Maximum Pseudo Likelihood Estimation

in

Network Tomography

Gang Liang and Bin Yu

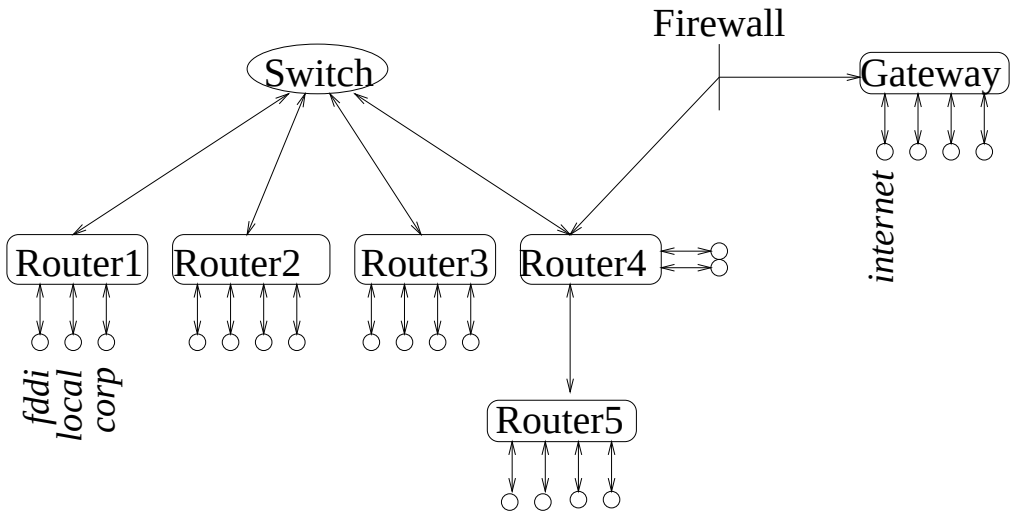
Statistics Department, Univ. of California at Berkeley

Outline

- Introduction to the Linear Network Tomography Model
- Two examples
  - Multicast delay estimation
  - Origin-Destination (OD) matrix estimation
- Maximum Pseudo-Likelihood Estimation (MPLE)
- Experimental Results
- Concluding Remarks

A Lucent Network

Network is like a postal system.



Basics of Network

- Network components
  - *nodes*
    - ∗ *edge nodes*: at boundary of network

      origination and destination of traffic (users of postal service)
    - ∗ *intermediate nodes*: routers that traffic passes through (postal hubs)
  - *links*
    - ∗ directional data pipes between nodes
- Routing
  - path

$$O \rightarrow R_1 \rightarrow R_2 \ldots \rightarrow R_{d-1} \rightarrow R_d \rightarrow D$$

  - routing by destination address.
  - fixed routing in Local Area Network (LAN).

Tomography

Unix Webster on tomography:

Date: 1935

: a method of producing a three-dimensional image of the internal structures of a solid object (as the human body or the earth) by the observation and recording of the differences in the effects on the passage of waves of energy impinging on those structures

Medical tomography:

- Computer Assisted Tomography (CAT scanning)
- Positron Emission Tomography (PET scanning)
- Single Photon Emission Tomography (SPECT scanning)

All inverse problems...

Network Tomography

The network tomography model was first used by Vardi (1996) to capture the similarities between origin destination (OD) matrix estimatino through link counts and medical tomography: in network inference, it is common that one does not observe quantities of interest but their aggregations instead and this goes beyond OD estimation.

General linear network tomography model

At a given time t,

$\mathbf{x}_t$: unknown quantity of interest (of dim J) (e.g, link delay, traffic flow counts).

$\mathbf{y}_t$: known aggregations of $\mathbf{x}_t$ (of dim I).

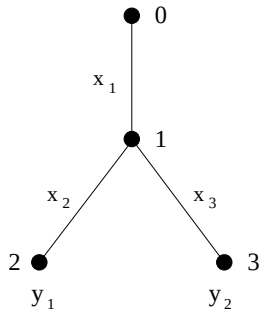
Problem: predict or estimate $\mathbf{x}_t$ from $\mathbf{y}_t$ with

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

where $\mathbf{A}$ is a 0-1 routing matrix.

