Pseudo Likelihood Estimation in Network Tomography

*Gang Liang and *Bin Yu
Department of Statistics
Unversity of California at Berkeley
E-mail: {liang,binyu}@stat.Berkeley.EDU

***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 form simple subproblems and construct a product of marginal likelihood of subproblems by the ignoring their dependences. As a result, it 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) internal link delay distribution inference 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 link loss and packet delay. Usually one cannot rely on the collaboration of individual routers or servers to measure network traffic directly; estimation of 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, a unified pseudo likelihood method is proposed for a group of network tomography problems. The idea of modifying likelihood actually is not new, and some likelihood modification methods have already been proposed, e.g., pseudo likelihood [1], [2] in Markov random fields (MRF) by Besag (1974), partial likelihood [3] in hazards regression by Cox (1973), and quasi-maximum likelihood [4] in finance by White (1994). Our method is partly motivated by the pseudo likelihood method in solving MRF problems by Besag: both pseudo likelihood functions are constructed by focusing on smaller and simpler local dependence structures instead of the global complex one. Sub-problems are formed by considering variables involved in such local structures. Usually subproblems 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 subproblems. Besag's pseudo likelihood is based on the neighborhood conditional likelihood decomposition. While the problem of inferencing internal link delay distributions through multicast end-to-end traffic can be viewed in his MRF framework, all nodes are fully connected; therefore, the neighborhood decomposition scheme is no longer advantageous here, and further dependence simplification is necessary.

This paper is organized as follows. In Sec. II, we introduce a general network tomography model, and two examples are used to illustrate the general model: (1) internal link delay distribution inference 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 general network tomography model. Finally, in Sec. IV, we apply the pseudo likelihood approach to the above two examples. Proofs of Theorems 4 and 5 can be found in 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).

Let $\mathbf{X} = (X_1, ..., X_J)^t$ be a J dimension random variable vector, which reflects the network dynamics of interest, e.g., packet link delay, traffic flow counts. Let $\mathbf{Y} = (Y_1, ..., Y_I)^t$

*This research is supported in part by NSF Grants FD98-02314, DMS-9803063, FD01-12731 and ARO grant DAAG55-98-1-0341.

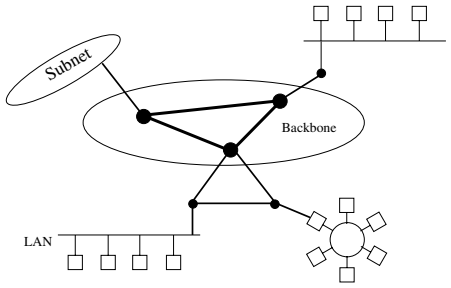

Fig. 1. An illustration of Internet topology

be an I dimension measurement vector. Generally, there is a linear relationship between observable $\mathbf{Y}$ and unobservable $\mathbf{X}$. As in Coates *et al.* [5], such network tomography problems can be approximately (or exactly) represented by a linear model:

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

where A is a known $I \times J$ routing matrix, determined by the network topology and routing tables at each router. In this paper, we restrict ourselves to fixed routing schemes, so A is a 0-1 matrix. It is worth noting that we assume there is no measurement or any other errors in (1) to further simplify the model.

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

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

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

Throughout the paper, let $y_1, ..., y_T$ be the independent observed data vectors at T consecutive time points or intervals and $x_1, ..., x_T$ 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. (However, it is worth noting that, if necessary, we could use the local likelihood approach as employed in Cao *et al.* [6] to deal with the nonstationary nature of the data. In that approach, the data are assumed iid within a small time window.) Next we will use two concrete examples to illustrate the above setup.

A. Example: Multicast Internal Delay Inference

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 measures the network link delays at link-level interfaces directly, while the external approach monitors delays through end-to-end measurements. The Multicast-based Inference of Network-internal Characteristics (MINC) Project [7] pioneered the use of multicast probing for network delay distribution estimation. The use of end-to-end measurement through multicast probing is due to the limitations of the internal approach: the collaboration between internal routers is not always available, and an extra heavy load burden will be imposed by the probing process. A similar approach through unicast end-to-end measurements [8] can be found in Coate *et al.* (2001). 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 node is labeled with a number, and we adopt a notation 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 receivers 4–7. For any pair of receivers, packets experience the same amount of delay over their common path. For instance, copies of the same packet received by receiver 4 and 5 experience the same amount of delay on link 1 and 2. Measurements are made at end receivers, so only aggregated delays over the paths from root to end receivers are observed.

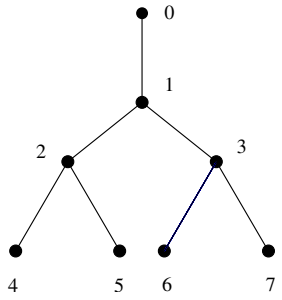

Fig. 2. An arbitrary virtual multicast tree with four receivers. Link i connects node i with its parental node, e.g., link 4 is the link connects node 4 and 2.

