

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 aggregative 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 form simple subproblems and ignore the dependences among the subproblems to form a product likelihood of the subproblems. As a result, this approach keeps a good balance between the computational complexity and the statistical efficiency of the parameter estimation. Some statistical properties of the pseudo likelihood estimator, such as consistency and asymptotic normality, are established. A pseudo expectation-maximization (EM) algorithm is developed 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— End-to-end measurement, multicast tree, network tomography, origin-destination matrix, pseudo likelihood.

I. INTRODUCTION

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 delays. Usually one cannot rely on the collaboration of individual routers and servers to measure network traffic directly; 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 network inference problems 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. The idea of

modifying likelihood is not new, and some likelihood modification methods have been proposed, e.g., pseudo likelihood [2], [3] for Markov random fields (MRF) by Besag (1974), partial likelihood [9] for hazard regressions by Cox (1973), and quasi-maximum likelihood [23] for finance by White (1994). Our method is partly motivated by the pseudo likelihood method used in solving MRF problems by Besag: both pseudo likelihood functions are constructed by taking a product of the likelihood of sub-problems with smaller and simpler dependence structures between variables of interest instead of the global complex dependence. The key difference between Besag's method and ours is how to form sub-problems. Besag's pseudo likelihood is based on the neighborhood conditional likelihood decomposition, while we use the product of the marginal likelihoods from the subproblems.

This paper is organized as follows. In Section II, we introduce a general model for network tomography problems, for which a pseudo likelihood approach is proposed. In Section III, the pseudo likelihood approach is applied to two examples of the general network tomography model: (1) multicast internal delay distribution inference through end-to-end delay measurements; (2) origin-destination matrix inference through link traffic counts. Simulated and real data sets are used to illustrate the proposed approach for these two examples. Proofs of Theorems 3 and 4 are given in the appendix.

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).

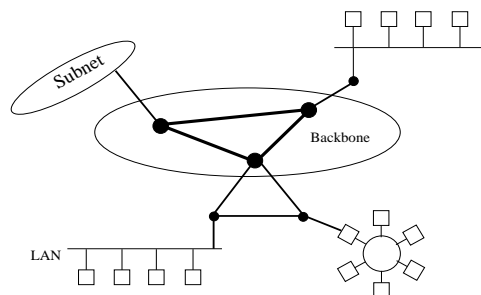


Fig. 1. An illustration of Internet topology

Manuscript received July 29, 2003. This work was supported in part by NSF Grants FD98-02314, DMS-9803063, FD01-12731 and ARO grant DAAG55-98-1-0341.

Gang Liang and Bin Yu are with the Department of Statistics, University of California at Berkeley, 367 Evans Hall, Berkeley, CA 94720 USA. (fax: 510-642-7892; email: {liang,binyu}@stat.Berkeley.EDU).

Let $\mathbf{X} = (X_1, \dots, X_J)'$ be a J dimensional random vector, which reflects the network dynamics of interest, e.g., link delay, traffic flow counts at a particular time interval. Let $\mathbf{Y} = (Y_1, \dots, Y_I)'$ be an I dimensional measurement vector. The goal of the network tomography is to estimate \mathbf{X} from the observed \mathbf{Y} . As in Coates *et al.* [7], such network tomography problems can be approximately (or exactly) represented by a linear equation:

$$\mathbf{Y} = \mathbf{A}\mathbf{X}, \quad (1)$$

where \mathbf{A} is a known $I \times J$ 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, \mathbf{A} is a fixed 0-1 matrix. It is worth noting that we assume that there is no measurement or any other errors in (1) to further simplify the model.

Equation (1) reveals the aggregative nature of network measurements, and so the estimation of the distribution of \mathbf{X} is an inverse problem. In a general network tomography scenario, \mathbf{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. A 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. Furthermore, we assume that

$$X_j \sim f_j(\theta_j), \quad j = 1, \dots, J, \quad (2)$$

where f_j is a density function with θ_j being its parameter. Then the parameter of the whole model is $\theta = (\theta_1, \dots, \theta_J)$.

Throughout the paper, let $\mathbf{Y}_1, \dots, \mathbf{Y}_T$ be the observed data vectors at T consecutive time points or intervals, and $\mathbf{X}_1, \dots, \mathbf{X}_T$ be the corresponding unobserved network performance quantities of interest and Y_{it}, X_{ij} be the i th and j th element of \mathbf{Y}_t and \mathbf{X}_t respectively. $\mathbf{X}_1, \dots, \mathbf{X}_T$ (and hence $\mathbf{Y}_1, \dots, \mathbf{Y}_T$) are assumed to be identical independent distributed (i.i.d.). To deal with the nonstationary nature of the data, we use a local likelihood approach as employed in Cao *et al.* [5] by assuming that observations within a small moving time window are i.i.d.

Now we are ready to construct the pseudo likelihood for the general network tomography model specified in (1).

A. Subproblem Forming and Pseudo Likelihood Construction

To obtain the pseudo likelihood for model (1), subproblems are formed by selecting some pairs of rows from the routing matrix \mathbf{A} . In this paper, we select all possible pairs, but a subset can be judiciously chosen to reduce the computation. We first devise this approach for the multicast internal delay inference problem, which, after decomposition, is equivalent to the delay inference problem in the unicast framework [8]. We then generalize it to the general network tomography model (1), for which we will now present the construction of the pseudo likelihood.

Let S denote the set of subproblems by selecting all possible pairs of rows from the routing matrix \mathbf{A} : $S = \{s = (i_1, i_2) : 1 \leq i_1 < i_2 \leq I\}$. Then for each subproblem $s \in S$, we have

$$\mathbf{Y}^s = \mathbf{A}^s \mathbf{X}^s, \quad (3)$$

where \mathbf{X}^s is the vector of network dynamic components involved in the given subproblem s , \mathbf{A}^s is the corresponding sub-routing matrix, and $\mathbf{Y}^s = (Y_{i_1}, Y_{i_2})'$ is the corresponding observed measurement of s . For later use, we introduce two notations here:

$$\begin{aligned} \text{given } s, \quad J^s &= \{j : \text{the } j\text{th element of } \mathbf{X} \text{ is involved in } s\}; \\ \text{given } j, \quad S^j &= \{s : \text{the } j\text{th element of } \mathbf{X} \text{ is involved in } s\}. \end{aligned}$$