Usually the number J of unknowns is much larger than number I of knowns so badly ill-posed linear inverse problem.

Example 1: Multicast Link Delay Estimation



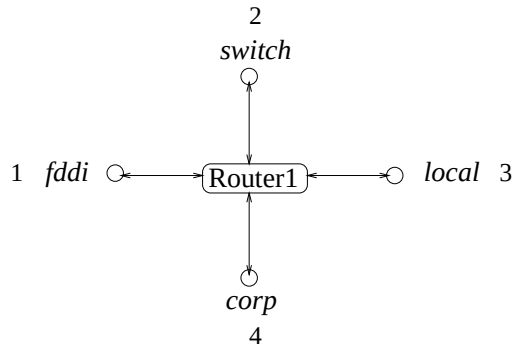
$$y_{1,t} = x_{1,t} + x_{2,t},\ y_{2,t} = x_{1,t} + x_{3,t};\ I = 2,\ J = 3$$

Test probes are sent from the root of a multicasting tree (where routers duplicate the probes and send them to its downstreem routers) and delays ($\mathbf{y}_t$) are observed at the receiver/end nodes. The problem is to infer the delays experienced by the probes along the path links ($\mathbf{x}_t$). Obviously,

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

where 1's in the ith row of $\mathbf{A}$ specify the links that the ith component of $\mathbf{y}_t$ travels through.

OD Traffic Matrix Estimation



- n = 4 edge nodes, 1 router
- $J = n^2$ = 16 OD pairs in $\mathbf{x}_t$
- I = 7 indep. links in $\mathbf{y}_t$