Due to aggregation of measured delays, the network tomography model defined by (1) and (2) can be naturally applied to the multicast internal delay distribution inference 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. 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. As an example, for the multicast tree depicted in Fig. 2, (1) becomes

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_7 \end{pmatrix},$$

where $y_1, ..., y_4$ are the measured delays at end receivers $4, ..., 7$ and $x_1, ..., x_7$ are the delays over links 1,..., 7.

Each link has a certain amount of minimal delay (overhead), which are assumed to be known beforehand. After compensating the minimal delay for each link, a discretization scheme is imposed on link-level delay by Lo Presti *et al.* (1999), such that X_j can only take finite possible values $\{0, q, 2q, ..., mq, \infty\}$,

where q is the bin width and m is a constant. Assume q is known, without loss of generality, set $q = 1$. Therefore, each X_j is an independent multinomial variable with parameter $\theta_j = (\theta_{j0}, \theta_{j1}, ..., \theta_{jm}, \theta_{j\infty})$. 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 [9] defined as those satisfying

$$P(X_j = 0) > 0,\ j = 1, ..., J,$$

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

For the problem of multicast internal delay inference, the maximum likelihood method is usually infeasible for realistic size networks, because the likelihood function involves finding all possible internal delay vectors x, which can account for each observed delay vector y. We can show that the computational complexity grows at a non-polynomial (NP) rate. Lo Presti *et al.*'s recursive algorithm [9] is a computational efficient method in estimating internal delay distributions, but it suffers from some performance drawbacks. In later section, we will apply a pseudo likelihood approach to the multicast delay distribution inference problem, and compare its performance with that of recursive algorithm in terms of estimation precision.

B. Example: Origin-Destination Matrix Inference

Origin-destination (OD) tomography quantifies path-level network dynamics through viable network measurements. OD matrix estimation is very important because it is a key input to many routing algorithms, e.g., it can be used to assign appropriate link weights in the Open Shortest Path First (OSPF) routing protocol. Two types of approaches are used to carry out the OD matrix inference: direct and indirect. Direct approaches are made possible via some router software such as *Netflow* supported by Cisco routers [10]. Indirect approaches use link traffic counts at router interfaces, which are much easier to obtain. Such indirect approaches do not impose a heavy data collection burden on routers. In this paper, we will concentrate on an indirect approach through link traffic counts.

Vardi [11] was the first to investigate this problem. Assuming iid Poisson distributions for the OD traffic byte counts on a general network topology, he specifies the identifiability of the Poisson model and develops an expectation-maximization (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. Related work treated the special case involving a single set of link counts: Vanderbei and Iannone (1994) apply the EM algorithm, and Tebaldi and West (1998) have a Bayesian perspective and a Markov Chain Monte Carlo implementation.

With real network data, Cao *et al.* [6] revise the Poisson assumption and propose a normal model. They also address the non-stationarity of network OD traffic. Their methodologies are validated through comparisons with directly collected (but expensive) OD traffic in a simple network at Lucent Technologies depicted in Fig. 3. Another recent work in OD traffic matrix estimation is done by Medina *et al.* [12]. They have a detailed comparative evaluation of the three existing techniques and propose a new direction for OD matrix estimation based on *choice models* in order to incorporate additional data and information about the network into OD traffic modelling. Our pseudo likelihood method is based on Cao *et al.*'s work, so next we will briefly review their normal model and methodologies in the network tomography framework.

The network tomography model specified by (1) is applicable to the OD matrix inference through link traffic counts. Here, $\mathbf{Y} = (Y_1, Y_2, ..., Y_I)'$ 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. $\mathbf{X}$ 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. For instance, considering the single-router network shown in Fig. 3(b), we can write its routing matrix A (0 for entries not indicated) as

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

Each column of A is associated with one OD pair and each row corresponds to an observed link traffic. For instance, the first row of the above A means that the traffic originated from node *fddi* consists of 4 pairs of unobserved OD pair traffic originated from *fddi* and destinated for to all other 4 nodes (including itself).

In the normal model [6], 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, where ϕ is a positive scalar applicable to all OD pairs and c is a power constant. It implies that

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

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

The mean-variance relationship is a key 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 [6], [13]. Because $c = 1$ or $c = 2$ give similar results, in this paper, we use $c = 1$ as done in [13]. But note

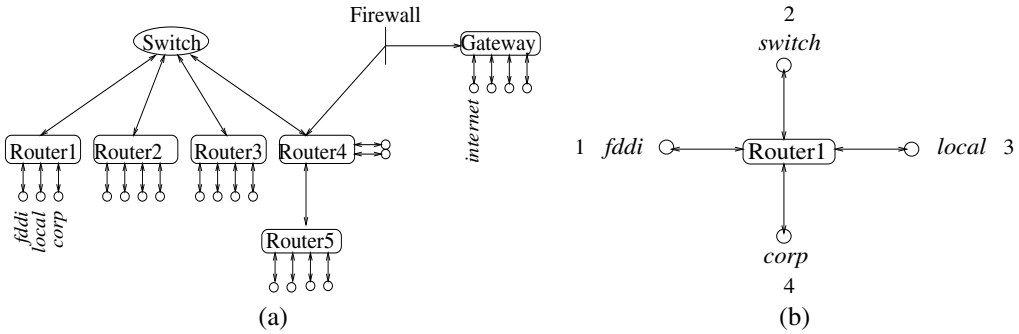

Fig. 3. (a) A Simple Router Network at Lucent Technologies, (b) Network Topology around Router 1.

that the pseudo likelihood method we will propose later can deal with $c = 2$ without any additional technical difficulties.