Let θ^s be the model parameter of the subproblem s , and $p^s(\mathbf{Y}^s; \theta^s)$ be its marginal likelihood function. Then for the observed measurement vector \mathbf{Y} , the pseudo log-likelihood function is defined as

$$L^P(\mathbf{Y}; \theta) = \sum_{s \in S} \log l^s(\mathbf{Y}^s; \theta^s), \quad (4)$$

where $l^s(\mathbf{Y}^s; \theta^s) = \log p^s(\mathbf{Y}^s; \theta^s)$ is the log-likelihood function of subproblem s . Given observed independent data vectors $\mathbf{Y}_1, \dots, \mathbf{Y}_T$, let $\mathbf{X}_1^s, \dots, \mathbf{X}_T^s$ and $\mathbf{Y}_1^s, \dots, \mathbf{Y}_T^s$, respectively, be the corresponding unobserved and observed data vectors for subproblem s . The overall pseudo log-likelihood function is then

$$L_T^P(\mathbf{Y}_1, \dots, \mathbf{Y}_T; \theta) = \sum_{t=1}^T L^P(\mathbf{Y}_t; \theta). \quad (5)$$

Maximizing the pseudo likelihood function gives the maximum pseudo likelihood estimate (MPLE) of parameter θ . Often one seeks to solve the following pseudo likelihood equation:

$$\frac{\partial}{\partial \theta} L_T^P(\mathbf{Y}_1, \dots, \mathbf{Y}_T; \theta) = 0. \quad (6)$$

To construct a pseudo likelihood function, selecting three or more rows each time may also be 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 shown later suggests that selecting two rows each time gives satisfactory results while keeping the computational cost reasonable.

B. Asymptotic Properties of MPLE

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

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

- A1) $L^P(\mathbf{Y}; \theta)$ is distinct, i.e., for any $\theta_1 \neq \theta_2$, there exists a set Δ in the sample space \mathbb{R}^I , such that $P_{\theta_0}(\Delta) > 0$ and for all $\mathbf{Y} \in \Delta$, $L^P(\mathbf{Y}; \theta_1) \neq L^P(\mathbf{Y}; \theta_2)$;
- A2) The parameter space contains an open interval ω of which the true parameter θ_0 is an interior point;
- A3) $L_T^P(\mathbf{Y}_1, \dots, \mathbf{Y}_T; \theta)$ is differentiable with respect to θ . The pseudo likelihood equation (6) has a unique solution in ω almost surely when $T \rightarrow \infty$.

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

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

Theorem 2: (Asymptotic normality) In addition to the assumptions specified in Theorem 1, suppose the following conditions are also satisfied:

- B1) $L_T^p(\mathbf{Y}_1, \dots, \mathbf{Y}_T; \theta)$ is twice continuously differentiable with respect to θ . In addition, expectation and differentiation operations of $L_T^p(\mathbf{Y}_1, \dots, \mathbf{Y}_T; \theta)$ can be inter-exchanged;
- B2) With probability approaching 1, the Hessian matrix $\nabla^2 L_T^p(\mathbf{Y}_1, \dots, \mathbf{Y}_T; \theta)$ is invertible in a small open interval ω around the true parameter θ_0 as $T \rightarrow \infty$;
- B3) For the same ω , $\nabla^2 L_T^p(\mathbf{Y}_1, \dots, \mathbf{Y}_T; \theta) \rightarrow H(\theta)$ in distribution uniformly for $\theta \in \omega$.

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

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

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

Since almost identical proofs can be found in [15] or [23], proofs of Theorems 1 and 2 are omitted. In the later sections, direct applications of Theorems 1 and 2 lead to Theorems 3 and 4. In their proofs, the following observations from [23] are used to check the conditions in Theorems 1 and 2: 1) The uniform convergence of $\nabla^2 L_T^p(\mathbf{Y}_1, \dots, \mathbf{Y}_T; \theta)$ to $H(\theta)$ in the open neighborhood ω of θ_0 can often be verified by checking that $E_{\theta_0} \frac{\partial^3 L_T^p}{\partial \theta_j \partial \theta_k \partial \theta_l}$ is bounded in ω , which holds for most analytic functions L_T^p ; 2) With necessary regularity conditions, the invertibility of the Hessian matrix $\nabla^2 L_T^p(\mathbf{Y}_1, \dots, \mathbf{Y}_T; \theta)$ is ensured by the convexity of $E \nabla^2 L_T^p(\mathbf{Y}; \theta)$ at θ_0 for both multicast and OD inference problems.

C. Pseudo EM Algorithm

Maximizing the pseudo function leads to the MPLE, but often the pseudo likelihood equation (6) cannot be solved analytically; hence, a numerical optimization algorithm has to be adopted. The EM algorithm [11] is a well-known method for maximizing the likelihood function numerically. We use an pseudo-EM (an EM like algorithm) to maximize our pseudo likelihood functions.

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

$$Q(\theta, \theta^{(k)}) = \sum_{s \in S} \sum_{t=1}^T E_{\theta^{(k)}}(l^s(\mathbf{X}_t^s; \theta^s) | \mathbf{Y}_t^s), \quad (7)$$

which is obtained by assuming the independence of subproblems in the expectation step. As holds for the EM algorithm,

we have the following result for the pseudo-EM algorithm: during pseudo-EM iteration steps, the pseudo log-likelihood function $L_T^p(\mathbf{Y}_1, \dots, \mathbf{Y}_T; \theta)$ is non-decreasing; moreover, if L_T^p is unimodal, then the pseudo-EM algorithm will converge to the unique maximum point.

One needs to note that even though Theorem 1 and 2 guarantee the consistency and asymptotic normality of MPLE, the pseudo-EM algorithm may fall into a local maxima and hence not converge to the global maximum point. This problem occurs with any EM-like algorithms. On the other hand, for the two examples below, the proposed pseudo-EM algorithm is well-behaved, i.e., the algorithm actually finds the global maxima in most of the simulation and real data experiments.

III. APPLICATIONS OF PSEUDO LIKELIHOOD ESTIMATION

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 can be found in [8], but unicast methods suffer from scalability issues when large networks are of interest. Here we use the multicast-based external approach to estimate internal link delay distributions.

Consider a general multicast tree depicted in Fig. 2. Each node is labeled with a number; we adopt the convention that link i connects node i to its parental node. 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, copies of the same packet received at receiver 4 and 5 experience the same amount of delay on their common links 1 and 2. Measurements are made at end receivers, so only the aggregated delays over the paths from root to end receivers are observed.