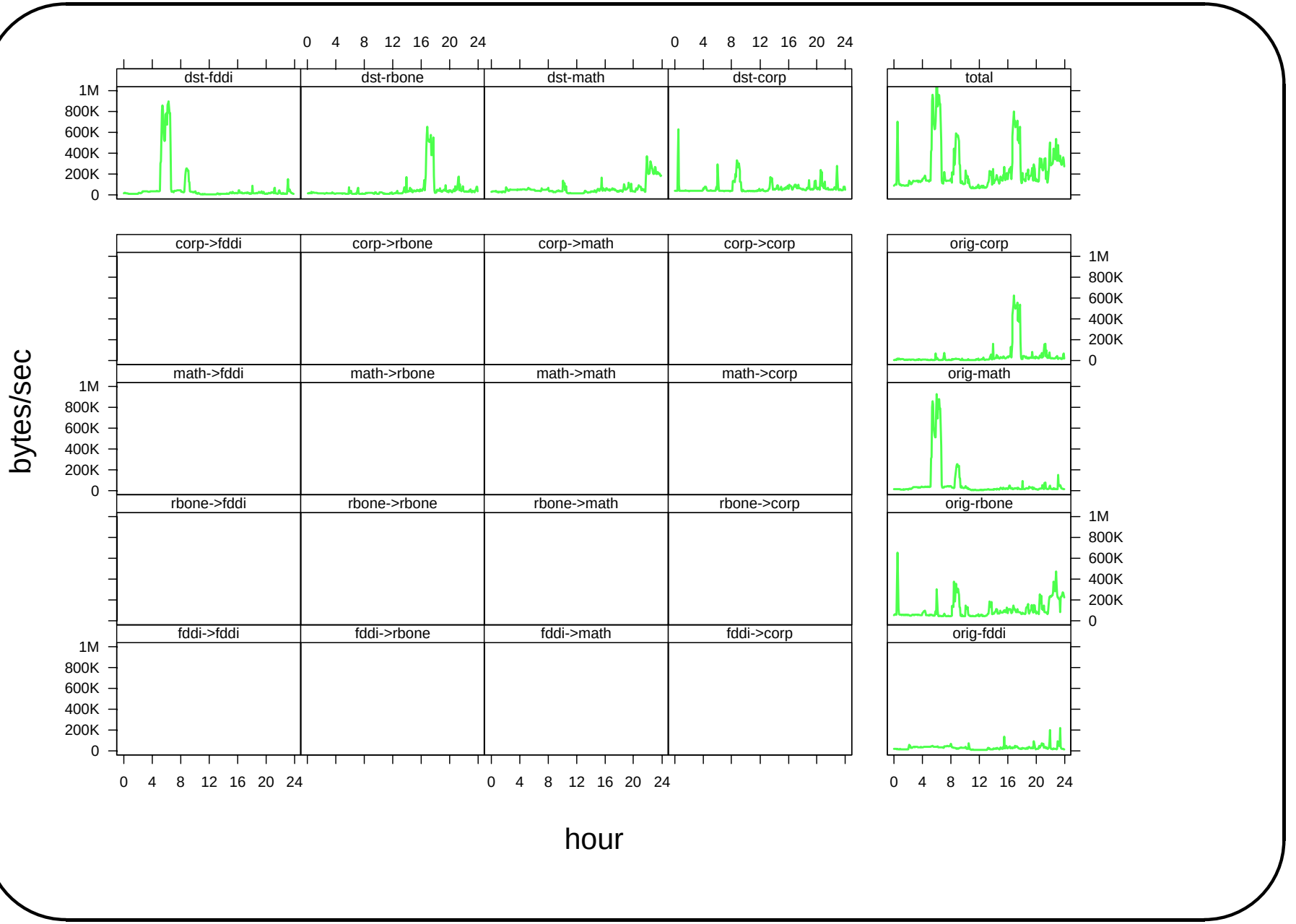
dst-fddi *1*	*dst-switch* *2*	*dst-local* *3*	*dst-corp* *4*	*total*	
4 → 1	*4 → 2*	*4 → 3*	*4 → 4*	*4*	*orig-corp*
3 → 1	*3 → 2*	*3 → 3*	*3 → 4*	*3*	*orig-local*
2 → 1	*2 → 2*	*2 → 3*	*2 → 4*	*2*	*orig- switch*
1 → 1	*1 → 2*	*1 → 3*	*1 → 4*	*1*	*orig-fddi*

Router 1 Measurements

- *Data*
  - Link measurements $\mathbf{y}_t$
  - Avg. bytes/s on consecutive 5 min intervals
  - Automatic collection by SNMP
  - Data collection started Aug. 1998

- *Plot shows*
  - time varying traffic
  - source patterns similar to destination patterns

Router 1 Link Data, Feb. 22



Maximum likelihood estimation (MLE) for the network tomography model is often computationally intractable for large networks.

1. Multicast delay estimation:

- Lo Presti et al (1999) give a fast and consistent recursive method based on a discrete-valued delay model ($X_j \in \{0, q, 2q, ..., mq, \infty\}$);
- We tried MLE, but could only do a very small tree.

2. OD estimation:

- Vardi (1996): MLE and method of moments for a Possion model.

  Follow-ups: Vanderbei and Iannone (1994) and Tebaldi and West (1998): EM and Bayesian estimation via MCMC, but with only one single-shot measurement.

- Cao et al (2000) use
  - SNMP link data of a Lucent network;
  - a Gaussian model for OD traffic: $X_j \sim N(\lambda_j, \phi^c \lambda_j)$ where $c = 1, 2$;
  - MLE via EM;
  - component-wise conditional expectation to impose non-negativity and iterative proportional refitting to impose $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t$;
  - an iid moving window (local likelihood) to address nonstationarity.

  But MLE is too slow for large networks.

Maximum Pseduo-Likelihood Estimation (Liang and Y, 2003, IEEE-SP)

In order to overcome the computational difficulty of MLE, Besag (1974) proposed a pseudo likelihood (PL) approach for Markov random field (MRF) inference problems:

- Subproblems are formed by neighborhood decomposition;
- Pseudo likelihood function is obtained by multiplying the conditional likelihoods from the subproblems, ignoring the dependences between the subproblems.