Cao *et al.* [6] 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 $E_{\hat{\theta}}(x_t|y_t, x_t > 0)$ is computed as an initial estimate of x. Then an iterative proportional fitting (IPF) algorithm [14], 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 [6], 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 [13], the computational complexity of each EM step is $O(n^5)$ after exploiting sparse matrix calculation. A pseudo likelihood approach will be applied to the above normal model, and comparisons will be made between full likelihood and pseudo likelihood methods with respect to the computational complexity and estimation efficiency in later sections.

III. Pseudo Likelihood in Network Tomography Inference

A. Forming Subproblems

Either in the problem of internal link delay distribution estimation through multicast end-to-end delay measurements, or in the problem of OD matrix inference through link byte counts, the maximum likelihood estimate (MLE) is computationally intensive. A pseudo likelihood approach would be desirable if its computation cost is much lower, while its estimation efficiency is still comparable to MLE.

Our pseudo likelihood approach is motivated by the decomposition of multicast spanning trees. After decomposition, it is equivalent to the delay inference problem in the unicast framework [8]. Consider a subtree decomposition scheme depicted in Fig. 4. A virtual two-leaf subtree is formed by only considering two receivers R_1, R_3 in the original multicast tree. The marginal likelihood function of the virtual two-leaf subtree is tractable because of its simple structure. Each virtual link in a subtree may consist of several real links, so each subtree only gives the specific path-level but not link-level delay distributions. For a multicast tree with I end receivers, there are totally $I(I-1)/2$ subtrees: different subtrees contain delay distribution information on different paths. All these path-level delay information together enable us to recover the link-level delay distributions. An efficient way of estimating link-level delay parameters is to consider all subproblems simultaneously. If we ignore dependences between subtree problems, then the pseudo likelihood function is obtained by multiplying marginal likelihood functions of all subproblems.

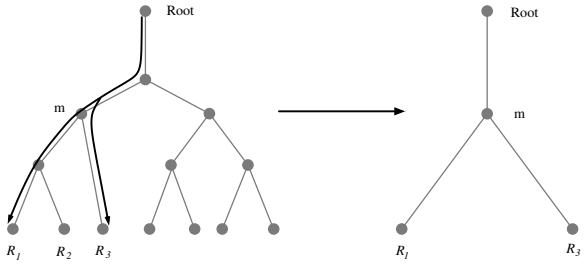

Fig. 4. Pseudo Likelihood: subtree decomposition

The above treatment for internal link delay distribution estimation is equivalent to picking up a pair of rows from its routing matrix A to form a subproblem. Such an idea can be extended to the general network tomography model specified in (1). Let S denote the set of subproblems by picking up all possible pairs of rows from the routing matrix A: $S = \{s = (i_1, i_2) : 1 \le i_1 < i_2 \le I\}$. Here, we define two notations for later use:

$$\text{given } s,\ J^s = \{j :\ \mathbf{X}_j \text{ is involved in } s\ \};$$
$$\text{given } j,\ S^j = \{s :\ \mathbf{X}_j \text{ is involved in } s\ \}.$$

For each $s \in S$,

$$\mathbf{Y}^s = A^s \mathbf{X}^s, \tag{3}$$

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

multicast tree depicted in Fig. 2, the subproblem formed by only considering end receivers 4 and 5 can be written as:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_4 \\ x_5 \end{pmatrix},$$

where $y_1, ..., y_4$ are the measured delays at end receivers $4, ..., 7$ and $x_1, ..., x_7$ are the delays over links 1,..., 7 as before.

Let θ^s be the model parameter of a subproblem s, and $l^s(\mathbf{Y}^s; \theta^s)$ be its log-likelihood function. For an observed vector $\mathbf{Y}$, define the pseudo log-likelihood function as

$$L^p(\mathbf{Y}; \theta) = \sum_{s \in S} l^s(\mathbf{Y}^s; \theta^s). \tag{4}$$

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 unobserved and observed data vectors for the subproblem s, then the overall pseudo log-likelihood function is defined as:

$$L_T^p(y_1, ..., y_T; \theta) = \sum_{t=1}^{T} L^p(y_t; \theta). \tag{5}$$

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

$$\frac{\partial}{\partial \theta} L_T^p(y_1, ..., y_T; \theta) = 0. \tag{6}$$

For constructing a pseudo likelihood function, 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 these two examples shows that picking up two rows each time gives satisfactory estimation results while keeping the computational cost within a reasonable range.

