Maximum Pseudo Likelihood Estimation in Network Tomography

Gang Liang Bin Yu

Abstract

Network monitoring and diagnosis are key to improving network performance. The difficulties of performance monitoring lie in today's fast growing Internet, accompanied by increasingly heterogeneous and unregulated structures. Moreover, these tasks become even harder since one cannot rely on the collaboration of individual routers and servers to directly measure network traffic. Even though the aggregatory nature of possible network measurements gives rise to inverse problems, existing methods for solving inverse problems are usually computationally intractable or statistically inefficient.

In this paper, a pseudo likelihood approach is proposed to solve a group of network tomography problems. The basic idea of pseudo likelihood is to ignore the global dependences and focus on simpler association structures instead, which, in turn, keeps a good balance between the computational complexity and the statistical efficiency of the estimation. Some statistical properties of the pseudo likelihood estimator, such as consistency and asymptotic normality, are established. A pseudo expectation-maximization (EM) algorithm is developped to maximize the pseudo log-likelihood function. Two examples with simulated or real data are used to illustrate the pseudo likelihood proposal: (1) inference of the internal link delay distributions through multicast end-to-end measurements; (2) origin-destination matrix estimation through link traffic counts.

Index Terms

Manuscript received August 30, 2002. This work was supported in part by NSF Grants FD98-02314, DMS-9803063, FD01-12731 and ARO grants DAAG55-98-1-0341.

Gang Liang is with the Department of Statistics, University of California at Berkeley, 367 Evans Hall, Berkeley, CA 94720 USA. (phone: 510-524-0725; fax: 510-642-7892; email: liang@stat.Berkeley.EDU).

Bin Yu is with the Department of Statistics, University of California at Berkeley, 367 Evans Hall, Berkeley, CA 94720 USA. (email: binyu@stat.Berkeley.EDU).

End-to-end measurement, multicast tree, network tomography, origin-destination matrix, pseudo likelihood.

I. Introduction

Likelihood is central to most statistical inference problems. But, for some applications, the full likelihood function is difficult or impossible to compute directly. Several modifications of the likelihood method have been proposed and they include pseudo likelihood [2], [3] in Markov random fields (MRF) by Besag (1974), partial likelihood [8] in life table and hazards regression by Cox (1973), and quasi-maximum likelihood [18] in finance by White (1994).

With today's fast growing Internet, network monitoring and inference need to deal with a large number of network performance parameters, such as individual link loss rates and packet delay. Usually one cannot rely on the collaboration of individual routers and servers to directly measure network traffic; estimation of the performance parameters can only be based upon measurements made at a limited subset of computers. *Network Tomography* was first coined by Vardi (1996) to illustrate the similarities between the network inference and medical tomography. In order to harness such challenging tasks, the simplest possible model is adopted and intricate details regarding network transportation are ignored. But even with this, the full likelihood method is still computationally infeasible or time consuming for most network tomography problems.

In this paper, for a group of network tomography problems, a unified pseudo likelihood method is proposed. Our method is partly motivated by the pseudo likelihood method used in solving MRF problems by Besag: both pseudo likelihood functions are constructed by focusing on smaller and simpler dependence structures between variables of interest instead of the globally complex dependence. Sub-problems are formed by considering variables involved in such simpler dependence structures. Sub-problems usually are dependent, but ignoring such dependencies allows for obtaining a pseudo likelihood function. The key difference between Besag's method and ours is how to form sub-problems. Besag's pseudo likelihood is based on the neighborhood decomposition. The problem of inferencing internal link delay distributions through multicast end-to-end traffic can be viewed as an MRF problem. But in the graphical representation of this MRF, all nodes are fully connected. Therefore the neighborhood decomposition scheme is no longer valid here and further dependence structure simplification is necessary.

This paper is organized as follows. In Sec. II, we introduce a general model for network tomography problems and two examples are used to illustrate the general model: 1) inference of the internal link delay distributions through multicast end-to-end measurements; 2) origin-destination matrix inference through link traffic counts. Then, in Sec. III, a pseudo likelihood approach is proposed for the given network tomography model. Finally, in Sec. IV, we apply the pseudo likelihood approach to the above two examples. The Appendix contains all the proofs of the theorems.

II. Model and Framework

Fig. 1 illustrates a general network topology, in which a *node* represents a computer or a subnet (a collection of computers). A connection between any two nodes in the network is called a *path*, which may consist of several *links* — direct connections between two nodes without intermediate nodes. A *packet* is a unit of data of bits. Information is exchanged by sending packets along a path from a source node to destination node(s).

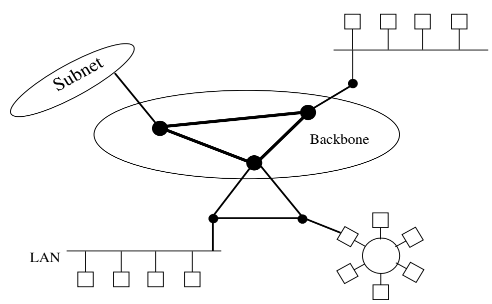


Fig. 1. An illustration of Internet topology

Let $\mathbf{X} = (X_1, ..., X_J)'$ be a J dimensional random variable, which reflects the network dynamics of interest, e.g., link delay, traffic flow counts at a particular time interval. Let $\mathbf{Y} = (Y_1, ..., Y_I)'$ be an I dimensional measurement vector. Generally, there is a linear relationship between observable $\mathbf{Y}$ and unobservable $\mathbf{X}$. As in Coates *et al* [6], such network tomography problems can be approximately (or exactly) represented by the linear equation:

$$\mathbf{Y} = A\mathbf{X}, \tag{1}$$

where A is the $I \times J$ routing matrix, determined by the network topology and the routing table at each router in the network. In this paper, we restrict ourselves to a fixed routing scheme and ignore the possibility of the dynamic routing. Thus A is a 0-1 matrix. It is worth noting that we assume that there is no measurement error or any other errors in (1) in order to further simplify the model.

Equation (1) reveals the aggregatory nature of network measurements, and so the estimation of the distribution of $\mathbf{X}$ is an inverse problem. But in a general network tomography scenario, A is not a full rank square matrix, where typically $I \ll J$. Hence some constraints have to be introduced to ensure the identifiability of the model.

Another key assumption of the general network tomography model is that all components of $\mathbf{X}$ are independent of each other. Such an assumption does not hold strictly in a real network due to the temporal and spatial correlations between network traffic, but it is a good first step approximation. Assume

$$X_j \sim f_j(\theta_j),\ j = 1, ..., J. \tag{2}$$

where f_j is a density function with respect to Lebesgue measure $\mu_{\mathcal{R}}$, and θ_j is its corresponding parameter. Then the parameter of the whole model is $\theta = (\theta_1, ..., \theta_J)$.

Throughout the paper, let $y_1, ..., y_T$ be the observed data vectors at T consecutive time points or intervals: they are assumed to be independent. Nonstationarity can be dealt with as in Cao *et al* by using iid block models. Let $x_1, ..., x_T$ be the corresponding unobserved network performance quantities of interest. Let y_{ti}, x_{tj} be the ith and jth element of y_t and x_t respectively. Next we will illustrate the above setup in two concrete examples.

A. *Example: Multicast Inference of Internal Delay Distribution*

Packet link delay is a major indicator of the network performance. Two different approaches have been used for link delay monitoring: internal and external. The internal approach directly measures the delays at link-level interfaces, while the external approach monitors delays through end-to-end measurements.

The Multicast-based Inference of Network-internal Characteristics (MINC) Project [1] pioneered the use of multicast probing for network delay distribution estimation. The choice of end-to-end measurement through multicast probing is due to the limitations of the internal approach: 1) the collaboration between internal routers is not always available; 2) an extra heavy load burden might be caused by the probing process. A similar approach through unicast end-to-end measurements [7] can be found in Coate and Nowak (2001), but unicast methods suffer from scalability issues when large network is of interest as discussed in [16]. In this paper, we use the multicast-based external approach to estimate internal link delay distributions.

