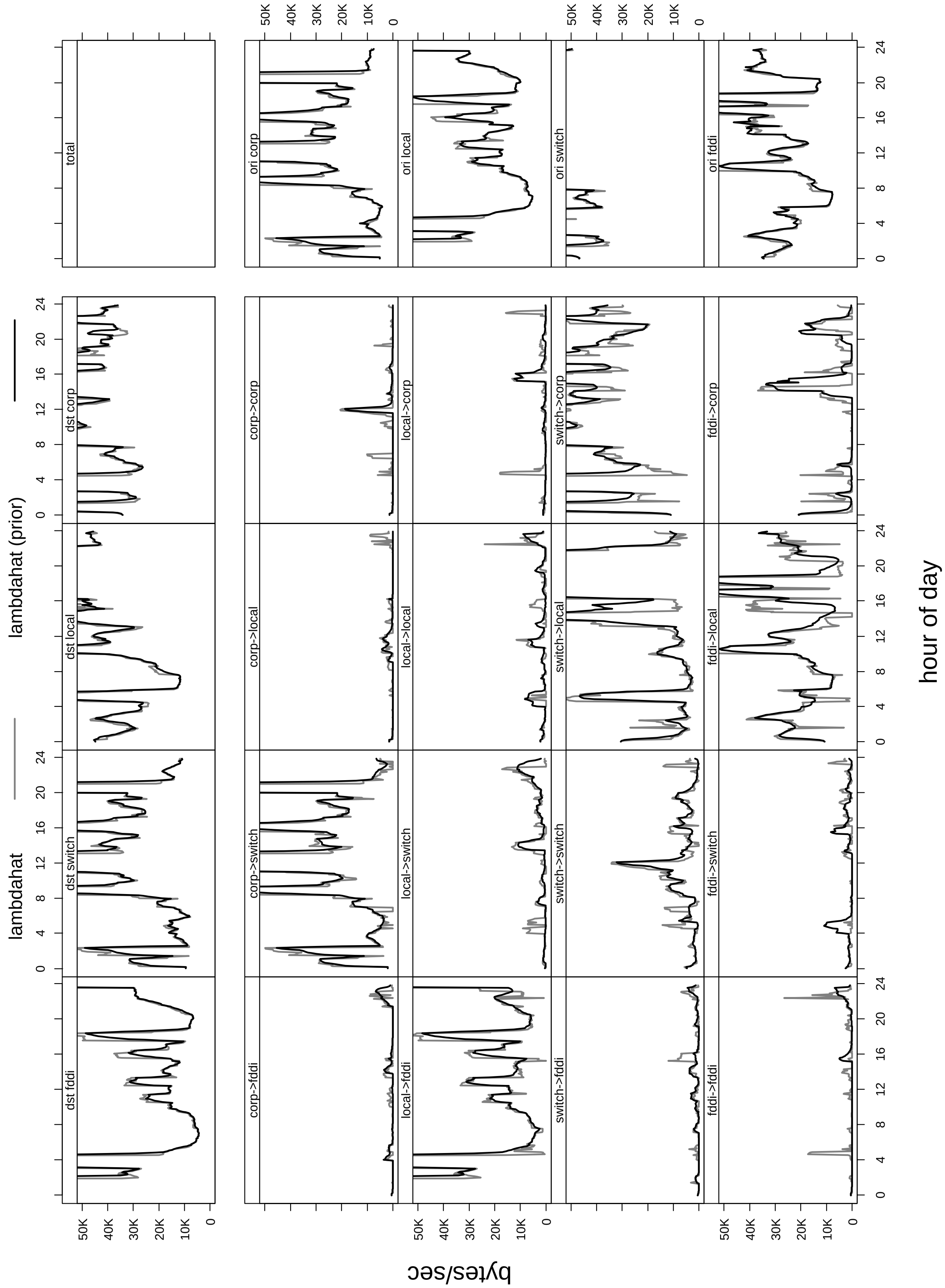

Figure 6: Comparison of mean OD traffic estimates $\hat{\boldsymbol{\lambda}}_t$ obtained from the original local model (gray) and the refined model (black) that incorporates an adaptive prior. Estimates from the refined model are smoother, especially near zero.