Due to the aggregation of the measured delays, the network tomography model defined in (1) can be naturally applied to estimate the internal delay distribution. 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. 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. For the multicast tree depicted in Figure 2(a),

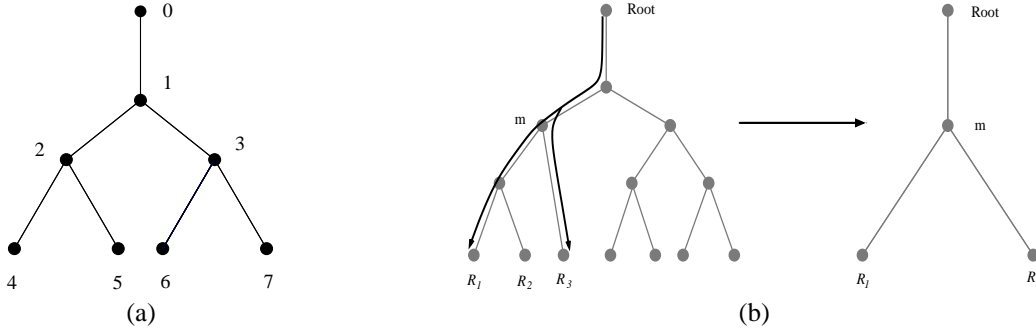


Fig. 2. (a) An arbitrary virtual multicast tree with four receivers; (b) Psuedo Likelihood Decomposition: a virtual two-leaf multicast tree is formed by only considering a pair of end receivers (R_1, R_2).

(1) can be spelled out as:

$$\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_7 \end{pmatrix},$$

where Y_1, \dots, Y_4 are the measured delays at end receivers 4, ..., 7 and X_1, \dots, X_7 are the delays over internal links ending at nodes 1, ..., 7.

Each link has a certain amount of minimal delay (overhead), which is assumed to be known beforehand. After compensating the minimal delay of each link, a discretization scheme is imposed on link-level delay by Lo Presti *et al.* (1999) such that X_j takes finite possible values $\{0, q, 2q, \dots, mq, \infty\}$, where q is the bin width and m is a constant. Assume q is known, so without loss of generality, set $q = 1$. Therefore, each \mathbf{X}_j is a independent multinomial variable with $\theta_j = (\theta_{j0}, \theta_{j1}, \dots, \theta_{jm}, \theta_{j\infty})$. 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. In order to ensure identifiability, we only consider canonical multicast trees [17] defined as those satisfying

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

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

For the problem of multicast delay inference through end-to-end traffic, the maximum likelihood method is usually infeasible, because its likelihood function involves finding all possible internal delay vectors x given an observed delay vector y , and we can show that the computational complexity grows at a non-polynomial rate. Although Lo Presti *et al.*'s recursive algorithm [17] is a computational efficient method in estimating internal delay distributions, it suffers from some performance drawbacks. But we can apply the pseudo likelihood approach to this problem. In this case, each row of the routing matrix A corresponds to an end receiver in the multicast tree, and subproblems are formed by choosing two end receivers each time as illustrated in Fig. 2(b).

1) Parameter Estimation through Pseudo-EM: For a given subproblem s , each component of \mathbf{X}^s is an independent multinomial random variable, so that the log-likelihood function given

the complete data $\mathbf{X}_1^s, \dots, \mathbf{X}_T^s$ is

$$l^s(\mathbf{X}_1^s, \dots, \mathbf{X}_T^s; \theta^s) = \sum_{j \in J^s} \sum_l n_{jl}^s \log(\theta_{jl}),$$

where $\theta_{jl} = P(X_j = l)$, $n_{jl}^s = \sum_t \mathbf{1}_{\{X_{tj}^s = l\}}$.

Let $\theta^{(k)}$ be the parameter estimate obtained in the k th step of Pseudo EM, then according to (7), the target function to be maximized in the $(k+1)$ th step is

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

E-step: compute

$$\hat{n}_{jl} = \sum_{s \in S} E_{\theta^{(k)}} \left(\sum_{t=1}^T \mathbf{1}_{\{X_{tj}^s = l\}} \middle| \mathbf{Y}_t^s \right);$$

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

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

The initial value of the pseudo-EM algorithm can be chosen arbitrarily. We find that small starting values may have difficulties to increase even when the true parameter θ_0 is not small. A uniform distribution, i.e., $\theta_{jl}^{(0)} = 1/(m+2)$ for all possible j and l , is used as the starting point for our simulations below. Such a uniform starting point gives satisfactory results.

Let P be the average number of links per path, then the overall computational complexity of each step of the pseudo-EM algorithm is $O(m^3 l^2 P^2)$. Meanwhile, specializing Theorem 1 and 2 to this example gives the consistency and asymptotic normality of the MPLE.

Theorem 3: Assume that θ_0 is in the interior of the parameter space (note that it implies that the multicast tree is a canonical tree.) Let $\hat{\theta}_T^P$ be the internal delay distribution estimate through maxiizing the pseudo likelihood, then $\hat{\theta}_T^P$ is a consistent estimator, and $\sqrt{T}(\hat{\theta}_T^P - \theta_0)$ converges in distribution to a multivariate normal random variable as $T \rightarrow \infty$.

2) Experimental results: In order to assess the performance of the pseudo likelihood methodology, model simulations are carried out on a 4-leave multicast tree depicted in Fig. 2(b). Due to its small size, the MLE method is implemented for this multicast tree, so we can compare the performance of MPLE

with that of MLE and recursive algorithm for this simulation. The initial value of the EM algorithm in implementing MLE is also set to be the uniform distribution. For each link, the bin size parameter m is set to be 14. During each simulation, 2000 i.i.d. multicast delay measurements are generated with each internal link having an independent discrete delay distribution. Fig. 3 shows the delay distribution estimates of three arbitrarily selected links along with their true delay distributions in one such experiment. The plot shows both MPLE and MLE capture most of the link delay distributions and their performance is comparable, while the recursive algorithm sometimes gives estimate far from the truth. Still we can see sometimes both MPLE and MLE have estimates quite different from the true probability. Mainly, this is due to that the number of parameters needs to be estimated, $7 \times 16 = 112$, is large, and moreover by its nature such an inverse problem is ill-posed.

Fig. 4 shows the L_1 error norm of MLE, MPLE and recursive estimate for each link, as averaged over 30 independent simulations. For each link, the L_1 error norm is simply the sum of the absolute differences between probability estimates and the true probabilities. As a common measure of the performance of density estimates, the L_1 error norm enjoys several theoretical advantages as discussed in [18]. The plot shows that MLE and MPLE have a comparable estimation performance for tracking link delay distributions, while the recursive algorithm has much larger L_1 errors on all links. Moreover, we can see that MPLE has smaller standard deviations (SD) in terms of L_1 error than MLE on all links, implying that MPLE is more robust than MLE. This is probably because the pseudo likelihood function, which is a product of less complex likelihood functions on sub-problems, has a nicer surface than the full likelihood function [4].