B. Asymptotic Properties of MPLE

Consistency and asymptotic normality are basic properties of MLE. Under some general conditions, the consistency and asymptotic normality of MPLE in (5) can also be established. In the rest of the paper, let θ_0 be the true parameter of the network tomography model defined in (1) and (2), and θ_0^s be the true parameter of a subproblem 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 with positive probability, such that 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 (6) 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 a 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)** In additional to the assumptions specified in Theorem 1, if 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 interexchanged;
- B2) As $T \to \infty$, the Hessian matrix $\nabla^2 L_T^p(y_1, ..., y_T; \theta)$ is invertible in an open neighborhood ω around θ_0 with probability 1;
- 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 $C(\theta) = H(\theta)^{-1} B(\theta) H(\theta)^{-1}$.

The uniformly convergence of $\nabla^2 L_T^p(y_1, ..., y_T; \theta)$ to $H(\theta)$ in an open interval ω can be often verified by checking the boundness of $\mathrm{E}\partial^3 L_T^p / \partial\theta_j \partial\theta_k \partial\theta_l$, which is true for most L_T^p of analytic functions. With satisfied continuous conditions, the invertibility of the Hessian matrix $\nabla^2 L_T^p(y_1, ..., y_T; \theta)$ can be verified by the convexity of $\mathrm{E}_{\theta_0} \nabla^2 L_T^p(\mathbf{Y}; \theta)$ at the true parameter θ_0. These observations are used in the proofs of Theorem 4 and Theorem 5.

Because almost identical proofs can be found in [15] or [4], proofs of Theorem 1, 2 are omitted. Please see Lehmann (1998) and White (1994) for more details.

C. Pseudo-EM Algorithm

Maximizing the pseudo function leads to our estimate, but usually the pseudo likelihood equation (6) cannot be solved analytically; hence, a numeric optimization algorithm has to be adopted. The EM algorithm [16] is a well known method for maximizing the likelihood function numerically, but it does not work for any objective function. The following 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 of a subproblem 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

$$Q(\theta, \theta^{(k)}) = \sum_{s \in S} \sum_{t=1}^{T} \mathrm{E}_{\theta^{s(k)}} \left(l^s(x_t^s; \theta^s) | y_t^s \right). \tag{7}$$

which is obtained by assuming the independence of subproblems in the expectation step. As holds for the EM algorithm, we have the following theorem for the pseudo-EM algorithm.

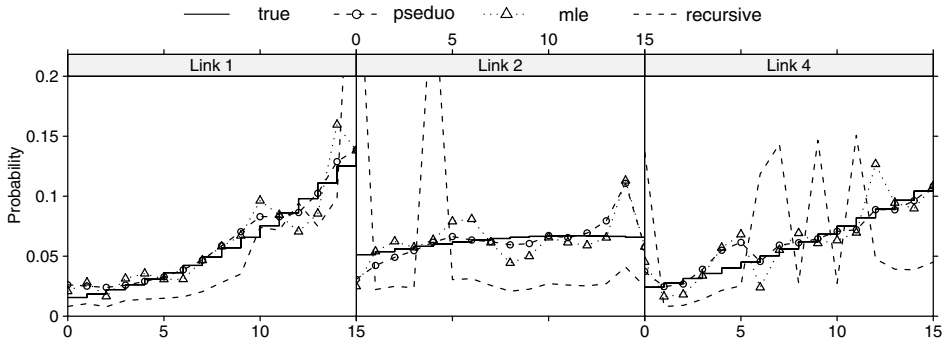

Fig. 5. Delay distribution estimates of 3 arbitrarily selected internal links: Link 1, Link 2 and Link 4. Solid step function is the true distribution, dash line with circle is MPLE, dotted line with triangle is MLE, and dash line only is recursive estimate.

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 the pseudo-EM algorithm will converge to the unique maximum point.

IV. Applications of Pseudo Likelihood

A. Multicast Internal Delay Distribution Inference

1) Parameter Estimation through Pseudo-EM: For any subproblem s in the problem of the internal delay distribution estimation through multicast end-to-end measurements, each component of $\mathbf{X}^s$ is an independent multinomial random variable, so the log-likelihood function given the complete data $x_1^s, ..., x_T^s$ can be written as

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

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

Let $\theta^{(k)}$ be the parameter estimate obtained in the kth step of the pseudo-EM. According to (7), we have

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

E-step: Compute

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

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

$$\theta_{jl}^{(k+1)} = \frac{\hat{n}_{jl}}{\sum \hat{n}_{j\cdot}}.$$

The initial value of the pseudo-EM algorithm can be chosen arbitrarily. We find that small starting values may have difficulties increasing even when the true parameter is large. 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 complexity of each step of the pseudo-EM algorithm is $O(m^3 I^2 P^2)$. Meanwhile, we have the following theorem about the consistency and asymptotic normality of the MPLE.

Theorem 4: Let $\tilde{\theta}_T^p$ be the MPLE of the problem of internal delay distribution inference through multicast end-to-end measurements, then $\tilde{\theta}_T^p$ is a consistent estimator and

$$\sqrt{T}(\tilde{\theta}_T^p - \theta_0)$$

converges to a multivariate normal random variable in distribution when $T \to \infty$.