Consider a general multicast tree depicted in Fig. 2. Each probing packet with a time stamp sent from root node 0 will be received by all end receiver computers 4–7. For any pair of receivers, each packet experiences the same amount of delay over their common path. For instance, for copies of the same packet received at receiver 4 and 5, they experience the same amount of delay from 0 to 1 then to 2. Measurements are made at end receivers, hence only delays over the paths from root to end receivers are observed.

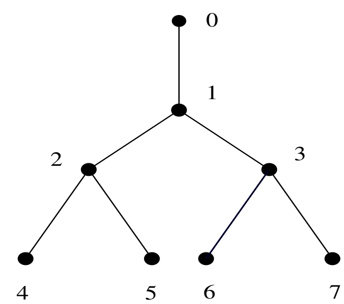


Fig. 2. An arbitrary virtual multicast tree topology with four receivers

Due to the aggregation of the measured delays, the network tomography model defined by (1) and (2) can be naturally applied to multicast inference of the internal delay distribution estimation problem. For each probing packet, $\mathbf{X}$ is the vector of unobserved delays over each link, and $\mathbf{Y}$ is the vector of observed path-level delays at each end receiver. When T probes are sent from root over a short interval, we observe iid samples $y_1, ..., y_T$. A is an $I \times J$ routing matrix determined by the multicast spanning tree, where I is the number of end receivers and J the number of internal links. The tree structure demands that $J = 2I - 1$.

Each link has a certain amount of minimal delay (overhead), which are assumed to be known beforehand. After compensating the minimal delay of each link, discretization is imposed on link-level delay by Lo Presti *et al* (1999). They proposed a multinomial model for the discretized link delay: X_j takes finite possible values $\{0, q, 2q, ..., mq, \infty\}$, where q is the bin width and m is a constant. When the delay is infinite, it implies the packet is lost during the transmission. The choice of m enables us to decide how fine we want to approximate the true delay distributions. Each X_j is an independent multinomial random variable with $\theta_j = (\theta_{j0}, \theta_{j1}, ..., \theta_{jm}, \theta_{j\infty})$.

In order to ensure identifiability, only canonical multicast trees are considered [16] and we assume:

$$\theta_{j0} = P(X_j = 0) > 0, \; j = 1, ..., J$$

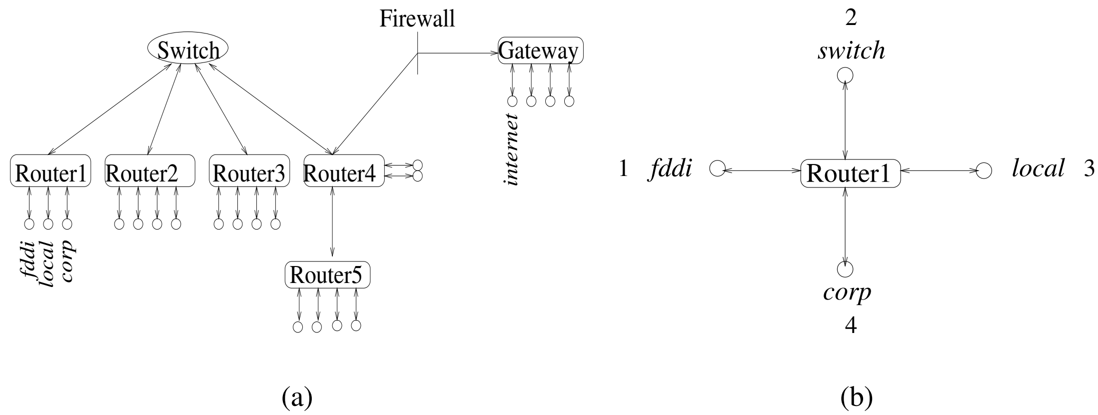


Fig. 3. (a) A Simple Router Network at Lucent Technologies, (b) Network Topology around Router 1.

i.e. each individual packet has a positive probability to have zero delay over any internal link.

B. Example: Origin-Destination Matrix Inference

Origin-destination (OD) tomography quantifies the path-level network dynamics through viable network measurements. OD matrix estimation is very important because it is a key input to any routing algorithm, e.g., in the Open Shortest Path First (OSPF) routing protocol, OD matrix is a common metric used to assign appropriate weights on each link.

Two approaches are possible to carry out the OD matrix estimation: direct and indirect. The direct approach is made possible via some router software such as *Netflow* supported by Cisco routers [11]. Link traffic counts at router interfaces are much easier to obtain relative to the direct approach. Meanwhile the indirect approach does not impose such a data collection burden on routers. The aggregation of OD traffic at link interfaces leads to an inverse problem. In this paper, we will concentrate on the indirect approach through link traffic counts.

Vardi [17] first investigated this problem within a general network topology, assuming independent Poisson distributions for the OD traffic counts. The identifiability of the model is proven under the Poisson model and the expectation-maximization (EM) algorithm is used to estimate the parameters for both deterministic and Markov routing schemes. In order to ease the difficulty in implementing the EM algorithm, a moment estimation method is proposed in Vardi's paper, but it is far less efficient than the maximum likelihood estimate (MLE).

With real network data, Cao *et al* [4] revise the Poisson assumption and approximate the OD traffic byte counts with normal distributions under a mean-variance relationship. Their methodology is validated by comparisons with directly collected OD traffic in a simple network at Lucent Technologies depicted in Fig. 3.

The general network tomography model denoted by (1) and (2) is applicable to the OD matrix inference through link traffic counts. In the model, $\mathbf{Y} = (Y_1, Y_2, ..., Y_I)'$ is the vector of observed traffic byte counts measured on each link interface during a given time interval and $\mathbf{X} = (X_1, X_2, ..., X_J)'$ is the corresponding vector of unobserved true OD traffic byte counts at the same time period. For instance, in Fig. 3(b), one element of $\mathbf{X}$ corresponds to the number of bytes originated from *fddi* and destined for *local*, and one element of $\mathbf{Y}$ corresponds to the number of bytes observed at link from *fddi* to *Router1*, which consists all traffic originated from *fddi*. $\mathbf{X}$ is called OD traffic matrix, even though it is arranged as a column vector for notational convenience. Under a fixed routing scheme, $\mathbf{Y}$ is determined uniquely by $\mathbf{X}$ through the $I \times J$ routing matrix, A, in which I is the number of measured incoming/outgoing unidirectional links and J is the number of possible OD pairs.

Assume that each component of OD traffic counts $\mathbf{X}$ is normally distributed, and satisfies the mean-variance relationship: $X_j \sim N(\lambda_j, \phi\lambda_j)$ independently, where ϕ is a positive scaler applicable to all OD pairs. The distribution can also be rewritten as

$$\mathbf{X} \sim N(\lambda, \Sigma), \tag{3}$$

where $\Sigma = \phi \mathrm{diag}(\lambda)$ and the parameter of the full model is $\theta = (\phi, \lambda)$ and

$$\mathbf{Y} = A\mathbf{X} \sim N(A\lambda, A\Sigma A').$$

The mean-variance relationship implies that each OD pair with larger traffic byte counts tends to have larger variance, which is a key assumption to ensure the identifiability of the OD matrix tomography problem. In [4], the mean-variance relation is stated as $\mathrm{Var}(X_j) = \phi\lambda_j^c$, where λ_j is the mean of X_j and c is a predetermined constant. Usually c takes value 1 or 2. Reference [4] shows that $c = 1$ and $c = 2$ give very similar results. So in this paper, we use $c = 1$ as done in [5].

The non-stationarity of network traffic is addressed by a local likelihood model in Cao *et al* (2000) and an iterative proportional fitting (IPF) algorithm is employed to estimate the OD traffic x_t. We apply a pseudo likelihood approach to the same Lucent data based on the above model specification. Parameter and OD traffic counts estimates from both full and pseudo likelihood will be compared with respect to the computational complexity and estimation efficiency in later section.