Our pseduo likelihood

- has a different scheme for forming subproblems and
- uses likelihood instead of conditional likelihood.

But they share the same divide-and-conquer principle.

Forming the subproblems

Recall that each row of $\mathbf{A}$ corresponds to a link traffic count r.v. Define the set of subproblems to be

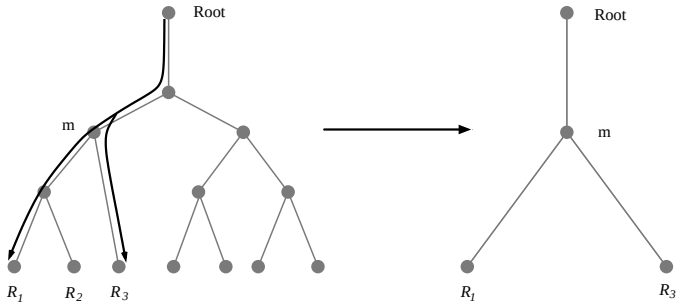
$$S = \{(i_1, i_2) : 1 \leq i_1 < i_2 \leq I\}$$

i.e., forming subproblems by only focusing on two rows of $\mathbf{A}$ at a time.

- For selected pairs rows of $\mathbf{A}$, write down the likelihood and multiply them to get the pseudo likelihood. The maximum pseudo likelihood estimator (MPLE) maximizes this pseudo likelihood.
- Consistency and asymptotic normality can be shown for this MPLE under regularity conditions, provided that $(\mathbf{x}_t)$ are iid (we believe they also hold under weak dependence).

This MPLE approach works for the general network tomography problem with $\mathbf{A}\mathbf{x}_t = \mathbf{y}_t$ (including multicast link delay estimation and OD estimation).

Example 1 (continued): Multicast link delay estimation

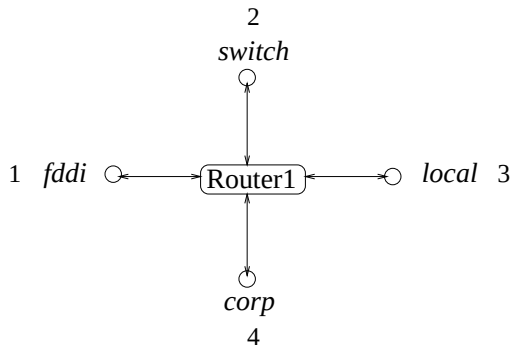


We employ Lo Presti et al (1999)'s discrete value model.

In this talk, we use all possible pairs, but consistency is guarranted if and only if the selected pairs split at all internal nodes. Given consistency, asymptotic normality also holds under regularity conditions. We are working on the design issues of pair selections...

At the pair-wise level, the likelihood is equivalent to that of the unicast delay estimation of Coates and Nowak (2001) and that of the bicast delay estimation of Lawrence, Michailidis and Nair (2002).

Example 2 (continued): OD estimation



Taking two rows of $\mathbf{A}$ corresponds to taking two links.

Consistency and normality for MPLE can be shown based on similar conditions as those for MLE.

One can decide algorithmically which pairs of links to take to ensure consistency, but it is hard to describe in general terms since it depends on the varied topology of the network.

Estimation of MPLE via Pseudo-EM

Let $\ell_s(X^s;\theta_s)$ be the log-likelihood function on a sub-problem s given the complete data X^s. Let θ^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

$$Q(\theta,\theta^k)=\sum_s\sum_{t=1}^{T}E_{\theta_s^k}\left(\ell_s(X_t^s;\theta_s)|Y_t^s\right). \tag{1}$$

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

Example 1 (continued): a simulated experiment

Complexity comparisons for different methods:

I – number of end receivers; grows with tree

m – number of possible discrete values; fixed, but easily > 10

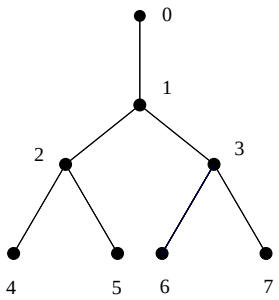
P – average number of links per path; $O(logI)$ for complete trees

Recursive: $O(mI)$

MLE: roughly in between $O((m+2)^(I+1))$ and $O((m+2)^{2I+1})$

MPLE: $O(m^3 I^2 P^2)$

Simulation set-up:



Independent discrete delays with $m = 14$ for all 7 internal links

n=2,000.

computation times:

Recursive: negligible

MLE: 25 seconds on average

MPLE: 16 seconds on average

When n=20,000, performance of recursive catches up.

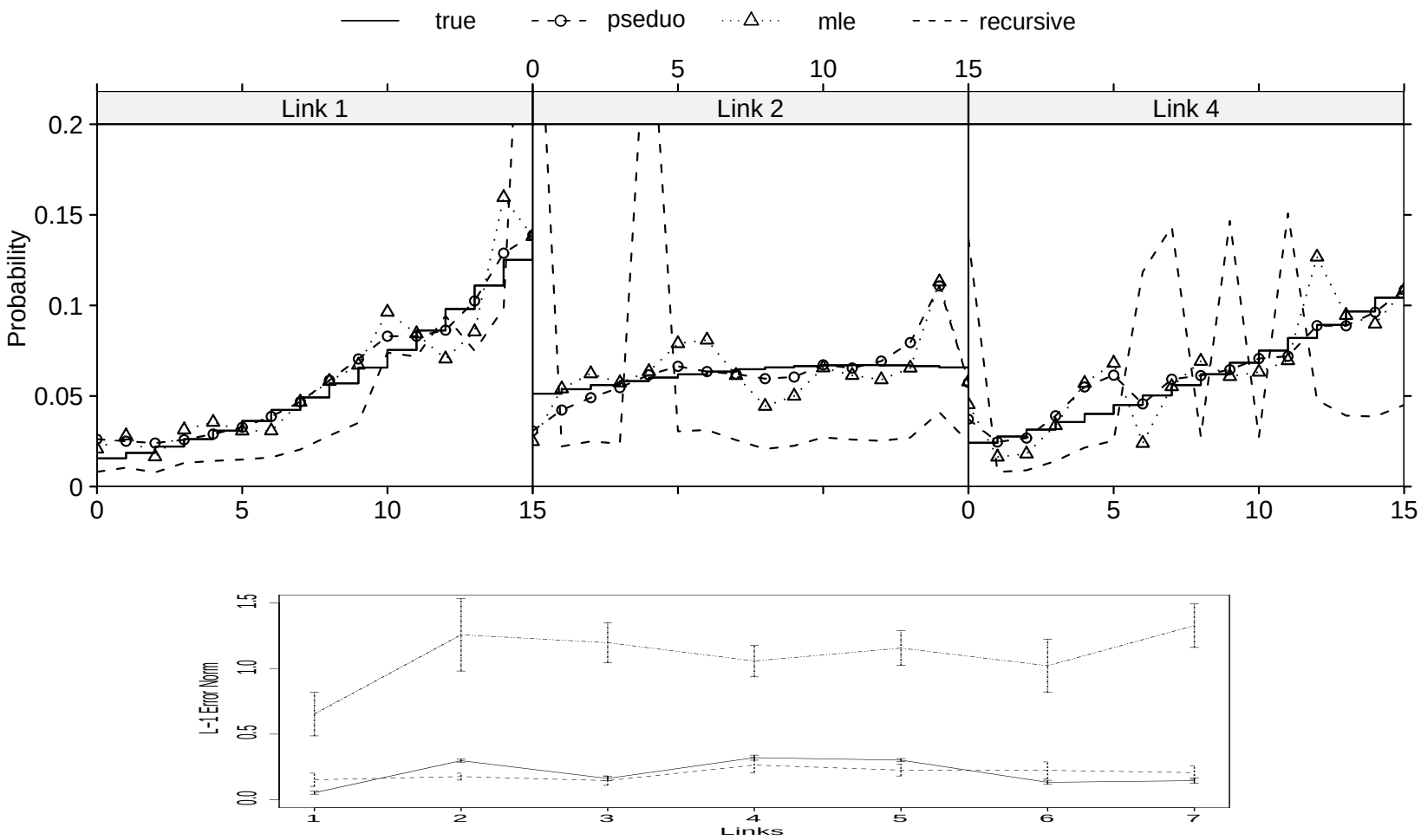


Figure 1: Dotted – Recursive; Solid – MLE; Dashed – MPLE. n=2,000

Example 2 (continued): OD estimation

MLE computation as in Cao et al (2000):

- *Stage 1.* EM iterations go to the neighborhood

- *Stage 2.* Newton method