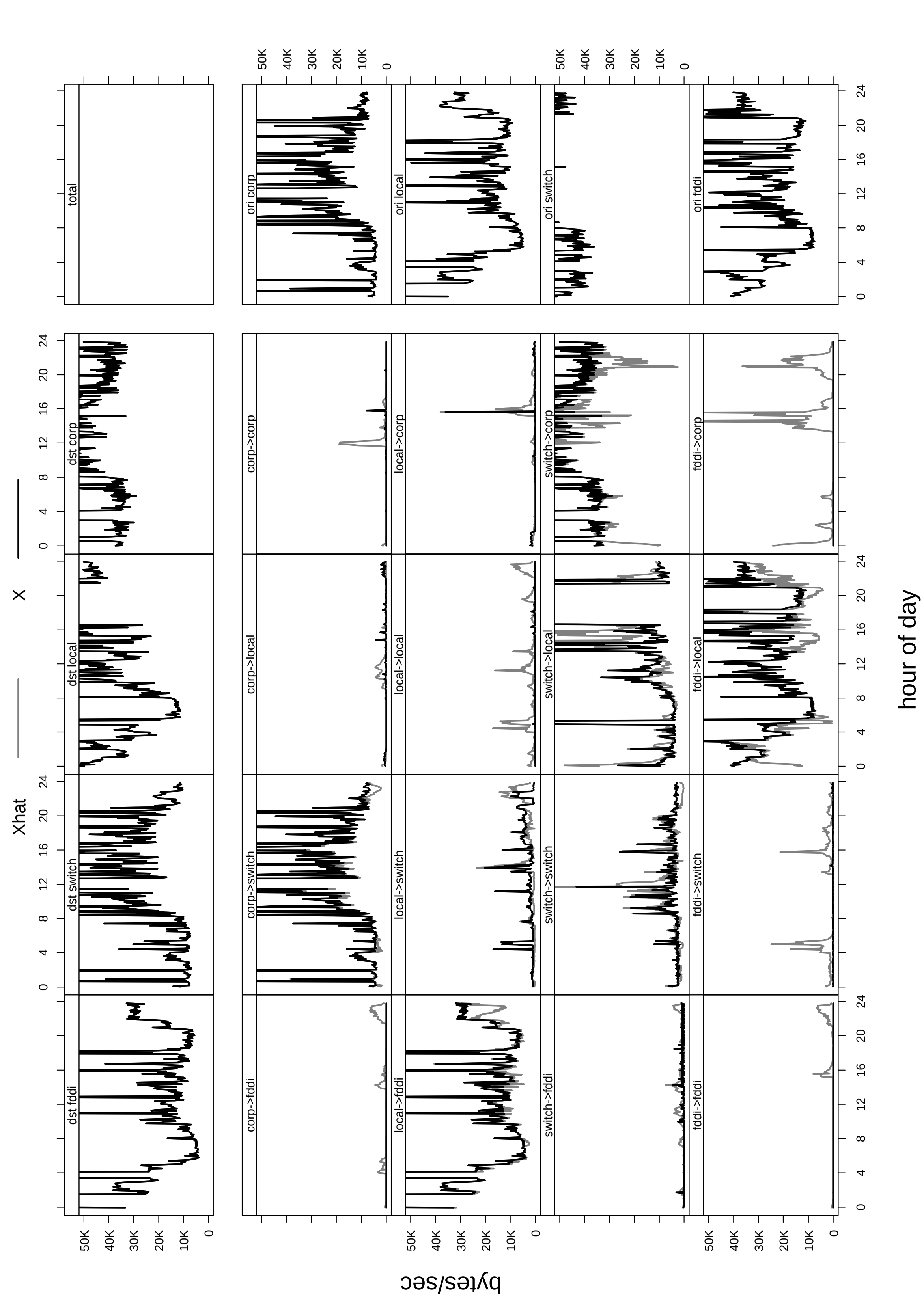

Figure 7: Validation measurements, $\mathbf{x}_t$ (black), plotted atop estimates, $\hat{\mathbf{x}}_t$ (gray). The resolution on the vertical scale highlights artifacts of the estimation procedure that occur especially when the actual traffic is near zero. The panel for $fddi \rightarrow corp$ in the lower right is particularly bad. The marginal plots of $\mathbf{x}_t$ and $\hat{\mathbf{x}}_t$ match exactly as required by the fitting algorithm.

we sometimes predict a substantial amount of traffic when the actual amount is zero. This is especially apparent in the lower right panel labeled *fddi* $\rightarrow$ *corp* where the traffic is greatly overestimated (relative to the actual low traffic). Due to the constraint that fitted margins must match the data, overestimates in one panel are compensated by underestimates in another. This produces a checkerboard pattern of estimation errors in which positive errors are balanced by negative errors in each row and column. Perhaps additional smoothing from the prior would be useful. We have considered setting the prior variances and local window width using a cross-validation approach but we have not pursued this possibility in depth.