III. Pseudo Likelihood in Network Tomography Inference

A. Forming Sub-problems

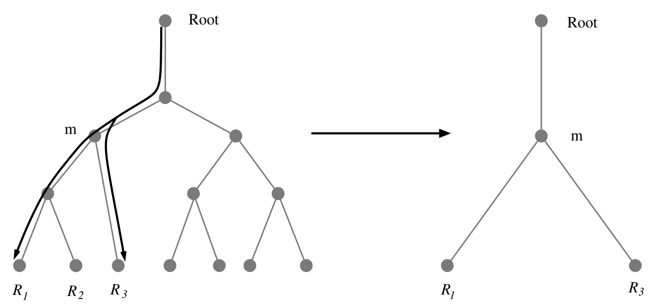


Fig. 4. Pseudo Likelihood: subtree decomposition

Either in the problem of internal link delay distribution estimation through multicast end-to-end traffic data, or in the problem of OD matrix inference through link byte counts, the maximum likelihood estimate (MLE) is computationally intensive. Especially for the former problem, it would be almost impossible to get the exact MLE estimate for a fairly large multicast tree because for each observed end delay vector $\mathbf{Y}$, the task of keeping track of all possible combinations of internal link delays $\mathbf{X}$ is an NP problem. Maximum pseudo likelihood would be desirable if its computation cost is much lower while its estimation efficiency is comparable to MLE.

One way to construct our pseudo likelihood comes from the decomposition of the multicast spanning tree. Consider a subtree decomposition scheme depicted in Fig. 4. A virtual two-leaf subtree is formed by only looking at two receivers R_1, R_3 in the original multicast tree. MLE for the virtual two-leaf subtree is computable because of its much simpler dependence structure. It is worth noting that because one virtual link in the subtree may consist of several real links, each subtree will give the specific path-level delay distributions but not link-level delay distributions. For a multicast tree with I end receivers, $I(I-1)/2$ subtree decomposition schemes are available: different subtrees give delay distributions on different paths. Putting all these path-level delay distributions together enables us to recover the link-level delay distributions. But a more efficient way of estimating link-level delay parameters is to consider all sub-problems simultaneously. Subtree problems are dependent, but if we ignore such dependences, and treat them as independent questions, then our pseudo likelihood function can be obtained by multiplying likelihood functions of all sub-problems.

The above treatment for internal link delay distribution estimation is not unique. Forming subtrees in Fig. 4 is equivalent to selecting two rows from the routing matrix A. Picking up three or even more rows each time may also sound reasonable, but there is a trade-off between the computational complexity incurred and the estimation efficiency achieved by taking more dependence structures into account. Our experience with the two examples shows that picking up two rows each time gives satisfactory estimation results while keeping the computational cost within a reasonable range.

To be more precise, let S denote the set of all possible sub-problems by picking up two rows from the routing matrix A:

$$S = \{s = (i_1, i_2) : 1 \leq i_1 < i_2 \leq I\}.$$

For each sub-problem $s \in S$,

$$\mathbf{Y}^s = A^s \mathbf{X}^s, \tag{4}$$

where $\mathbf{X}^s$ is the vector of network dynamic components involved in the given sub-problem s, A^s is the corresponding sub-routing matrix and $\mathbf{Y}^s = (Y_{i_1}, Y_{i_2})'$ is the corresponding observed network performance measurements in the sub-problem s.

Let θ^s be the model parameter of the sub-problem s, and $l^s(\mathbf{Y}^s; \theta^s)$ be its corresponding log-likelihood function. For the observed measurement vector $\mathbf{Y}$, define

$$L^p(\mathbf{Y}; \theta) = \sum_{s \in S} l^s(\mathbf{Y}^s; \theta^s). \tag{5}$$

Given observed independent data vectors $y_1, ..., y_T$, let $x_1^s, ..., x_T^s$ and $y_1^s, ..., y_T^s$, respectively, be the corresponding unobserved and observed data vectors for the sub-problem s. The overall pseudo log-likelihood function is then defined as:

$$L_T^p(y_1, ..., y_T; \theta) = \sum_{t=1}^{T} L^p(y_t; \theta). \tag{6}$$

Maximizing the pseudo likelihood function gives the maximum pseudo likelihood estimate (MPLE) of θ. Usually it is equivalent to find solution to the following pseudo likelihood equation:

$$\frac{\partial}{\partial \theta} L_T^p(y_1, ..., y_T; \theta) = 0. \tag{7}$$

B. Asymptotic Properties of MPLE

Consistency and asymptotic normality are basic properties of MLE. Under some very general conditions, the consistency and asymptotic normality of MPLE in (6) can be proved. In the rest of the paper, let θ_0 be the true parameter of the model defined in (1) and (2), and θ_0^s be the true parameter in a sub-problem s.

Theorem 1: **(Consistency)** For a network tomography model defined in (1) and (2), assume following conditions are satisfied:

A1) $L^p(y;\theta)$ is distinct, i.e., for any $\theta_1 \neq \theta_2$, there exists a set Δ in sample space $\mathcal{R}^I$, such that $P_{\theta_0}(\Delta) > 0$ and for all $y \in \Delta$, $L^p(y;\theta_1) \neq L^p(y;\theta_2)$;

A2) The parameter space contains an open interval ω of which the true parameter θ_0 is an interior point;

A3) $L_T^p(y_1, ..., y_T;\theta)$ is differentiable with respect to θ. The pseudo likelihood equation (7) has unique solution in ω almost surely when $T \to \infty$.

Then the pseudo likelihood estimate $\tilde{\theta}_T^p$ is consistent, i.e., $\tilde{\theta}_T^p \to \theta_0$ in probability as $T \to \infty$.

For asymptotic normality, stronger conditions are needed to ensure the second order expansion at the neighborhood of the true parameter θ_0. Assume that the pseudo log-likelihood function $L_T^p(y_1, ..., y_T;\theta)$ is twice continuously differentiable. Let $H(\theta) = \mathrm{E}_{\theta_0}(\nabla^2 L^p(Y;\theta))$ and $B(\theta) = \mathrm{Var}_{\theta_0}(\nabla L^p(Y;\theta))$.

Theorem 2: **(Asymptotic normality)** If in additional to the assumptions specified in Theorem 1, the following conditions are also satisfied:

B1) $L_T^p(y_1, ..., y_T;\theta)$ is twice continuously differentiable with respect to θ. In addition, expectation and differentiation operations of $L_T^p(y_1, ..., y_T;\theta)$ can be inter-exchanged;

B2) With probability 1, as $T \to \infty$, the Hessian matrix $\nabla^2 L_T^p(y_1, ..., y_T;\theta)$ is invertible in ω;

B3) For $\theta \in \omega$, $\nabla^2 L_T^p(y_1, ..., y_T;\theta) \to H(\theta)$ in distribution uniformly.

Then the maximum pseudo likelihood estimate $\tilde{\theta}_T^p$ is strongly consistent and as $T \to \infty$,

$$\sqrt{T}(\tilde{\theta}_T^p - \theta_0) \to N(0, C(\theta_0)) \text{ in distribution},$$

where $C(\theta) = H(\theta)^{-1}B(\theta)H(\theta)^{-1}$.

Usually, the uniformly convergence of $\nabla^2 L_T^p$ to $H(\theta)$ is verified by checking the boundness of $\mathrm{E}\partial^3 L_T^p/\partial\theta_j\partial\theta_k\partial\theta_l$ in the open interval ω. It is true for most analytic functions such as L_T^p. This result is used in later theorem proofs.

C. Pseudo EM Algorithm

Maximizing the pseudo function leads to our estimate, but usually the pseudo likelihood equation (7) cannot be solved analytically, hence a numeric optimization algorithm has to be adopted. The EM algorithm [10] is a well known method for maximizing the likelihood function numerically. EM will not work for any objective function. However Theorem 3 shows that Pseudo EM (an EM like algorithm) is applicable in maximizing the pseudo likelihood.