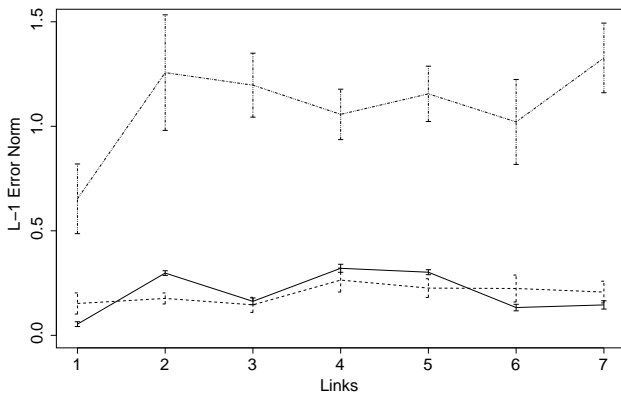


Fig. 4. Link L_1 error norm averaged over 30 simulations: solid line is MPLE, dashed line is MLE, and dotted line is recursive algorithm. For each link, the vertical bar shows the SD of L_1 error norm for the given link.

B. Example II: Origin-Destination Matrix Inference

The origin-destination (OD) matrix, whose elements are the byte counts of traffic between all origin-destination pairs in a network, is a key input to any routing algorithm, since the link weights of the Open Shortest Path First (OSPF) routing protocol are related to the traffic on the path. Two approaches exist to carry out the OD matrix estimation: direct and indirect. The

direct approach via router softwares, such as *Netflow* supported by Cisco routers, is described in [12]. Indirect approaches use link traffic counts at router interfaces, which are much easier to obtain. Hence, indirect methods based on the observed link counts at router interfaces have been more intensively studied ([22], [21], [19]).

Vardi [22] was the first to investigate this problem. Assuming i.i.d. Poisson distributions for the OD traffic byte counts on a general network topology, he specifies the identifiability of the Poisson model and develops an EM algorithm to estimate Poisson parameters in both deterministic and Markov routing schemes. In order to ease the difficulty in implementing the EM algorithm, he proposes a moment estimation method and briefly discusses the normal model as an approximation to the Poisson model. Two related works treat the special case involving a single set of link counts: Vanderbei and Iannone (1994) apply the EM algorithm; Tebaldi and West (1998) is a follow-up on Vardi (1996) with a Bayesian perspective and a Markov Chain Monte Carlo implementation.

With real network data, Cao *et al.* [5] revise the Poisson assumption and propose a normal model. They also address the non-stationarity of the network OD traffic. Their methodology is validated through comparisons with directly collected (but expensive) OD traffic in a simple network at Lucent Technologies depicted in Fig. 5. Our pseudo likelihood method is based on their normal model. Next we briefly review the normal model and their methodologies in OD matrix inference.

The network tomography model specified by (1) is applicable to the OD matrix inference through link traffic counts. Recall that $\mathbf{Y} = (Y_1, Y_2, \dots, Y_I)'$ is the vector of observed traffic byte counts measured on each link interface during a given time interval and $\mathbf{X} = (X_1, X_2, \dots, X_J)'$ is the corresponding vector of unobserved true OD traffic byte counts at the same time period. \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. Each column of A is associated with an OD pair and indicates which links are used to carry traffic for that pair of nodes, while each row corresponds to an observed link traffic. For instance, the routing matrix A (0 for entries not shown) of the single-router network depicted in Fig. 5(b) is

$$A = \begin{matrix} \text{orig-1} \\ \text{orig-2} \\ \text{orig-3} \\ \text{orig-4} \\ \text{dest-1} \\ \text{dest-2} \\ \text{dest-3} \\ \text{dest-4} \end{matrix} \begin{pmatrix} 1 & 1 & 1 & 1 & & & & \\ & & & 1 & 1 & 1 & 1 & \\ & & & & & 1 & 1 & 1 & 1 \\ & & & & & & 1 & 1 & 1 & 1 \\ 1 & & & 1 & & 1 & & 1 & & \\ & 1 & & & 1 & & 1 & & 1 & \\ & & 1 & & 1 & & 1 & & 1 & \\ & & & 1 & & 1 & & 1 & & 1 \end{pmatrix},$$

where the first column of A indicates that two links (rows of A) *orig1* and *dest1* contain the OD traffic from *fddi* to itself. The first row of A means that the traffic originated from node *fddi* consists of 4 pairs of unobserved OD pair traffic originated from *fddi* and destined to all other 4 nodes (including itself).

In the normal model [5], each component of \mathbf{X} is assumed to be independent normally distributed, satisfying the following mean-variance relationship: $X_j \sim N(\lambda_j, \phi \lambda_j^c)$ independently,

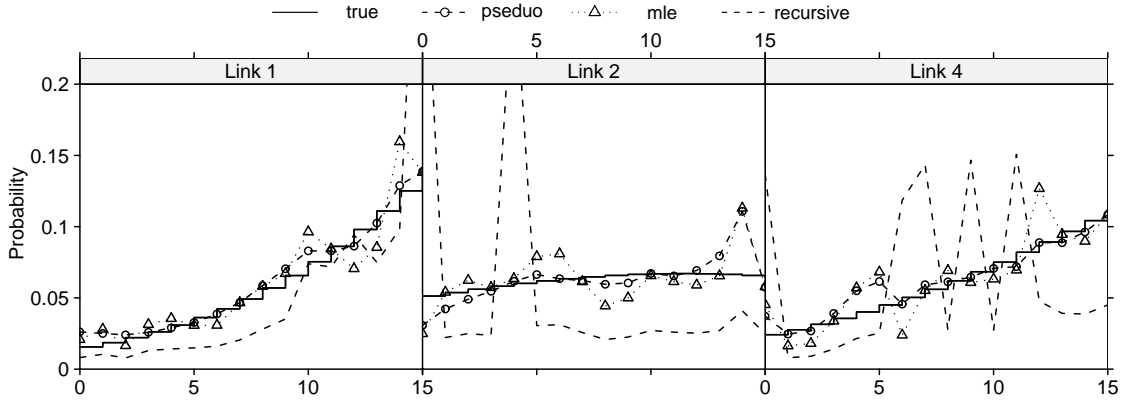


Fig. 3. Delay distribution estimates of 3 arbitrarily selected internal links: Link 1, Link 2 and Link 4. Solid step function is the true distribution, dash line with circle is MPLE, dotted line with triangle is MLE, and dash line only is recursive estimate.