2) Experiment results: In order to assess the performance of the pseudo likelihood methodology, model simulations are carried out on a 4-leaf multicast tree depicted in Fig. 2. The MLE method is implemented for this multicast tree due to its small size, so we can compare the performance of MPLE with those of MLE and recursive algorithm. The initial value of the EM algorithm in computing MLE is also set to be the uniform distribution. For each link, the number of bins m is set to be 14. During each simulation, 2000 iid multicast measurements are generated with each internal link having an independent discrete delay distribution. For three arbitrarily selected links, the results of delay distribution estimate from one experiment are shown in Fig. 5 along with their true delay distributions. The plot shows both MPLE and MLE capture most of the link delay distributions and their performace is comparable, while the recursive algorithm sometimes gives estimate far from the truth. The recursive algorithm is derived from relationships between a multicast tree and its subtrees, which yield polynomial constraints. The poor performance of recursive algorithm in this case is partly due to instability of the roots of such polynomial functions. Also from the plot, both MPLE and MLE still have some estimates quite different from the true probability. Mainly, this is because 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.

The same procedure is repeated independently 30 times for $T = 2000$. Fig. 6 shows the L_1 error norm of MPLE, MLE and recursive estimate for each link, as averaged over those 30 independent simulations. For each link, the L_1 error norm is simply the sum of the absolute difference 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 [17]. The plot shows that MLE and MPLE have comparable

estimation performance for tracking link delay distributions, while the recursive algorithm has much larger L_1 errors on all links. Meanwhile, we can see that MPLE has smaller SD of L_1 error norm than MLE on all links, implying that MPLE is more robust than MLE. This is possibly because the pseudo likelihood function, which is a product of less complex likelihood functions on subproblems, has a nicer surface than the full likelihood function.

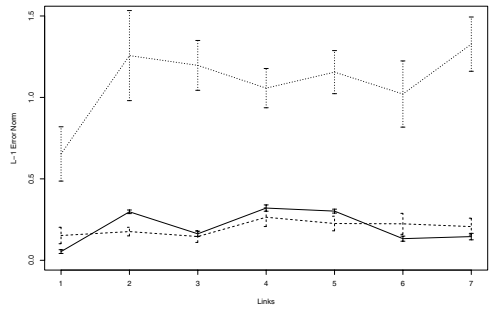

Fig. 6. Link L_1 error norm averaged over 30 simulations: solid line is MPLE, dashed line is MLE, and dotted line is recursive algorithm. For each link, the vertical bar shows the SD of L_1 error norm for the given link.

B. 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

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

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

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

where

$$m_t^{s(k)} = E_{\theta^{s(k)}}(x_t^s|y_t^s),\ \mathcal{R}^{s(k)} = \mathrm{Var}_{\theta^{s(k)}}(x_t^s|y_t^s).$$

Furthermore, we have

$$\begin{aligned} m_t^{s(k)} &= \lambda^{s(k)} + \Sigma_s^{(k)}A^{s\prime}(A^s\Sigma_s^{(k)}A^{s\prime})^{-1}(y_t^s - A^s\lambda^{s(k)}), \\ \mathcal{R}^{s(k)} &= \Sigma_s^{(k)} - \Sigma_s^{(k)}A^{s\prime}(A^s\Sigma_s^{(k)}A^{s\prime})^{-1}A^s\Sigma_s^{(k)}. \end{aligned}$$

Because Σ_s is a 2×2 matrix, (8) can be simplified as

$$\begin{aligned} Q(\theta, \theta^{(k)}) \propto -\sum_{s\in S}\sum_{j\in J^s}\Bigg\{ & \sum_{t=1}^{T}\frac{(m_{tj}^{s(k)} - \lambda_j)^2}{\phi\lambda_j} \\ & + T\left[\log\phi + \log\lambda_j + \frac{r_j^{s(k)}}{\phi\lambda_j}\right]\Bigg\} \end{aligned}$$

where $r_j^{s(k)}$ 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 $\frac{\partial}{\partial\theta}Q(\theta, \theta^{(k)}) = 0$ is equivalent to

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

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

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

M-step: Solve (9). Equation (9a) shows a quadratic constraints between λ_j and ϕ. Hence, we can find positive solutions to these equations explicitly. In conjunction with these solutions, (9b) gives a functional constraint on ϕ. This function is strictly increasing, so fast algorithms are available to solve these equations. Because the cost of the E-step is high related to M-step, a Multiple-Step Gradient EM algorithm (a natural extension to Lange's Gradient EM algorithm [18]) 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 experiments below, we adopt the same initial value used in [6], 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 a starting point gives very stable performance for the Lucent data set.

For a network with n nodes, the number of observed unidirectional linka 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. The computation cost of

each subproblem is proportional to its number of OD pairs, so for the n^2 subproblems, the overall computational complexity of each pseudo-EM step is $O(n^{3.5})$. Compared with the one-step 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.

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) Experiment results: We apply the pseudo likelihood approach to a small but real network depicted in Fig. 3(b), then our results are compared with those of full likelihood. 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 a good comparison, we use the the data collected on Feb 22, 1999, the same day used in [6], which consists of 288 data points.