Let $l^s(\mathbf{X}^s;\theta^s)$ be the log-likelihood function on a sub-problem s given the complete data $\mathbf{X}^s$. Let $\theta^{(k)}$ be the estimate of θ obtained in the kth step; then the objective function $Q(\theta,\theta^{(k)})$ to be maximized in the $(k+1)$th step of the Pseudo EM algorithm is defined as

$$Q(\theta,\theta^{(k)}) = \sum_{s\in S}\sum_{t=1}^{T} \mathrm{E}_{\theta^{s(k)}}\left(l^s(x_t^s;\theta^s)|y_t^s\right). \tag{8}$$

Theorem 3: During the Pseudo EM iteration steps, the value of the objective pseudo log-likelihood function L_T^p is non-decreasing. If L_T^p is unimodal, then Pseudo EM algorithm will converge to the unique maximum point.

IV. Applications of Pseudo Likelihood Estimation

First we define two notations which will be used in Pseudo EM computation below:

$$\text{given } s,\ J^s = \{j:\ \text{the } j\text{th OD pair is involved in } s\};$$

$$\text{given } j,\ S^j = \{s:\ \text{the } j\text{th OD pair is involved in } s\}.$$

A. Inference of Multicast Internal Delay Distribution

1) Parameter Estimation through Pseudo EM: For the problem of the internal delay distribution estimation through multicast end-to-end measurements, $\mathbf{Y}$ is the observed delay vector at end receiver computers and $\mathbf{X}$ is the corresponding unobserved link-level delay vector.

For any sub-problem s, each component of $\mathbf{X}^s$ is an independent multinomial random variable so that the log-likelihood function given the complete data $x_1^s,...,x_T^s$ is

$$l^s(x_1^s,...,x_T^s;\theta^s) = \sum_{j\in J^s}\sum_{l} n_{jl}^s \log(\theta_{jl})$$

where $\theta_{jl} = P(X_j = l)$, $n^s_{jl} = \sum_t \mathbf{1}_{\{x^s_{tj}=l\}}$.

Lo Presti *et al*'s recursive algorithm [16] is a computational efficient method to estimate the internal delay through multicast end-to-end traffic. Comparison between recursive algorithm and pseudo likelihood method will be given below. In our Pseudo EM implementation, the estimate from the recursive algorithm serves as the initial parameter estimate: $\theta^{(0)}$.

Let $\theta^{(k)}$ be the estimate of the parameter obtained in the kth step of Pseudo EM. According to (8), we have

$$Q(\theta, \theta^{(k)}) = \sum_{s\in S}\sum_{j\in J^s}\sum_{l} \log(\theta_{jl}) E_{\theta^{s(k)}}\left(\sum_{t=1}^{T} \mathbf{1}_{\{x^s_{tj}=l\}} \middle| y^s_t\right).$$

E-step: Compute

$$\hat{n}_{jl} = \sum_{s\in S} E_{\theta^{s(k)}}\left(\sum_{t=1}^{T} 1(x^s_{tj}=l) \middle| y^s_t\right).$$

M-step: Update $\theta^{(k)}$ by

$$\theta^{(k+1)}_{jl} = \frac{\hat{n}_{jl}}{\sum_{r\in R}\hat{n}_{jr}}, \text{ where } R = \{0, 1, ..., m, \infty\}$$

As before, m is the maximum delay level for each link, I is the total number of end receivers. Let P be the average number of links per path. The overall complexity of each step of the Pseudo EM algorithm is then $O(m^3 I^2 P^2)$.

Theorem 4: Let $\tilde{\theta}^p_T$ be the internal delay distribution estimate through pseudo likelihood. Then $\tilde{\theta}^p_T$ is a consistent estimator and when $T \to \infty$, $\sqrt{T}(\tilde{\theta}^p_T - \theta_0)$ converges in distribution to a multivariate normal random variable.

2) Experimental results: First, in order to compare the performance of the pseudo likelihood method and the recursive algorithm, the following model simulations are carried out on a 4-leave multicast tree depicted in Fig. 2. IID multicast delay measurements are generated from independent distributions on internal links: each link has a discrete delay distribution with $m = 13$. Fig. 5 shows delay distribution estimates of 3 arbitrarily selected links, in which left and right columns give the estimates when $T = 2000$ and $T = 20000$ respectively. Fig. 5 demonstrates that the MPLE can track the delay distribution of a network well when $T = 2000$, but even when $T = 20000$, the recursive algorithm still gives estimates quite off the truth.

Secondly, the same procedure is repeated 100 times independently for $T = 2000$. The root mean square (RMS) of the estimate errors are defined as $\sqrt{\sum_{k=1}^{100}(\hat{\theta}^k_{jl} - \theta^0_{jl})^2/100}$, where $\hat{\theta}^k_{jl}$ can be either MPLE or recursive algorithm estimates in the kth run and θ^0_{jl} are the true parameters. The natural logarithm scaled RMS ratio (RMS error of recursive

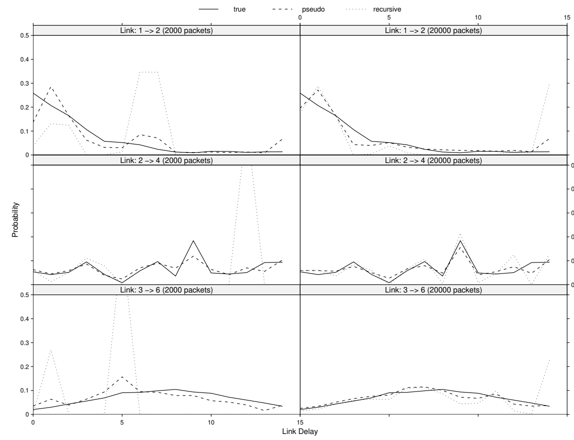
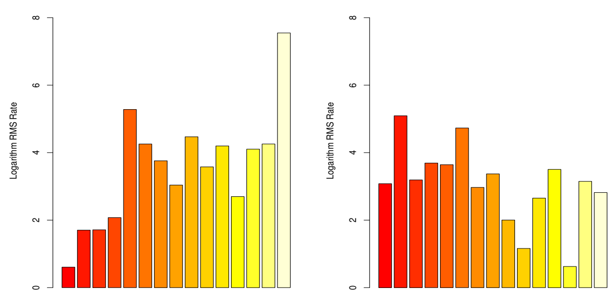


Fig. 5. Delay distribution estimates of 3 arbitrarily selected internal links. Left column: T = 2000. Right column: T: 20000.

algorithm over that of MPL) is shown in Fig. 6. For a given link j, the 15 bars correspond to its 15 delay parameters. Bars with positive height imply that recursive algorithm has larger estimation errors. The plot shows that the MPLE has uniformly smaller estimation errors and the differences are substantial: for 25 out of 30 parameters shown in Fig. 6, recursive estimate always gives RMS error larger than MPLE by at least a factor of $e^2 = 7.39$.



Link $2 \to 3$ Link $4 \to 5$

Fig. 6. Natural logarithm scaled RMS ratio for two arbitrarily selected links. (RMS error of recursive algorithm over RMS error of MPL)

B. OD Matrix Tomography

1) Parameter Estimation through Pseudo EM: For the problem of the OD matrix estimation through link byte counts, $\mathbf{Y}$ is the observed vector of link-level byte counts during a given time period and $\mathbf{X}$ is the corresponding unobserved OD byte counts.