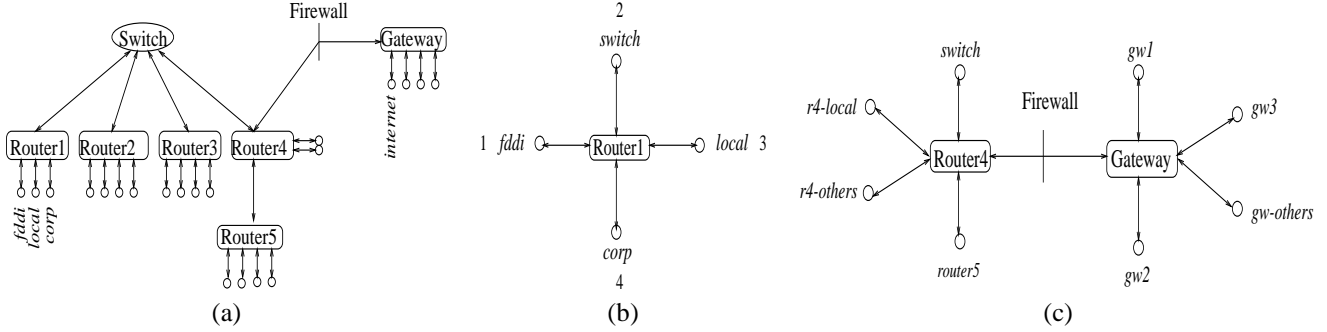


Fig. 5. (a) A Simple Router Network at Lucent Technologies, (b) Network Topology around Router 1, (c) A two-router Network around Router 4 and Gateway.

where ϕ is a positive scalar applicable to all OD pairs and c is a power constant. It implies that

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

where $\lambda = (\lambda_1, \dots, \lambda_J)$ and $\Sigma = \phi \text{diag}(\lambda_1^c, \dots, \lambda_J^c)$. So the parameter of the full model is $\theta = (\phi, \lambda)$.

The mean-variance relationship is a crucial assumption to ensure the identifiability of the normal model. It implies that an OD pair with large traffic byte counts tends to have large variance with the same scale factor ϕ . For the power constant c , both $c = 1$ and 2 work well with the Lucent network data as shown in [5], [6]. Because $c = 1$ or $c = 2$ give similar results, in this paper, we use $c = 1$ as done in [6]. But note that the pseudo likelihood method we will propose later can deal with $c = 2$ without any additional technical difficulties.

Cao *et al.* [5] also address the non-stationarity of network traffic by a local likelihood model, i.e., for any given time interval t , analysis is based on observations within a symmetric window of size w around t . Within each window, observations are assumed to be independent. Maximum likelihood estimate is carried for each window via a combination of EM algorithm and a second-order optimization routine. In order to estimate the underlying true OD traffic x , the conditional expectation $E_{\hat{\theta}}(\mathbf{X}_t | \mathbf{Y}_t, \mathbf{X}_t > 0)$ is computed as an initial estimate of x . Then an iterative proportional fitting (IPF) algorithm [10], widely used in contingency table analysis to adjust table to match the observed margins, is employed to enforce the linear constraint $y = Ax$ to obtain the final estimate of OD traffic x . In [5], to smooth the parameter estimate, a random walk model is applied

to the parameter λ 's and ϕ over the sliding time windows.

For the normal model, the computational complexity is high. Let n be the number of edge nodes in a network, as discussed in [6], the computational complexity of the full MLE is $O(n^5)$ after exploiting sparse matrix calculation. Hence, the pseudo likelihood approach is applied to the above normal model. Comparisons will be made between full and pseudo likelihood with respect to the computational complexity and estimation efficiency in later sections.

1) *Parameter Estimation through Pseudo-EM:* For each subproblem s , \mathbf{Y}^s is the vector of observed traffic byte counts and \mathbf{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 subproblem s is then $\theta^s = (\phi, \lambda^s)$. The log-likelihood function for subproblem s given complete data $\mathbf{X}_1^s, \dots, \mathbf{X}_T^s$ is

$$l^s(\mathbf{X}_1^s, \dots, \mathbf{X}_T^s; \theta^s) = -\frac{T}{2} \log |\Sigma_s| - \frac{1}{2} \sum_{t=1}^T (\mathbf{X}_t^s - \lambda^s)' \Sigma_s^{-1} (\mathbf{X}_t^s - \lambda^s).$$

Let $\theta^{(k)}$ be the parameter estimate in the k th step. According to (7), the objective function to be maximized in the $(k+1)$ th step is

$$Q(\theta, \theta^{(k)}) \propto - \sum_{s \in S} \left\{ T \left(\log |\Sigma_s| + \text{tr}(\Sigma_s^{-1} R^{s(k)}) \right) + \sum_{t=1}^T (m_t^{s(k)} - \lambda^s)' \Sigma_s^{-1} (m_t^{s(k)} - \lambda^s) \right\} \quad (8)$$

where $m_t^{s(k)} = E_{\theta^{s(k)}}(\mathbf{X}_t^s | \mathbf{Y}_t^s)$ and $R^{s(k)} = \text{Var}_{\theta^{s(k)}}(\mathbf{X}_t^s | \mathbf{Y}_t^s)$. We have

$$\begin{aligned} m_t^{s(k)} &= \lambda^{s(k)} + \Sigma_s^{(k)} A^{s'} (A^s \Sigma_s^{(k)} A^{s'})^{-1} (\mathbf{Y}_t^s - A^s \lambda^{s(k)}), \\ R^{s(k)} &= \Sigma_s^{(k)} - \Sigma_s^{(k)} A^{s'} (A^s \Sigma_s^{(k)} A^{s'})^{-1} A^s \Sigma_s^{(k)}, \end{aligned}$$

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

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

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

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

$$\lambda_j^2 + \phi \lambda_j - a_j^{(k)} = 0, \quad j = 1, \dots, J \quad (9a)$$

$$\sum_{j=1}^J (\lambda_j - b_j^{(k)}) = 0. \quad (9b)$$

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

M-step: Solve (8). Equation (8a) shows quadratic constraint between λ_j and ϕ . In conjunction with these solutions, (8b) 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 [14]) is employed to solve these equations only roughly.

The starting point $\theta^{(0)} = (\lambda^{(0)}, \phi^{(0)})$ can be quite arbitrary for the pseudo-EM algorithm. For the OD matrix inference experiments below, we adopt the same initial value used in [5], such that $\lambda^{(0)}$ is a constant vector with each component to be $\sum_{t=1}^T \mathbf{1}' \mathbf{Y}_t / (T \mathbf{1}' \mathbf{A} \mathbf{1})$ and $\phi^{(0)} = \text{Var}(Y_{ti}) / E(Y_{ti})$, 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 unidirectional links I is $O(n)$, and the number of OD pairs J is n^2 . Assuming that the average number of links between an OD pair is $O(\sqrt{n})$, it can be shown that on average each subproblem involves $O(n^{1.5})$ OD traffic pairs, thus for the n^2 subproblems, the overall computational complexity of each step of the pseudo-EM algorithm is $O(n^{3.5})$. Compared with the complexity of the full likelihood $O(n^5)$, the pseudo likelihood approach reduces the computational complexity considerably. Moreover, the pseudo likelihood approach fits in the framework of distributed computing, which can be advantageous for practical applications. Therefore, the pseudo likelihood approach is

more scalable to larger networks. As future research, we plan to use only a subset of subproblems in parameter estimation to further reduce the computational complexity.