In order to capture the time varying nature of the network traffic, we adopt the same window size 11 from Cao *et al.* for the local moving iid model. Fig. 7 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. The average absolute error for MLE is about 5.5k, while 9.6k for MPLE. It shows that both likelihood methods capture the dynamics of the OD traffic counts quite well, and the full likelihood method has slightly better performance than the pseudo likelihood method for this dataset.

For estimating the actual time-varying OD traffic counts 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 some selected OD traffic counts estimate with its vertical axis magnified by a factor of 20 is shown in Fig. 8. These OD pairs are selected because of large errors in their estimates. The plot demonstrates that both pseudo and full likelihood methods have quite comparable performance even in error-prone small scales. Even though sometimes the estimation errors are large, both estimates performs well if compared to the range of all feasible OD traffic estimates which are non-negative and may account for the observed link counts, i.e., the largest possible error we can make. For instance, we compute the estimate errors of all 16 OD pairs for both MLE and MPLE at 3:30 AM, then divide them by their largest possible errors. Ratios for these two methods are close: the maximum error ratio for MLE is 8%, while 9% for MPLE. In this measure, we can see both methods contribute a substantial amount in capturing the network OD traffic.

The above computations are completed using R 1.5.0 [19] on a 1G Hz laptop. In producing Fig. 7, 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.

V. Conclusion

In this paper, we proposed a pseudo likelihood approach to the network tomography problem and used 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.

Acknowledgment

We would like to thank two anonymous referees for their constructive comments on the first submission of the 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 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 4

Proof: First, in order to show the distinctness of $L_T^p(\theta)$, we only need to prove that for any $\theta_1 \neq \theta_2$, there exists some s, such that $l^s(\mathbf{Y}^s; \theta_1^s) \neq l^s(\mathbf{Y}^s; \theta_2^s)$. If we have $l^s(\mathbf{Y}^s; \theta_1^s) = l^s(\mathbf{Y}^s; \theta_2^s)$ for all s, then for each s, it is easy to show its three virtual sub-paths, e.g., in Fig. 2(b), $0 \to m$, $m \to R_1$ and $m \to R_3$, have the same distribution when the true parameter is either θ_1 or θ_2. Iterate the argument over all sub-problems, we have all such sub-paths have the same distribution under θ_1 or θ_2. By deconvolution, it implies each internal link must have the same distribution under θ_1 or θ_2, i.e., $\theta_1 = \theta_2$. Hence, we prove the distinctness of $L_T^p(\theta)$.

Second, we want to show $L_T^p(\theta)$ is strictly convex at a neighborhood of the true parameter θ_0 with probability 1