For each sub-problem s, $\mathbf{Y}^s$ is the vector of observed traffic byte counts and A^s is the sub-routing matrix. $\mathbf{X}^s$ is the vector of OD pair traffic counts involved in s. Let λ^s, Σ_s be the mean vector and covariance matrix of $\mathbf{X}^s$

respectively. The parameter of the sub-problem s is then $\theta^s = (\phi, \lambda^s)$. The log-likelihood function for sub-problem s given complete data $x_1^s, ..., x_T^s$ is

$$l^s(x_1^s, ..., x_T^s; \theta^s) = -\frac{T}{2}\log|\Sigma_s| - \frac{1}{2}\sum_{t=1}^{T}(x_t^s - \lambda^s)'\Sigma_s^{-1}(x_t^s - \lambda^s).$$

Let $\theta^{(k)}$ be the parameter estimate in the kth step. According to (8), the objective function to be maximized in the $(k+1)$th step is

$$\begin{aligned} Q(\theta, \theta^{(k)}) &\propto -\sum_{s\in S}\left\{T\left(\log|\Sigma_s| + \mathrm{tr}(\Sigma_s^{-1}\mathcal{R}^{s(k)})\right) + \sum_{t=1}^{T}(m_t^{s(k)} - \lambda^s)'\Sigma_s^{-1}(m_t^{s(k)} - \lambda^s)\right\} \\ &= -\sum_{s\in S}\sum_{j\in J^s}\left\{\sum_{t=1}^{T}\frac{(m_{tj}^{s(k)} - \lambda_j)^2}{\phi\lambda_j} + T\left[\log\phi + \log\lambda_j + \frac{r_j^{s(k)}}{\phi\lambda_j}\right]\right\} \end{aligned} \tag{9}$$

where

$$\begin{aligned} m_t^{s(k)} &= E_{\theta^{s(k)}}(x_t^s|y_t^s) &&= \lambda^{s(k)} + \Sigma_s^{(k)}A^{s\prime}(A^s\Sigma_s^{(k)}A^{s\prime})^{-1}(y_t^s - A^s\lambda^{s(k)}), \\ \mathcal{R}^{s(k)} &= \mathrm{Var}_{\theta^{s(k)}}(x_t^s|y_t^s) &&= \Sigma_s^{(k)} - \Sigma_s^{(k)}A^{s\prime}(A^s\Sigma_s^{(k)}A^{s\prime})^{-1}A^s\Sigma_s^{(k)}, \end{aligned}$$

and $r_j^{s(k)}$ and $m_{tj}^{s(k)}$ are, respectively, the elements in $\mathcal{R}^{s(k)}$ and $m_t^{s(k)}$ corresponding to OD pair j.

Let d_j be the number of elements in S^j, and

$$\begin{aligned} a_j^{(k)} &= \frac{1}{d_j}\sum_{s\in S^j}\left(r_j^{s(k)} + \frac{1}{T}\sum_{t=1}^{T}(m_{tj}^{s(k)})^2\right), \\ b_j^{(k)} &= \frac{1}{Td_j}\sum_{s\in S^j}\sum_{t=1}^{T}m_{tj}^{s(k)}, \end{aligned}$$

then it can be shown that the system equation $\partial Q/\partial\theta(\theta, \theta^{(k)}) = 0$ is equivalent to

$$\lambda_j^2 + \phi\lambda_j - a_j^{(k)} = 0, \quad j = 1, ..., J \tag{10a}$$

$$\sum_{j=1}^{J}(\lambda_j - b_j^{(k)}) = 0. \tag{10b}$$

E-step: Compute coefficients $a_j^{(k)}$ and $b_j^{(k)}$. Comparing with full likelihood method, the computation is much faster because $A^s\Sigma_sA^{s\prime}$ is a 2×2 matrix and hence there is no need to invert a very high dimensional matrix like full likelihood method does.

M-step: Solve (10). Equation (10a) shows quadratic constraint between λ_j and ϕ. In conjunction with these solutions, (10b) boils down to a monotonic functional constraint on ϕ, therefore fast searching algorithms are available to solve these equations. Because the cost of the E-step is relatively high, a Multiple-Step Gradient EM algorithm (a natural extension to Lange's Gradient EM algorithm [12]) is employed to solve these equations only roughly.

The starting point $\theta^{(0)}$ can be quite arbitrary for the Pseudo EM algorithm. For the OD matrix inference, we adopt the initial value used in [4], such that $\lambda^{(0)}$ is a constant vector with each component to be $\sum_{t=1}^{T} \mathbf{1}' y_t / (T \mathbf{1}' A \mathbf{1})$ and $\phi^{(0)} = \mathrm{Var}(y_{ti})/\mathrm{E}(y_{ti})$, where the sample variance and expectation is computed by pooling all y_{ti} together. Such starting point gives very stable performance for the Lucent and simulated data set.

For a network with n nodes, the number of observed link counts I is $O(n)$ and J, the number of OD pairs, is n^2. Assume that the average number of links per OD pair path is $O(\sqrt{n})$. For the pseudo likelihood approach, the overall computational complexity of each step of the EM algorithm is $O(n^{3.5})$. As discussed in [5], the complexity of the full MLE is $O(n^5)$ after exploiting sparse matrix calculation. Therefore the pseudo likelihood approach can be more scalable to larger networks.

Theorem 5: Let B be an $[I(I+1)/2] \times J$ matrix whose rows are the rows of A and the component-wise products of each different pair of rows from A. If B is of full column rank, then the model is identifiable. Let $\tilde{\theta}_T^p$ be the parameter estimate through pseudo likelihood, then $\tilde{\theta}_T^p$ is a consistent estimator and when $T \to \infty$, $\sqrt{T}(\tilde{\theta}_T^p - \theta)$ converges in distribution to a multivariate normal random variable.

2) Experimental results: First we consider a small real network with only 4 nodes [4] at Lucent Technologies depicted in Fig. 3(b). For this network, the true OD traffic counts collected through *Netflow* in every 5 minutes are available. Moreover the MLE is also feasible to compute due to its small size.

For this 4-node network, there are 4 incoming and 4 outgoing links; the total number of OD pairs J is 16. The routing matrix A (0 for entries not indicated) is

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

Each column of A is associated with an OD traffic and each row corresponds to an observed link traffic. For instance, the first row of A means that traffic originated from node 1 consists of unobservable OD traffic counts originated from

1, destinated for 1, 2, 3, and 4.

A local moving IID model [4] is used to capture the time varying nature of the network traffic by assuming the traffic counts are independent within a fixed size moving window. In this paper, we adopt the window size to be 11 from Cao *et al.* In order to have better comparison, we apply the pseudo likelihood approach to the data collected on Feb 22, 1999, the same day used in [4]. This data set consists of 288 data points. Fig. 8 shows the MPLE of λ along with the MLE estimate. For comparison, the 11-points moving average of true OD traffic counts are plotted too. We can see that both pseudo and full likelihood methods capture the dynamics of the OD traffic counts quite well.

Estimating the actual time-varying OD traffic counts x_t is the ultimate goal of the OD matrix inference. Iterative proportional fitting (IPF) [9] is widely used in contingency table analysis: it adjusts the table to match the observed margins. Borrow the same idea from [4], $E_{\hat{\theta}}(x_t|y_t, x_t > 0)$ serves as the initial point for the IPF algorithm, where $\hat{\theta}$ is the parameter estimate from maximizing either pseudo likelihood or full likelihood. Estimates of OD network traffic counts near high peaks usually have a relatively smaller error rate. In order to exhibit small-scale features, a zoomed-in version of the OD traffic counts estimate with its vertical axis magnified by a factor of 10 is shown in Fig. 9. The plot demonstrates that both pseudo and full likelihood methods have quite comparable performance even in error-prone small scales.

The above computations are completed using R 1.5.0 on a 1G Hz laptop. To produce Fig. 8, it takes about 12 seconds for computing the MPLE, and about 49 seconds for the MLE. In the pseudo likelihood approach, the computations of the coefficients $a_i^{(k)}$ and $b_i^{(k)}$ are done by C codes because the performance of R will be severely affected by multiple loops introduced by dealing with numerous sub-problems. Similarly, the EM algorithm is used to compute MLE and the only difference between EM and Pseudo EM in this problem is how they compute coefficients $a_i^{(k)}$ and $b_i^{(k)}$. In the E-step of EM for the full likelihood method, one matrix inversion and a few matrix multiplications are needed. All operations can be done in R very efficiently, hence introduction of C code in the E-step of the full likelihood method will barely speed up its execution.