Theorem 4: 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 $\hat{\theta}_T^p$ be the parameter estimate through pseudo likelihood, then $\hat{\theta}_T^p$ is a consistent estimator and when $T \rightarrow \infty$, $\sqrt{T}(\hat{\theta}_T^p - \theta)$ converges in distribution to a multivariate normal random variable.

2) Experimental results: Two types of experiments are conducted to assess the performance of the pseudo likelihood method. First, we consider a small real network at Lucent Technologies depicted in Fig. 5(b). For this 4-node network, there are 4 incoming and 4 outgoing links and the total number of OD pairs J is 16. For this network, the true OD traffic counts are collected through *Netflow* in every 5 minutes, so we can compare the estimated traffic counts (derived from the parameter estimates) with the true traffic counts. In order to have good comparison, we use the data collected on Feb 22, 1999, the same day used in [5], which consists of 288 data points. We adopt the same window size 11 from Cao *et al.* for the local moving i.i.d. model to capture the time varying nature of the network traffic. Fig. 6 shows the MPLE of λ , the mean OD traffic, along with its MLE estimate. For comparison, the 11-points moving average of true OD traffic counts are also plotted. We can see that both pseudo and full likelihood methods capture the dynamics of the OD traffic counts quite well.

For estimating the actual time-varying OD traffic counts \mathbf{X}_t , 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 20 is shown in Fig. 7. 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 [13] on a 1G Hz laptop. In producing Fig. 6, it takes about 12 seconds for computing the MPLE, and about 49 seconds for the MLE. In the pseudo likelihood approach, the computation of coefficients $a_i^{(k)}$ and $b_i^{(k)}$ is done by C codes because the performance of R will be severely affected by multiple loops introduced by dealing with numerous subproblems. Similarly, 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 the introduction of C codes in the E-step of the full likelihood method will barely speed up its execution.

Second, in order to assess the performance of MPLE more thoroughly, simulations are carried out on some larger networks through network simulator (NS) [16]. OD traffic is simulated on a two-router network depicted in Fig. 5(c). Every link is configured to have a capacity of 1.5Mbps and a propagation delay of 10ms with default drop tail queueing management. Each OD pair is assigned a bit rate; during the simulation, the node generates exponential on-off UDP packets to the destination node



Fig. 6. Mean OD traffic estimates $\hat{\lambda}_c$ obtained from pseudo and full likelihood methods for 4 nodes network around Router1 against the moving average of the true OD traffic c . Marginal traffic c is shown in side panels.

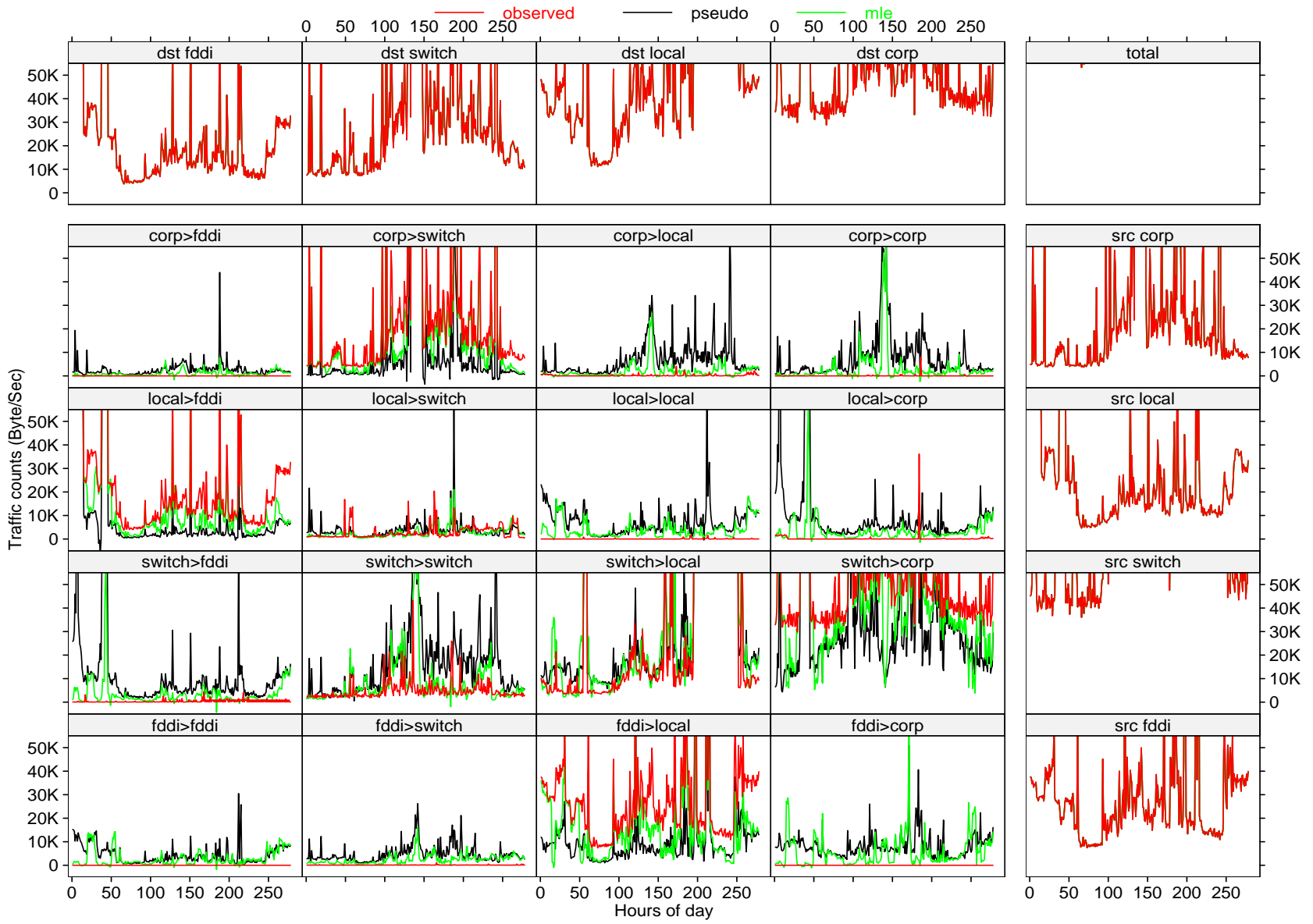


Fig. 7. OD traffic counts estimates, obtained from pseudo and full likelihood methods against the true OD traffic counts for 4 nodes network around Router1. The plot has been zoomed in by a factor of 20 to show small-scale features.

with the given rate. The burst time of the generating process is set to be 20ms and idle time 120ms. Background traffic consists of a mixture of exponential or pareto on-off sources using TCP or UDP. Each background connection is randomly attached to an OD pair and has random bit rate and start stop times. Flow monitor is utilized to collect flow, i.e., OD traffic statistics. The simulation lasts 288 seconds: every second represents a 5-minute interval in a 24 hours long data collection. In Fig. 8, we show the results of byte count estimates on 9 selected OD pairs (from 3 nodes to another 3 nodes). Those pairs are chosen because they have larger estimation errors compared with other pairs. From the plot, we see that both pseudo and full likelihood methods capture the dynamics of the simulated OD traffic under the zoomed-in scale. It takes about 18 seconds to compute the MPLE, while it takes around 88 seconds to compute the MLE.