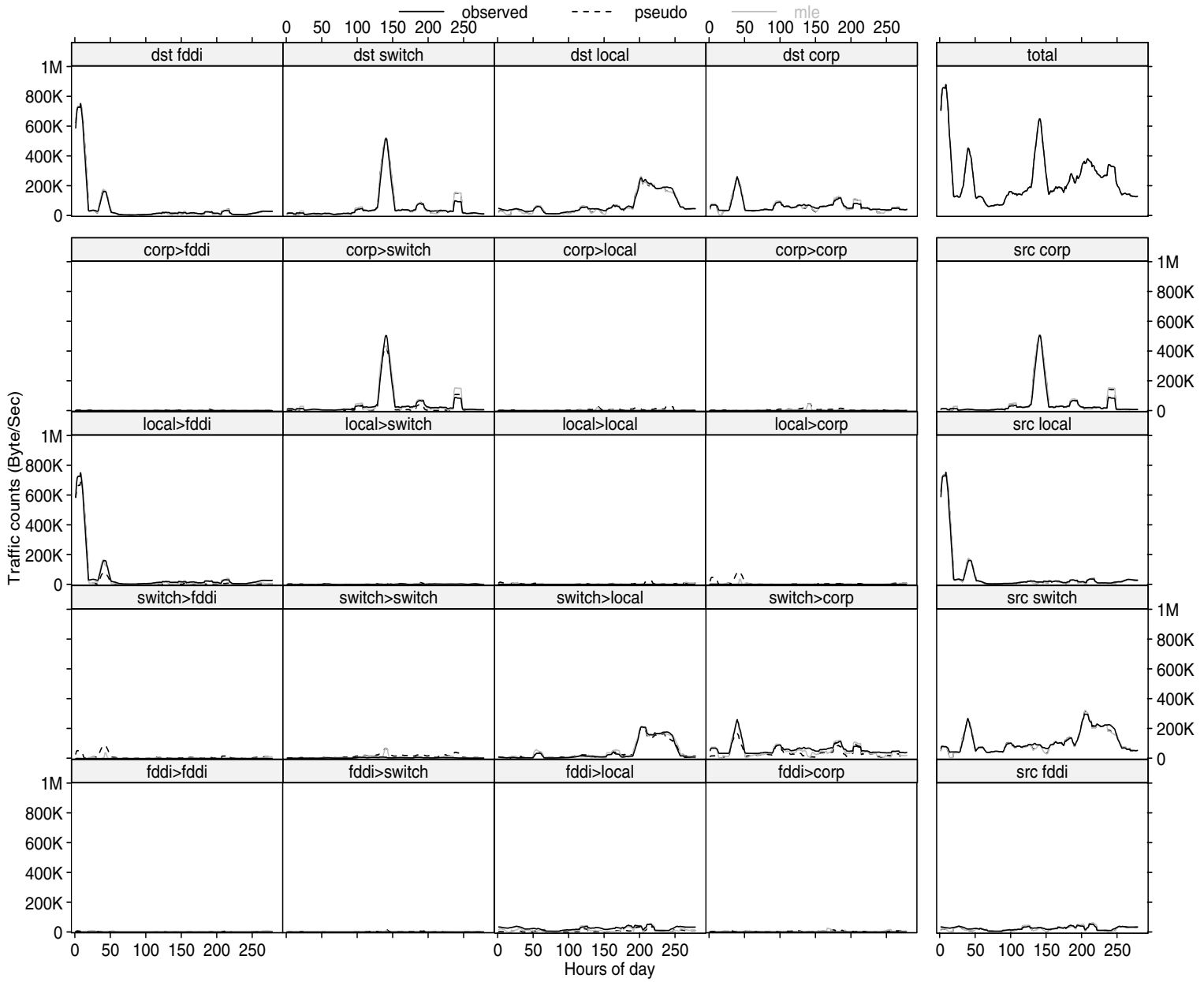

Fig. 7. Mean OD traffic λ estimate for 4 nodes network around *Router1* from pseudo and full likelihood against the moving average of true OD traffic. Marginal panels show the marginal traffic.

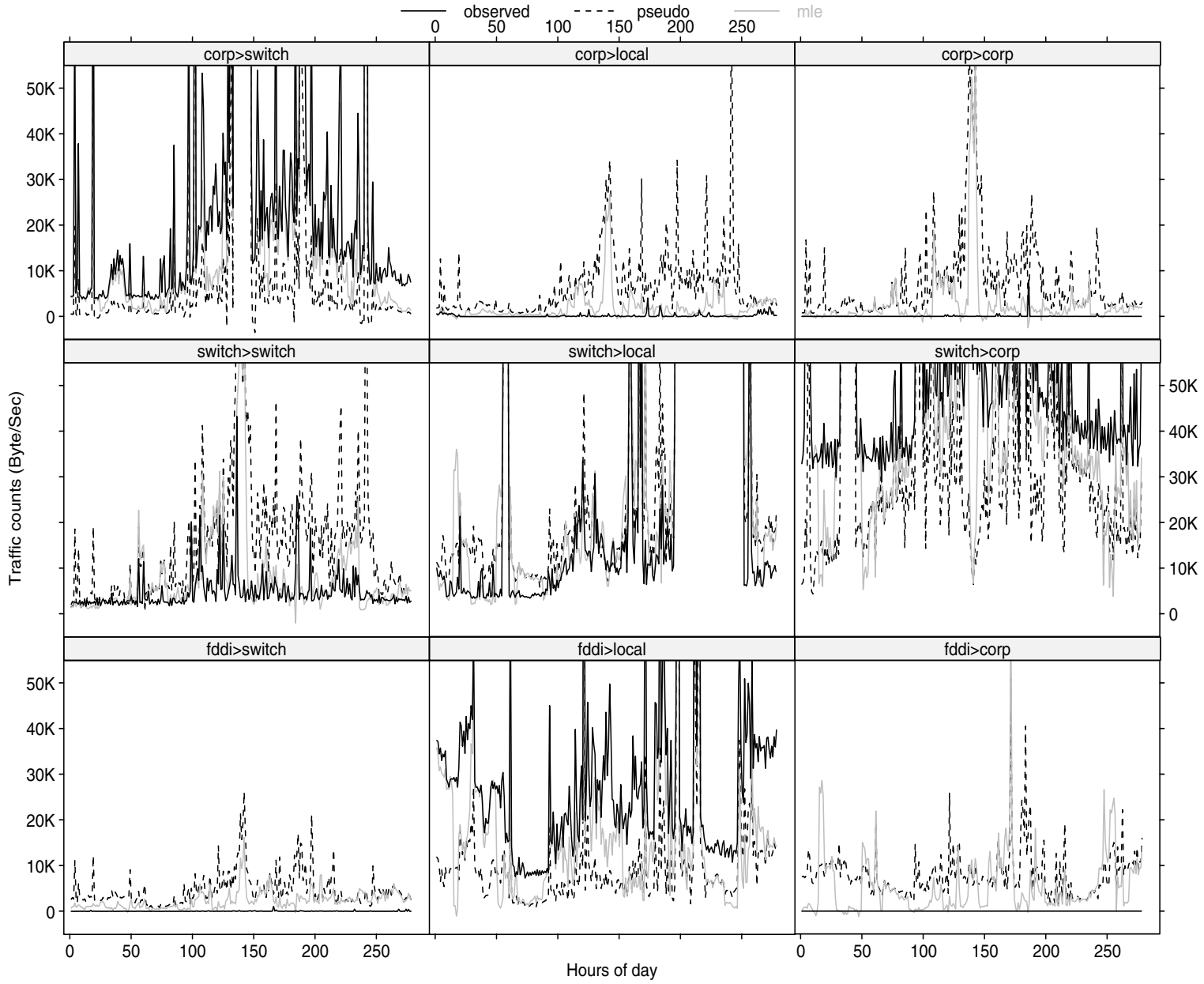

Fig. 8. OD traffic counts estimate for 4 nodes network around *Router1* from pseudo and full likelihood against the true OD traffic counts: only 9 selected pairs are shown. The scale is zoomed in by a factor of 20 to show detailed features.