  - use two derivatives of log-likelihood
  - use an optimization package in R (or S)

Recall: computation complexity of MLE: $O(n_e^5)$ using sparsity of $\mathbf{A}$ – this is not scalable to large networks.

Bottleneck: inversion of $n \times n$ matrices

MPLE computation:

Let K be the number of subproblems, i.e., the selected number of pairs of rows of $\mathbf{A}$, then

$$K \leq I(I-1)/2 = O(n_e^2).$$

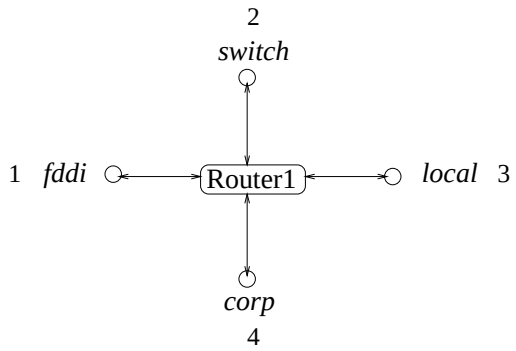
In our experiments, we use all pairs or $K = I(I-1)/2$.

Same initial values as Cao et al (2000).

E-step: Instead of the $n_e \times n_e$ matrix inversion for ML, we have K inversions of 2×2 matrices. And these inversions can be computed parallelly.

M-step: Multi-step gradient EM algorithm (fixed number of steps to solve the pseudo-likelihood equations approximately).