An even larger simulation is carried out on the Lucent network illustrated in Fig. 5(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 losing much estimation performance, so it is more scalable to larger networks.

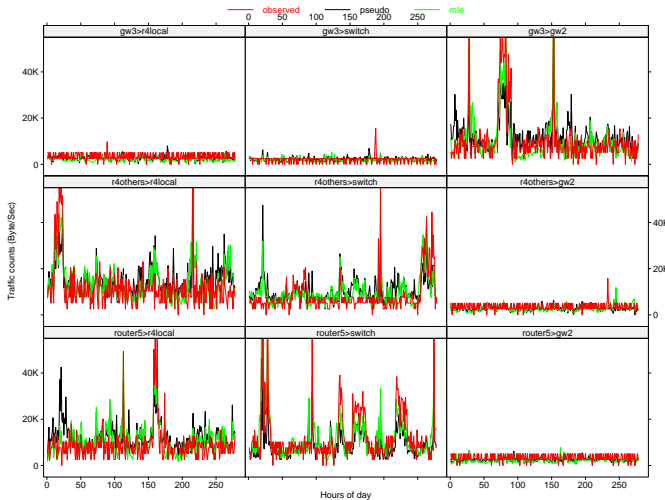


Fig. 8. OD traffic count estimates on selected OD pairs for NS simulated data on a two-router network depicted in Fig. 5(c). Each panel shows the pseudo and full likelihood methods against the observed simulated OD traffic.

IV. CONCLUSIONS

In this paper, we proposed a pseudo likelihood approach to the network tomography problem and included two special cases (multicast link delay estimation and OD traffic estimation) to demonstrate the potential of the proposed approach. In the two special cases, the MPLE shows strengths through its estimation efficiency and manageable computational complexity. Even though the basic idea of divide-and-conquer is not new, it 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 beyond the two special cases demonstrated here.

ACKNOWLEDGMENTS

We would like to thank two anonymous referees for their constructive comments on the previous versions of this paper. We also would like to thank Jonathan Grib, Mike Last, and Dave Graham-Squire for very helpful comments on the presentation of the paper. Last, but not least, we thank Jin Cao and Scott Vander Wiel for sharing the collected traffic data on the Lucent network and their Splus codes.

APPENDIX

For notational convenience, given observed $\mathbf{Y}_1, \dots, \mathbf{Y}_T$, let $L_T^p(\theta)$ be a shorthand for the pseudo log-likelihood function $L_T^p(\mathbf{Y}_1, \dots, \mathbf{Y}_T; \theta)$, and $l^s(\theta^s)$ for $l^s(\mathbf{Y}_1^s, \dots, \mathbf{Y}_T^s; \theta^s)$.

Proof of Theorem 3: We proceed by verifying the conditions in Theorems 1 and 2. The true parameter θ_0 is assumed in an open interval ω of the parameter space (condition A2). It is easy to show that the pseudo likelihood function $L^p(\mathbf{Y}, \theta)$ is infinitely differentiable, and its expectation and derivative operations can be inter-exchanged (condition B1). If we can show the distinctiveness of $L^p(\mathbf{Y}, \theta)$ (condition A1) and the pseudo log-likelihood function L_T^p is strictly convex in a neighborhood of the true parameter θ_0 as $T \rightarrow \infty$ (condition B2), then we can conclude that the pseudo likelihood equation (6) has a unique solution in a neighborhood of θ_0 almost surely (condition A3). Therefore, the MPLE is consistent by Theorem 1. Also because θ_0 is in the interior of the parameter space, $\frac{\partial^3 L_T^p}{\partial \theta_i \partial \theta_k \partial \theta_l}$ is continuous hence bounded in a neighborhood of true parameter θ_0 and so is its expectation (condition B3). It follows the normality of the MPLE by Theorem 2.

First, in order to proof the distinctiveness of $L^p(\mathbf{Y}, \theta)$, we need to show that if

$$L^p(\mathbf{Y}, \theta_1) = L^p(\mathbf{Y}, \theta_2) \quad (10)$$

holds for all possible values of \mathbf{Y} , then $\theta_1 = \theta_2$. Taking expectation with respect to P_{θ_1} , Equation (9) becomes the sum of Kullback-Leibler (KL) divergences over subproblems being zero. So each term is zero and hence the log-likelihood is equal for each subproblem: $l^s(\mathbf{Y}^s, \theta_1^s) = l^s(\mathbf{Y}^s, \theta_2^s)$. For each subproblem, it is easy to show that its three virtual sub-paths, e.g., $\text{Root} \rightarrow m$, $m \rightarrow R_1$ and $m \rightarrow R_3$ in Fig. 2(b), have the same delay distribution when the true parameter is θ_1 or θ_2 . Now we visit all sub-paths from top to down to show that each internal link has the same distribution under θ_1 and θ_2 . For instance, consider the multicast tree depicted in Fig. 2(a). Link 1 has the same distribution under θ_1 and θ_2 because it is a sub-path of $s = (1, 4)$ (subproblem formed by end nodes 4 and 7). Also we know that sub-path $0 \rightarrow 2$ has the same distribution under θ_1 and θ_2 by considering subproblem $s = (1, 2)$. Using the fact that the delay distribution for each sub-path is the convolution of its sub-link distributions, we know Link 2 has the same distribution by deconvolution. Repeating this process, we see that all

internal links have the same distribution under θ_1 or θ_2 . Hence we prove the distinctiveness of $L_T^p(\theta)$.