Secondly, in order to assess the performance of MPLE and MLE more thoroughly, we use network simulator (NS) [15] to generate OD traffic byte counts on some larger networks. a) OD traffic is simulated on a two-router network depicted in Fig. 7. Every link is set to have a capacity of 1.5Mb/sec and a propagation delay of 10ms with default drop tail queueing management and every OD pair is assigned a bit rate. During the simulation, the origin node generates packets to the destination node as a bit stream of the given rate comprising 160 byte UDP packets according to an exponential on-off process with burst time to be 20ms and idle time to be 120ms. The background traffic consists of a mixture of exponential and pareto on-off sources using TCP or UDP. Each background connection is randomly attached to an OD pair and has random start stop times and random bit rate. Flow monitor is utilized to collect flow i.e. OD traffic statistics every second. The simulation lasts 288 seconds: every second represents a 5-minute interval in a 24 hours long data collection. Fig. 10 shows an excellent agreement between the traffic count estimates from both pseudo and full likelihood methods and the simulated OD traffic data. It takes about 18 seconds in computing the MPLE, while around 88 seconds in computing the MLE. b) A larger simulation is carried out on the Lucent network illustrated in Fig. 3(a), which comprises 21 end nodes and 27 links. The OD traffic estimate plots (not shown) once again show that both MPLE and MLE capture the dynamic of the simulated OD traffic and yield similar results. The execution times of pseudo and full likelihood methods are 2.5 minutes and 39.9 minutes respectively.

The comparison of execution times between MLE and MPLE exhibits that the pseudo likelihood approach speeds up the computation without lossing much estimation performance, hence it is more scalable to larger networks.

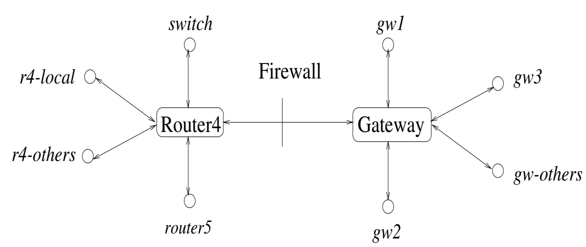


Fig. 7. Two-Router Network at Lucent Technologies.

V. Conclusion

In this paper, we used two examples to demonstrate the potential of the pseudo likelihood approach for network tomography. The MPLE shows strengths through its estimation efficiency and manageable computational complexity. The basic idea of divide-and-conquer is not new, but is very powerful when combined with pseudo likelihood for large network problems. We believe more decomposition schemes may emerge to solve other network tomography problems.

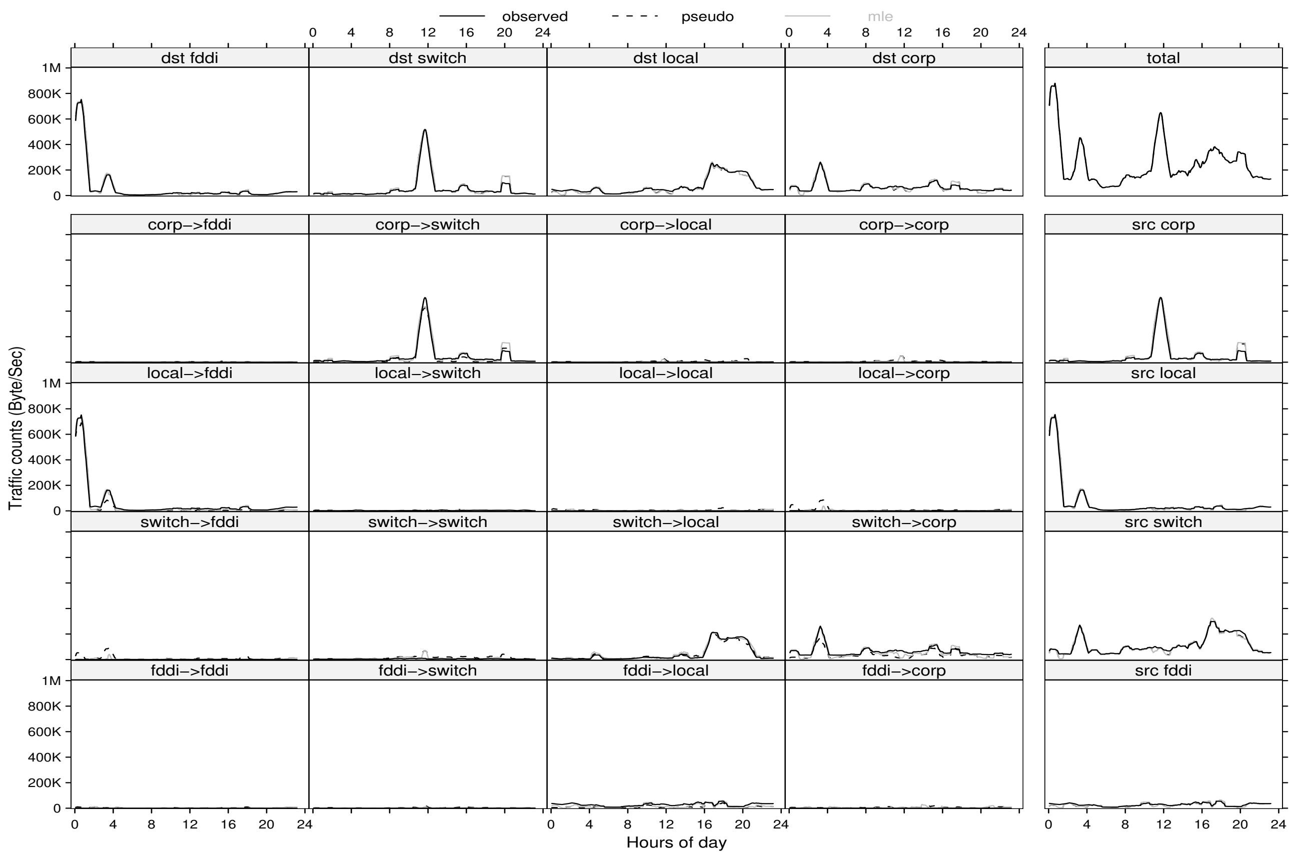


Fig. 8. Mean OD traffic estimates $\hat{\lambda}_t$ obtained from pseudo and full likelihood methods for 4 nodes network around *Router1* against the moving average of the true OD traffic. Marginal traffic is shown in side panels.

Acknowledgment

We would like to thank Jonathan Grib and Mike Last for very helpful comments on the presentation of the paper. And we thank Jin Cao and Scott Vander Wiel for sharing the collected traffic data on the Lucent network and the Splus code.

Appendix

Proofs of Theorem 1 and Theorem 2 are omitted. Almost identical proofs can be found in [13] and [18]. Cares only need to be taken about the fact we are working on pseudo log-likelihood function, which in fact is summation of a series of log-likelihood functions. Please see Lehmann (1998) and White (1994) for more details.

For fixed observation data vectors $y_1, ..., y_T$, let $L_T^p(\theta)$ be a shorthand notation for the pseudo log-likelihood function $L_T^p(y_1, ..., y_T; \theta)$, and $l^s(\theta^s) = l^s(y_1^s, ..., y_T^s; \theta^s)$.