Although estimation errors are sometimes large, the model-based estimates perform well when compared to the range of all possible OD estimates that are both nonnegative and consistent with the observed link data. As an example, at 3:30 AM we compute actual estimation errors for each of the 16 OD pairs and divide by the range of all possible estimates. Nine of these ratios are less than 0.14% and all are less than 8%. By this measure, the statistical model is contributing a tremendous amount to the estimation accuracy.

6 Beyond One-Router Networks

Figure 8 shows a two-router portion of the larger Lucent network depicted in Figure 1. The routers, labeled *Router4* and *Gateway*, each support four edge nodes and one internal link. This gives a total of 20 one-way link interfaces. *Router4* serves one organization within Lucent and *Gateway* is the router with a connection to the public Internet. The edge nodes represent sub-networks. Applying our methods to this somewhat more complex topology demonstrates that the approach is computationally feasible, though not fully scalable to larger problems. Experience with this network also motivates some directions for future work.

Among the 20 link count measurements there are, in principle, four linear dependencies corresponding to the balance between incoming and outgoing byte counts around each router and to the fact that traffic measurements between *Router4* and *gateway* are collected at interfaces on both routers. In reality there are inevitable discrepancies in the counts—up to 5%, for example, between total incoming and total outgoing traffic at these routers. We

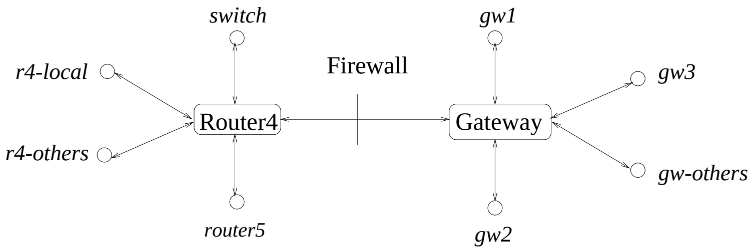

Figure 8: Two-Router Network at Lucent Technologies

reconcile these inconsistencies before estimating traffic parameters.

Suppose there are n redundancy equations arising from the network topology:

$$\mathbf{a}_i'\mathbf{y}_t = \mathbf{b}_i'\mathbf{y}_t, \quad i = 1, \ldots, n,$$

where $\mathbf{a}_i, \mathbf{b}_i$ are 0-1 vectors with $\mathbf{a}_i'\mathbf{b}_i = 0$. When measurements do not provide exact equality, we re-scale them using the following iterative proportional adjustment method. Without loss of generality, assume $\mathbf{b}_i'\mathbf{y}_t \geq \mathbf{a}_i'\mathbf{y}_t$. Moving cyclically through $i = 1, \ldots, n$, multiply the components of $\mathbf{y}_t$ corresponding to nonzero elements of $\mathbf{a}_i$ by the factor $(\mathbf{b}_i'\mathbf{y}_t)/(\mathbf{a}_i'\mathbf{y}_t)$. Iteration stops when $(\mathbf{b}_i - \mathbf{a}_i)'\mathbf{y}_t$ is sufficiently close to zero. Such reconciliation makes adjustments proportional to the actual link measurements, which is an intuitively fair approach. Following adjustment, we remove redundant elements from both $\mathbf{y}_t$ and $\mathbf{A}$ and then locally fit the iid normal model obtaining estimates of both $\boldsymbol{\lambda}_t$ and $\mathbf{x}_t$.

Estimates $\hat{\boldsymbol{\lambda}}_t$ for a recent one day period are shown in Figure 9 on a scale of 0 to 100K bytes/sec which clips off some of the highest peaks in the data. Qualitatively, we expected to see lower traffic levels in the upper-left and especially lower-right blocks of panels corresponding to OD pairs on different routers. We also anticipated that only a few pairs would usually dominate the data flow. This tends to make the estimation problem easier because simultaneous peaks in a single origin and single destination byte count can be attributed to a single OD pair with relatively little ambiguity.