Comparisons with Router 1 data with validation

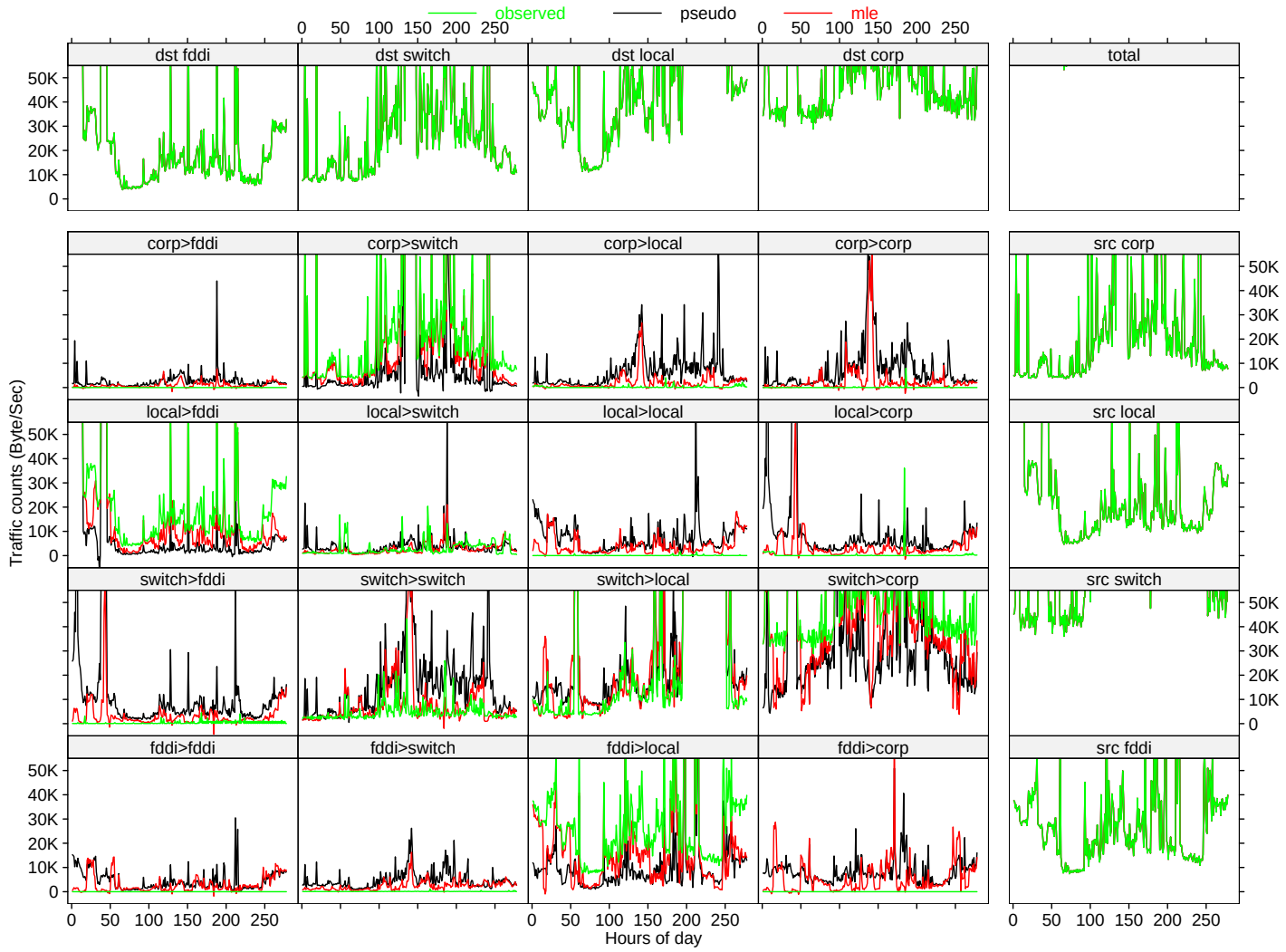


n = 4 edge nodes, $J = n^2$ = 16 OD pairs in $\mathbf{x}_t$, $I = 2n - 1$ = 7 indep. links in $\mathbf{y}_t$

- Measure OD traffic, $\mathbf{x}_t$, directly
- Requires special hardware and software
- CISCO's *netflow* pumps data to CAIDA's *cflowd*
- Automatic 5 minute aggregation

Estimated Mean Traffic

Estimated OD counts zoomed 25×



Computation time comparison for MLE and MPLE

Using network simulator ns, we simulated two networks of 8 end nodes and 21 end nodes, based on the Lucent network topology. For estimating the traffic counts, the computation times (in seconds) are as follows (using R and a 1GHz laptop):

# nodes	# links	MLE	MPLE	MPLE/MLE
4	7	48	12	0.25
8	16	82	18	0.21
21	49	2300	149	0.06

Recall that the computation complexity

- of MLE is $O(n^5)$ using sparsity of $\mathbf{A}$;
- of MPLE is $O(Kn^{1.5})$ if we assume the average link size for each OD path is $O(n^{0.5})$. When all pairs are used, it comes to $O(n^{3.5})$.