Proof of Theorem 3

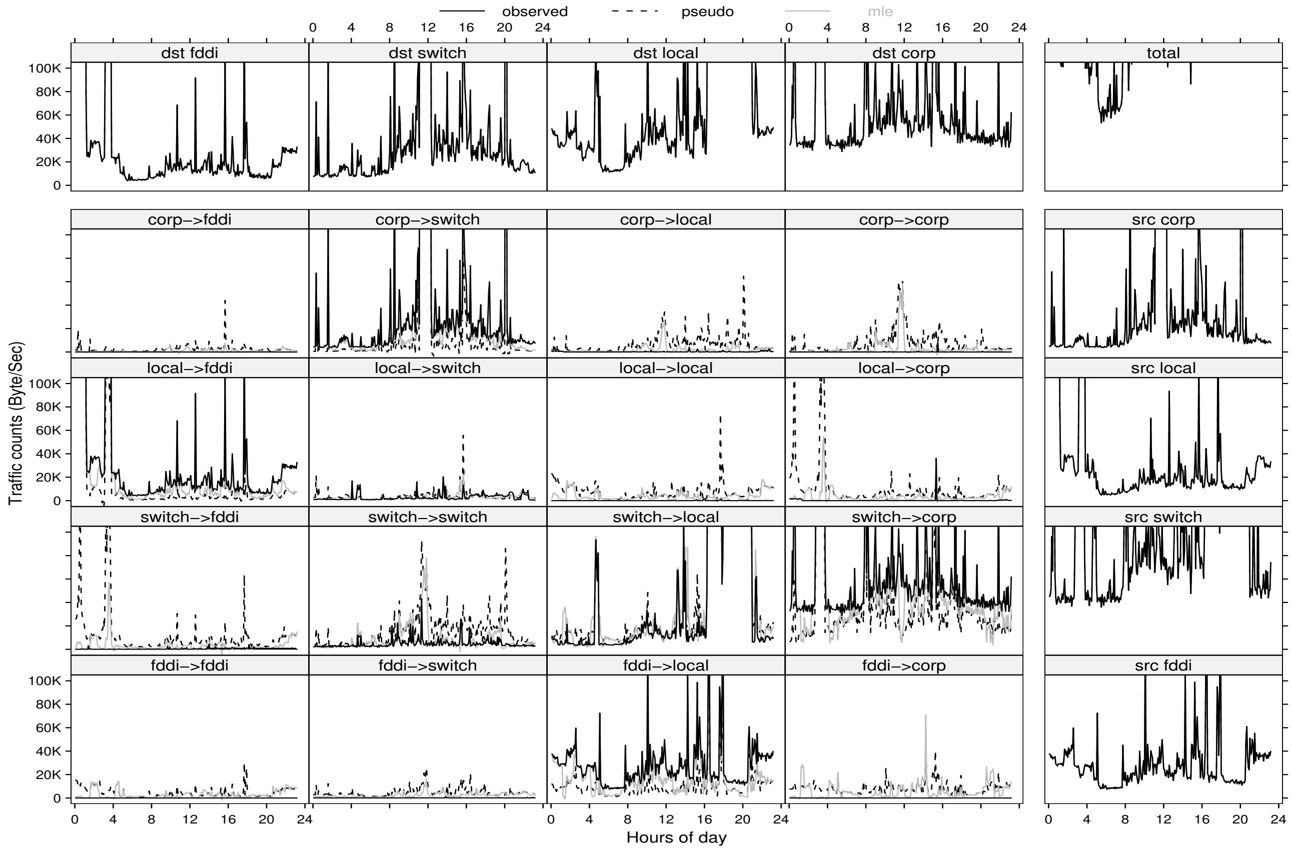


Fig. 9. OD traffic counts estimates $\hat{x}_t$ obtained from pseudo and full likelihood methods against the true OD traffic counts for 4 nodes network around *Router1*.

Proof: Given sub-problem s, let $f_c^s(x^s;\theta^s)$ be the likelihood function given the complete data x^s, and $f^s(y^s;\theta^s)$ be the likelihood function based on observable y^s. Define

$$k^s(x^s|y^s;\theta^s) = f_c^s(x^s;\theta^s)/f^s(y^s;\theta^s),$$

then for any fix θ, we have

$$L_T^p(\theta) = Q(\theta,\theta^{(k)}) - H(\theta,\theta^{(k)}),$$

where

$$H(\theta,\theta^{(k)}) = \sum_{s\in S}\sum_{t=1}^{T} E_{\theta^{s(k)}}\left(\log k^s(x_t^s|y_t^s;\theta^s)\middle|y_t^s\right).$$

For $L_T^p(\theta^{(k+1)}) - L_T^p(\theta^{(k)})$,

$$Q(\theta^{(k+1)},\theta^{(k)}) \geq Q(\theta^{(k)},\theta^{(k)}).$$

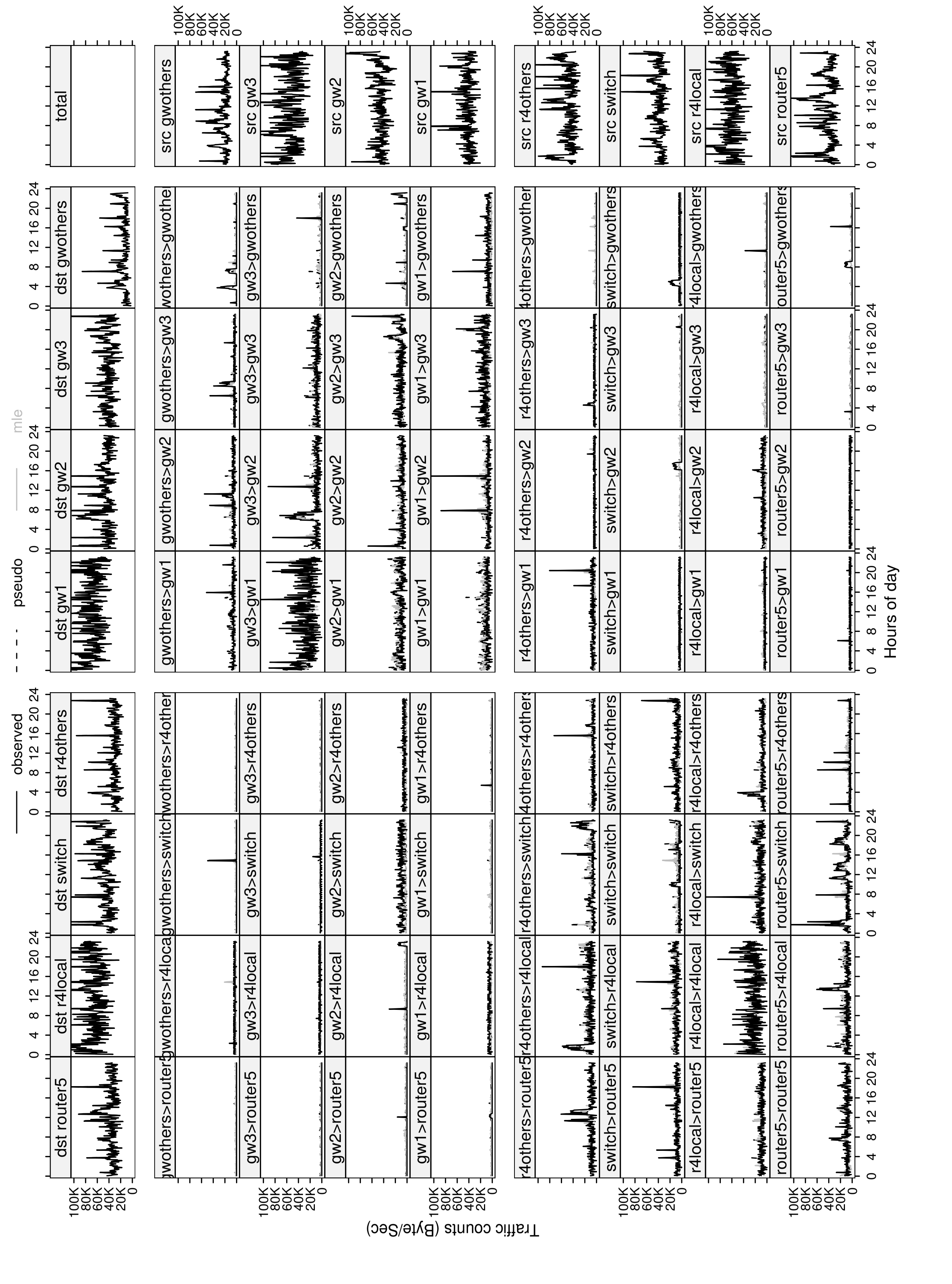


Fig. 10. OD traffic count estimates $\hat{x}_t$ obtained from pseudo and full likelihood methods against the true OD traffic for NS simulated data on the two-router network depicted in Fig. 7. Marginal traffic is shown in the side panels.

because $\theta^{(k+1)}$ is chosen to be the maximum point of function $Q(\theta,\theta^{(k)})$. Also given any θ,

$$\begin{aligned} H(\theta,\theta^{(k)}) - H(\theta^{(k)},\theta^{(k)}) &\leq \sum_{s\in S}\sum_{t=1}^{T} \log E_{\theta^{s(k)}} \left(\frac{k^s(x^s|y^s;\theta^s)}{k^s(x^s|y^s;\theta^{s(k)})} \middle| y^s \right) \\ &= \sum_{s\in S}\sum_{t=1}^{T} \log \int_{\mathcal{X}^s(y)} k^s(x^s|y^s;\theta^s)dx^s \\ &= 0. \end{aligned}$$

The first inequalify holds due to Jensen's inequality and the concavity of the logorithmic function. The second equality holds because $k^s(x^s|y^s;\theta^{s(k)})$ is the conditional density of x^s given y^s when parameter is $\theta^{s(k)}$ then it is cancelled when integrating out the conditional expectation. These establish that the pseudo log-likelihood function $L_T^p(\theta)$ is non-decreasing during Pseudo EM iterations.