Second, we want to show $L_T^p(\theta)$ is strictly convex in a neighborhood of the true parameter θ_0 with probability approaching 1 when T goes to infinity. Because $\mathbf{Y}_1, \dots, \mathbf{Y}_T$ are i.i.d. copies of \mathbf{Y} , we only need to prove that $EL^p(\mathbf{Y}; \theta)$ is strictly convex at θ_0 , i.e., the Hessian matrix $-\text{EV}^2 L^p(\mathbf{Y}; \theta_0) = -\text{EV}^2 L^p(\mathbf{Y}; \theta)|_{\theta=\theta_0}$ is positive definite. Note that $l^s(\mathbf{Y}^s; \theta^s)$ is the true log-likelihood function of subproblem s with θ_0^s being its true parameter, we have $\text{EV} l^s(\mathbf{Y}^s; \theta_0^s) = 0$, and that

$$-\text{EV}^2 l^s(\mathbf{Y}^s; \theta_0^s) = \text{Var} \nabla l^s(\mathbf{Y}^s, \theta_0^s)$$

is a non-negative definite matrix. By observing that $L^p(\mathbf{Y}; \theta)$ is a sum of log-likelihood functions of subproblems, we have $\text{EV} L^p(\mathbf{Y}; \theta_0) = 0$ and the Hessian matrix $-\text{EV}^2 L^p(\mathbf{Y}; \theta_0)$ is non-negative definite. What remains to be shown is that it is actually positive definite. Suppose there is a vector α , such that

$$0 = -\alpha' \text{EV}^2 L^p(\mathbf{Y}; \theta_0) \alpha = \sum_{s \in S} \text{Var}(\alpha' \nabla l^s(\mathbf{Y}^s, \theta_0^s))$$

then $\alpha' \nabla l^s(\mathbf{Y}^s, \theta_0^s) = 0$ for any possible values of \mathbf{Y} and s . Similarly we enumerate through all subproblems from top to down to show $\alpha = 0$. For instance, consider the multicast tree depicted in Fig. 2(a) again. For subproblem $s = (1, 4)$, $\alpha' \nabla l^s(\mathbf{Y}^s, \theta_0^s) = 0$ implies that the components of α corresponding to Link 1 are all 0. Based on this information, that $\alpha' \nabla l^s(\mathbf{Y}^s, \theta_0^s) = 0$ for subproblem $s = (1, 2)$ implies the components of α corresponding to Link 2 are all 0. We repeat this process to show that all components of α are zero, i.e., the Hessian matrix is positive definite. ■

Proof of Theorem 4: For the OD traffic matrix inference problem, $\theta = (\phi, \lambda)$. Similar to Theorem 3, we only need to show the distinctiveness of $L_T^p(\theta)$ and its convexity in a neighborhood of θ_0 . The same KL-divergence argument in the proof of Theorem 3 shows $L_T^p(\theta_1) = L_T^p(\theta_2)$ if and only if $l^p(\theta_1^s) = l^p(\theta_2^s)$ for all s , so for distinctiveness, it is enough to show that $l^s(\theta_1^s) = l^s(\theta_2^s)$ for all sub-problem s implies $\theta_1 = \theta_2$. For a sub-problem $s = (i_1, i_2)$, let B^s be a $3 \times J$ matrix, with the i_1 th, i_2 th rows of A being the first two rows and their component-wise product being the third row. Then $l^s(\theta_1^s) = l^s(\theta_2^s)$ implies

$$\phi_1 = \phi_2, B^s \lambda_1 = B^s \lambda_2.$$

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

Second, we want to show that $L_T^p(\theta)$ is strictly convex in a neighborhood of true parameter θ_0 , i.e., the Hessian matrix $-\text{EV}^2 L^p(\mathbf{Y}; \theta_0)$ is positive definite. Let $\alpha = (a, b)$, where a is a scalar and b is a J -dimensional vector. We can show that $\alpha \text{EV}^2 L^p(\mathbf{Y}; \theta_0) \alpha' = 0$ is equivalent to $\alpha \nabla l^p(\mathbf{Y}^s; \theta_0^s) = 0$ for all possible values of \mathbf{Y} and subproblem s . For each subproblem s ,

$$l^s(\mathbf{Y}^s, \theta^s) = -\frac{1}{2} \log |A^s \Sigma A^{s'}| - \frac{1}{2} (\mathbf{Y}^s - A^s \lambda) (A^s \Sigma A^{s'})^{-1} (\mathbf{Y}^s - A^s \lambda)',$$

and $\alpha \nabla l^p(\mathbf{Y}^s; \theta_0^s) = 0$ can be simplified as

$$a = 0 \text{ and } B^s b = 0.$$

Thus, we have $a = 0$ and $Bb = 0$. Once again, that B is of full column rank implies $b = 0$, i.e., $\alpha = 0$. It follows that the pseudo log-likelihood function $L_T^p(\theta)$ converges to a convex function in a neighborhood of θ_0 as $T \rightarrow \infty$. ■

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] David Blackwell. Approximate normality of large products. Technical report, University of California at Berkeley, 1973.
- [5] 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.
- [6] J. Cao, S. Vander Wiel, B. Yu, and Z. Zhu. A scalable method for estimating network traffic matrices. Technical report, Bell Labs, 2000.
- [7] M. Coates, A. Hero, R. Nowak, and B. Yu. Internet tomography. *Signal Processing Magazine*, 19(3):47–65, 2002.
- [8] M. Coates and R. Nowak. Network tomography for internal delay estimation, 2001.
- [9] D. R. Cox. Partial likelihood. *Biometrika*, 62:269–276, 1975.
- [10] I. Csiszár. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158, 1975.
- [11] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statistical Society B*, 39:1–38, 1977.
- [12] 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.
- [13] Ross Ihaka and Robert Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996.
- [14] K. Lange. A gradient algorithm locally equivalent to the em algorithm. *Journal of the Royal Statistical Society, Series B*, 57:425–437, 1995.
- [15] E. L. Lehmann and G. Casella. *Theory of Point Estimation*. Springer-Verlag, NY, 2nd edition, 1998.
- [16] ns (Network Simulator). <http://www.isi.edu/nsnam/ns>.
- [17] 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 Massachusetts, 1999.
- [18] D.W. Scott. *Multivariate Density Estimation: Theory, Practice and Visualization*. Wiley, New York, 1992.
- [19] C. Tebaldi and M. West. Bayesian inference on network traffic using link count data (with discussion). *Journal of American Statistics Association*, pages 557–576, June 1998.
- [20] Yolanda Tsang, Mark Coates, and Robert Nowak. Nonparametric internet tomography. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Orlando, Florida, May 2002.
- [21] R. J. Vanderbei and J. Iannone. An EM approach to OD matrix estimation. Technical report, Princeton University, 1994. SOR 94-04.
- [22] Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the American Statistical Association*, 91:365–377, 1996.
- [23] H. White. *Estimation, Inference and Specification Analysis*. New York: Cambridge University Press, 1994.