when T goes to infinity. Because $y_1, ..., y_T$ are iid copies of $\mathbf{Y}$, we only need to prove that $\mathrm{E}L^p(\mathbf{Y};\theta)$ is strictly convex at a neighborhood of θ_0, i.e., the Hessian matrix $-\mathrm{E}\nabla^2 L^p(\mathbf{Y};\theta_0) = -\mathrm{E}\nabla^2 L^p(\mathbf{Y};\theta)|_{\theta=\theta_0}$ 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, so we have $\mathrm{E}\nabla l^s(\mathbf{Y}^s;\theta_0^s) = 0$ and

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

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 $\mathrm{E}\nabla L^p(\mathbf{Y};\theta_0) = 0$ and the Hessian matrix $-\mathrm{E}\nabla^2 L^p(\mathbf{Y};\theta_0)$ is non-negative definite. Now we want to show it is actually positive definite. Suppose the true parameter θ_0 is in the interior of the parameter space. If there is a vector α, such that

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

It implies $\alpha'\nabla l^s(\mathbf{Y}^s,\theta_0^s) = 0$ for any $\mathbf{Y}$ and s. Now we will iterate all subproblems in a deterministic way to show $\alpha = 0$. For instance, consider the multicast tree depicted in Fig. 2(a). $\alpha'\nabla l^s(\mathbf{Y}^s,\theta_0^s) = 0$ in the subproblem $s = (1,4)$ (subproblem formed by considering only end nodes 4 and 7) shows the components of α corresponding to Link 1 are all 0. Combining with such information, subproblem $s = (1,2)$ will show the components of α corresponding to Link 2 are all 0. Iterate all subproblems in such a top-down (or bottom-up) fashion, we have $\alpha = 0$, i.e., the Hessian matrix is positive definite.

By Theorem 1, the MPLE is consistent. Also we can check that 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: Similarly for OD matrix inference problem, $L_T^p(\theta)$ is distinct if and only if $l^s(\theta_1^s) = l^s(\theta_2^s)$ 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_1th, i_2th rows of A be the first two rows and their component-wise product be the third row, then $l^s(\theta_1^s) = l^s(\theta_2^s)$ implies $\phi_1 = \phi_2$ and $B^s\lambda_1 = B^s\lambda_2$. So 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 subproblem 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 parameter space, the asymptotic normality follows by the fact that the third derivative of of $L_T^p(\theta)$ exists and its expectation is bounded in a neighborhood of θ_0.

■

REFERENCES

[1] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistical Society, Series B*, vol. 36, no. 2, pp. 192–236, 1974.

[2] ——, "Statistical analysis of non-lattice data," *Statistican*, vol. 24, no. 3, pp. 179–195, 1975.

[3] D. R. Cox, "Partial likelihood," *Biometrika*, vol. 62, pp. 269–276, 1975.

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

[5] M. Coates, A. Hero, R. Nowak, and B. Yu, "Internet tomography," *Signal Processing Magazine*, vol. 19, no. 3, pp. 47–65, 2002.

[6] J. Cao, D. Davis, S. V. Wiel, and B. Yu, "Time-varying network tomography: router link data," *Journal of American Statistics Association*, vol. 95, no. 452, pp. 1063–1075, 2000.

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

[8] M. Coates and R. Nowak, "Network tomography for internal delay estimation," 2001. [Online]. Available: citeseer.nj.nec.com/coates01network.html

[9] F. L. Presti, N. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal delay distributions," AT&T Laboratories and University of Massachustts, Tech. Rep., 1999.

[10] 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*, pp. 265–279, June 2001.

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

[12] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic Matrix Estimation: Existing Techniques and New Directions," in *ACM SIGCOMM*, Pittsburgh, USA, Aug. 2002.

[13] J. Cao, S. V. Wiel, B. Yu, and Z. Zhu, "A scalable method for estimating network traffic matrices," Bell Labs, Tech. Rep., 2000.

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

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

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

[17] D. Scott, *Multivariate Density Estimation: Theory, Practice and Visulization*. Wiley, Ney York, 1992.

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

[19] R. Ihaka and R. Gentleman, "R: A language for data analysis and graphics," *Journal of Computational and Graphical Statistics*, vol. 5, no. 3, pp. 299–314, 1996.

[20] Y. Tsang, M. Coates, and R. Nowak, "Nonparametric internet tomography," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Orlando, Florida, May 2002.