The above result lies in the fact that in its pure format, the pseudo log-likelihood function could really be a log-likelihood function given the artificial assumption of independent sub-problems. In Pseudo EM, all conditional expectations given observed data are taken based upon this artificial assumption, hence there is no real difference between EM and Pseudo EM. Therefore generic results of EM algorithm are applicable to the Pseudo EM algorithm without any modification. So if the pseudo likelihood function is unimodal, then the Pseudo EM sequence $\{\theta^{(k)}\}$ will converge to the unique MPLE. For more details, see Chpt. 3 of [14]. ■

Proof of Theorem 4

Proof: First we can show that the distinctness of $L_T^p(\theta)$ is equivalent to that for any $\theta_1 \neq \theta_2$, there are some s, $l^s(y^s;\theta_1^s)$ differs from $l^s(y^s;\theta_2^s)$. If $l^s(y^s;\theta_1^s) = l^s(y^s;\theta_2^s)$ for all s, then for each s, it is easy to show each corresponding sub-path, e.g., $0 \to m$, $m \to R_1$ and $m \to R_3$ in Fig. 4(b), has the same distribution under either θ_1 or θ_2. Iterate the argument along the whole multicast tree, we have $\theta_1 = \theta_2$. It implies the distinctness of $L_T^p(\theta)$.

Secondly, we will show $L_T^p(\theta)$ is strictly convex. In sub-problem s, let p_{kl} be the sum of $\prod_{j\in J^s}\theta_{jr}$ in which $\mathbf{X}_j = r$ will satisfy the end constraint $\mathbf{Y}_1^s = k, \mathbf{Y}_2^s = l$, then the log-likelihood function $l^s(\theta) = \sum_{kl} n_{kl}\log(p_{kl})$, where $n_{kl} = \sum_t \mathbf{1}_{\{y_1^s=k,y_2^s=l\}}$. The convexity of $\log(p_{kl})$ implies the convexity of $l^s(\theta)$, hence that of $L_T^p(\theta)$. The proof of strictness is similar to that of distinctness. Therefore the solution to the pseudo likelihood equation (7) is unique. By Theorem 1, the MPLE is consistent.

Suppose the true parameter θ_0 is in the interior of the paramter space, it is obvious that in a small neighborhood of

θ_0, the third derivative of of $L_T^p(\theta)$ exists and its expectation is bounded. By Theorem 2, when $T \to \infty$, $\sqrt{T}(\tilde{\theta}_T^p - \theta_0)$ converges in distribution to a normal distributed random variable. ■

Proof of Theorem 5

Proof: Under assumption (3), parameter $\theta = (\lambda, \phi)$. Similarly, $L_T^p(\theta)$ is distinct if $l^s(\theta_1^s) = l^s(\theta_2^s)$ for all sub-problem s, then $\theta_1 = \theta_2$. For a sub-problem $s = (i_1, i_2)$, let B^s be a $3 \times J$ matrix, with the i_1th, i_2th rows of A be the first two rows and their component-wise product be the third row. $l^s(\theta_1^s) = l^s(\theta_2^s)$ implies

$$\phi_1 = \phi_2, B^s\lambda_1 = B^s\lambda_2 \tag{11}$$

Thus, we have $B\lambda_1 = B\lambda_2$ and $\phi_1 = \phi_2$. Because B has full column rank, $B\lambda_1 = B\lambda_2$ implies $\lambda_1 = \lambda_2$. It establishes the distinctness of $L_T^p(\theta)$.

For sub-problem s,

$$l^s(\theta) = -\frac{T}{2}\log|A^s\Sigma_s A^{s\prime}| - \frac{1}{2}\sum_{t=1}^{T}(y_t^s - A^s\lambda^s)(A^s\Sigma_s A^{s\prime})^{-1}(y_t^s - A^s\lambda^s)'$$

Let $W^s = A^s(A^s\Sigma_s A^{s\prime})^{-1}$ with ijth element w_{ij}^s. Then the Fisher information $-\mathrm{E}(\partial^2 l^s/(\partial\lambda^s)^2)$ has entries

$$-\mathrm{E}\left(\frac{\partial^2 l^s}{\partial\lambda_i^s\partial\lambda_j^s}\right) = w_{ij}^s + \frac{1}{2}\phi^2(w_{ij}^s)^2.$$

W^s is a nonnegative definite matrix, so is matrix $\left([w_{ij}^s]^2\right)$. Therefore $-\mathrm{E}(\partial^2 L_T^p/\partial\lambda^2)$ is positive definite near θ_0, i.e., $L_T^p(\theta)$ will converge to a convex function in a neighborhood of θ_0 when $T \to \infty$, so the maximum point is almost surely unique in ω. By Theorem 1, MPLE $\tilde{\theta}_T^p$ is consistent.

Suppose the true parameter θ_0 is in the interior of the paramter space, the asymptotic normality follows by that the third derivative of of $L_T^p(\theta)$ exists and its expectation is bounded in a neighborhood of θ_0. ■

References

[1] Multicast-based Inference of Network-internal Characteristics (MINC), http://www.research.att.com/projects/minc/.

[2] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*, 36(2):192–236, 1974.

[3] J. Besag. Statistical analysis of non-lattice data. *Statistican*, 24(3):179–195, 1975.

[4] J. Cao, D. Davis, S. Vander Wiel, and B. Yu. Time-varying network tomography: router link data. *Journal of American Statistics Association*, 95(452):1063–1075, 2000.

[5] J. Cao, S. Vander Wiel, B. Yu, and Z. Zhu. A scalable method for estimating network traffic matrices. Technical report, Bell Labs, 2000.

[6] M. Coates, A. Hero, R. Nowak, and B. Yu. Internet tomography. *Signal Processing Magazine*, 19(3):47–65, 2002.

[7] M. Coates and R. Nowak. Network tomography for internal delay estimation, 2001.

[8] D. R. Cox. Partial likelihood. *Biometrika*, 62:269–276, 1975.

[9] I. Csiszár. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158, 1975.

[10] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statististical Society B*, 39:1–38, 1977.

[11] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. Deriving traffic demands for operational ip networks: Methodology and experience. *IEEE/ACM Transactions on Networking*, pages 265–279, June 2001.

[12] K. Lange. A gradient algorithm locally equivalent to the em algorithm. *Journal of the Royal Statistical Society, Series B*, 57:425–437, 1995.

[13] E. L. Lehmann and G. Casella. *Theory of Point Estimation*. Springer-Verlag, NY, 2nd edition, 1998.

[14] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley Interscience, 1997.

[15] ns (Network Simulator). http://www.isi.edu/nsnam/ns.

[16] F. Lo Presti, N.G. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal delay distributions. Technical report, AT&T Laboratories and University of Massachustts, 1999.

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

[18] H. White. *Estimation, Inference and Specification Analysis*. New York: Cambridge University Press, 1994.