Computation time for this network with 64 OD pairs was 35 sec per estimation window which compares to 6 sec for the *Router1* network with 16 OD pairs. In general, the number OD pairs can grow as fast as the square of the number of links.

Scaling these methods to work for larger problems is an area of current research. Our identifiability result implies that for a given OD pair, byte counts from only the two edge-nodes are needed to estimate the intensity of traffic between them. All of the other byte counts provide supplementary information. This suggests handling large networks as a series of subproblems, each involving a given subset of edge nodes and simplifying the rest of the network by forming aggregate nodes. The two-router example of Figure 8 could represent one such subproblem for analyzing the complete Lucent shown network of Figure 1(a). We are presently studying different schemes for dividing a large problem into manageable pieces without sacrificing too much estimation accuracy.

7 Conclusions

Practical realities dictate that information needed for managing computer networks is sometimes best obtained through estimation. This is true even though exact measurements could be made by deploying specialized hardware and software. We have considered inference of origin-destination byte counts from measurements of byte counts on network links such as can be obtained from router interfaces. All commercial routers can report their link counts whereas measuring complete OD counts on a network is far from routine.

Using a real-life example, we have shown that OD counts can be recovered with good accuracy relative to the degree of ambiguity that remains after marginal and positivity constraints are met. We model the counts locally as iid normal conditional on the mean. Identifiability of the OD parameters from link data requires a relation between means and variances of OD counts. A power-law family provides a model which is a natural extension of the normal approximation to the Poisson and incorporates extra-Poisson variation observed in sample data.

Simple local likelihood fitting of an iid model is not sufficient because large fitting windows over-smooth sharp changes in OD traffic, while a small windows cause estimates to be unreliable. A refinement in which the logs of positive parameters are modeled as random walks, penalizes the local likelihood surface enough to induce smoothness in parameter estimates while not unduly compromising their ability to conform to sharp changes in traffic. Section 4 laid out a fully normal approximation to this approach and demonstrated how

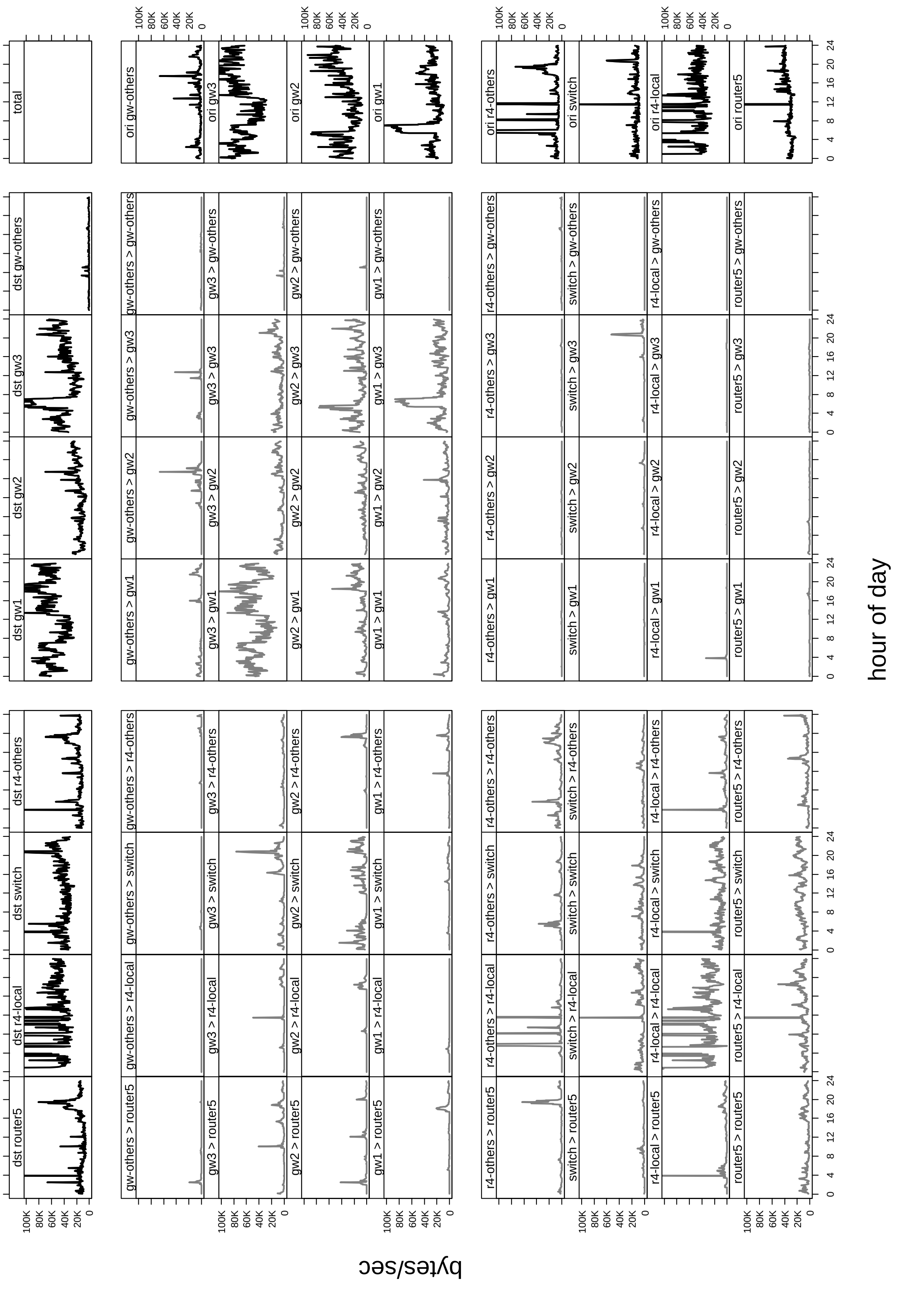

Figure 9: Estimates of the traffic, $\hat{\mathbf{x}}_t$, for all OD pairs in the two-router network around *Router4* and *Gateway* illustrated in Figure 8. The largest spikes have been clipped by the vertical scale. Major features of the estimates are consistent with usage patterns known to system administrators.

effectively it recovers OD byte counts for our *Router1* network.

There are several possible directions for further work. First and most important is developing an efficient algorithm for large size networks, as we discussed briefly in Section 6. Second, we would like to study how accuracy of traffic estimates for different OD pairs is affected by the network topology. Third, we would like to fold measurement errors into the data model rather than reconciling them up front and then proceeding as if there were no such errors. Additional topics are cross-validation for window sizes and prior variances, modeling changes in $\hat{\boldsymbol{\lambda}}_t$ with more realism than the random walk, and replacing the normal distribution with a heavy-tailed positive distribution.

Progress in these areas would be especially helpful for analyzing large networks but the basic approach we have outlined is still appropriate. Our methods can easily be used on portions of networks with a small-to-moderated number of edge nodes. Giving LAN administrators access to OD traffic estimates provides them with a much more direct understanding of the sizes and timing of traffic flows through their networks. This is an enormous help for network management, planning and pricing.

Acknowledgments

We thank Tom Limoncelli and Lookman Fazal for system administration support in setting up MRTG and Netflow data collection. We also thank Debasis Mitra for pointing us to this problem and Yehuda Vardi and Mor Armony for engaging discussions.

A Appendix: Proof of Identifiability

A.1 Proof of Theorem 2.1

Proof. From the basic model (2)–(4), if it is easy to see that given two parameter sets $(\boldsymbol{\lambda}, \phi)$ and $(\tilde{\boldsymbol{\lambda}}, \tilde{\phi})$, the model is identifiable if and only if the conditions

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{A}\tilde{\boldsymbol{\lambda}}, \qquad \phi\mathbf{A}\ \mathrm{diag}(\boldsymbol{\lambda}^c)\mathbf{A}' = \tilde{\phi}\mathbf{A}\ \mathrm{diag}(\tilde{\boldsymbol{\lambda}}^c)\mathbf{A}' \tag{12}$$

imply the equalities

$$\boldsymbol{\lambda} = \tilde{\boldsymbol{\lambda}}, \qquad \phi = \tilde{\phi}.$$

Note that $\mathbf{B}$ has full column rank if and only if $\mathbf{A}\ \mathrm{diag}(\mathbf{x})\mathbf{A}' = \mathbf{0}$ has only the solution $\mathbf{x} = \mathbf{0}$. Thus if $\mathbf{B}$ has full column rank, the second condition of (12) implies $\phi\boldsymbol{\lambda}^c = \tilde{\phi}\tilde{\boldsymbol{\lambda}}^c$, or equivalently, $a\boldsymbol{\lambda} = \tilde{\boldsymbol{\lambda}}$ where $a^c = \phi/\tilde{\phi} > 0$. Substituting this into the first condition of (12) gives

$$\mathbf{A}\boldsymbol{\lambda} - \mathbf{A}\tilde{\boldsymbol{\lambda}} = (1-a)A\boldsymbol{\lambda} = \mathbf{0}.$$

Let $\mathbf{1}$ be a row vector of ones with length J and note that $\mathbf{1}'\mathbf{A}$ gives the column sums of $\mathbf{A}$. Pre-multiply on both sides of the above equation by $\mathbf{1}$ to give $(1-a)(\mathbf{1}'\mathbf{A})\boldsymbol{\lambda} = 0$. If $\mathbf{B}$ has full column rank, it follows that the column sums of $\mathbf{A}$ are always positive. Therefore if $\lambda_i \geq 0$ with at least one strict inequality, we have $a = 1$ implying both $\boldsymbol{\lambda} = \tilde{\boldsymbol{\lambda}}$ and $\phi = \tilde{\phi}$. On the other hand, it is easy to see that if $\mathbf{B}$ doesn't have full column rank, condition (12) doesn't uniquely determine λ when $\phi = \tilde{\phi}$.

□

A.2 Proof of Corollary 2.2

Proof. We shall show that the matrix $\mathbf{B}$ has full column rank in this case. Note that $\mathbf{A}$ is a $J \times I$ matrix of zeros and ones whose rows represent how the link variables, y, are summed up from the OD variables x. For the ith OD pair, we focus on two link interfaces—one corresponds to the first that interface the traffic from the origin enters the network and the other to the last interface before the traffic leaves the network to the destination. Suppose these two links correspond to two rows r_1 and r_2 of $\mathbf{A}$. Multiplying them component-wise, produces a row vector of $\mathbf{B}$. Moreover, row r_1 has zero entries everywhere except for the OD pairs whose origin matches that of the ith OD pair. Similarly r_2 has zero entries everywhere except for the OD pairs whose destination matches that of the ith interface. It follows that the only component where they share a 1 is the ith. Hence the product vector has entries 0 everywhere except for the ith component where it has entry 1. Thus the rows of $\mathbf{B}$ from such chosen component-wise products form an identity matrix with rank I and $\mathbf{B}$ has I columns. Hence $\mathbf{B}$ is full column rank. □

References

[1] B. D. O. Anderson and J. B. Moore. *Optimal Filtering.* Prentice-Hall, 1979.

[2] J. M. Chambers and T. J. Hastie, editors. *Statistical Models in S.* Chapman & Hall, 1993.

[3] I. Csiszár. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 1975.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihhod from incomplete data via the EM algorithm, with discussion. *Journal of Royal Statistical Society, Series B*, 1977.

[5] The Cooperative Association for Internet Data Analysis. *Cflowd Flow Analysis Software (version 2.0).* Author, 1999. www.caida.org/Tools/Cflowd.

[6] D. M. Gay. Algorithm 611: Subroutines for unconstrained minimization using a model/trust-region approach. *ACM Trans. Math. Software*, 1983.

[7] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis.* Chapman & Hall, 1995.

[8] T. J. Hastie and R. J. Tibshirani. *Generalized Additive Models.* Chapman & Hall, London, 1990.

[9] C. T. Ireland and S. Kullback. Contingency tables with given marginals. *Biometrika*, 1968.

[10] H. H. Ku and S. Kullback. Interaction in multidimensional contingency tables: an information theoretic approach. *J. Res. Nat. Bur. Standards*, 1968.

[11] K. Lange. A gradient algorithm locally equivalent to the EM algorithm. *Journal of the American Statistical Association*, 1995.

[12] C. Loader. *Local Regression and Likelihood.* Springer-Verlag, New York, 1999.

[13] C. Tebaldi and M. West. Bayesian inference on network traffic using link count data. *Journal of the American Statistical Association*, 1998.

[14] R. J. Vanderbei and J. Iannone. An em approach to OD matrix estimation. *Technical Report SOR 94-04, Princeton University*, 1994.

[15] Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the American Statistical Association*, 1996.

[16] M. West and J. Harrison. *Bayesian Forecasting and Dynamic Models*. Springer-Verlag, New York, 1997.