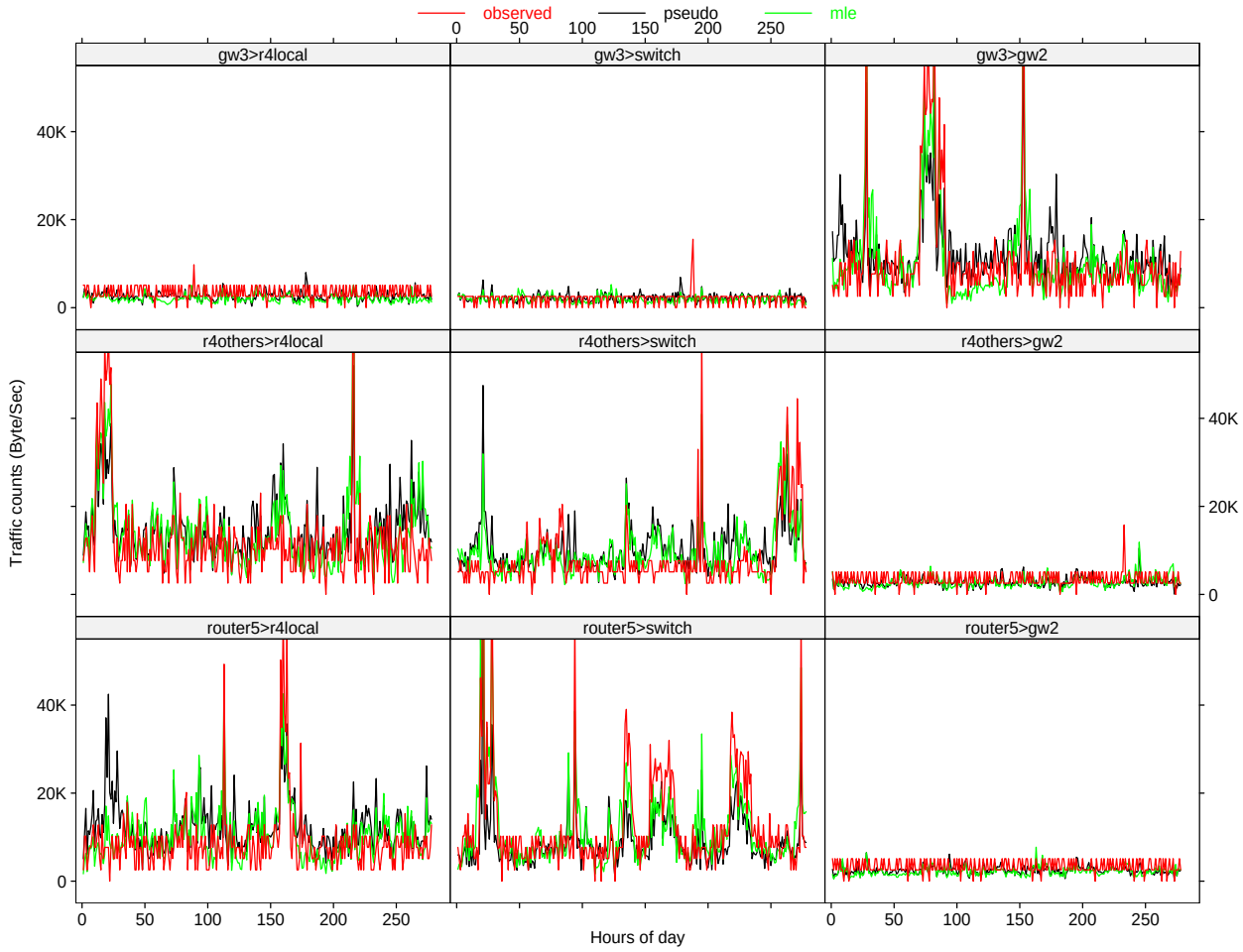


Figure 2: OD estimates of selected OD pairs for MLE and MPLE for the 8-node subnet work.

Concluding Remarks

- The maximum pseudo-likelihood approach gives a good trade-off between computation speed and statistical estimation efficiency for both multicast delay and OD estimations.
- OD traffic matrix estimation is crucial to dynamic updating of routing tables hence crucial to an economical use of the internet bandwidth resource.

  For OD estimation, we are working with people at Sprint Labs to test the pseudo likelihood approach with the Sprint network. With them, we are also starting to look at the OD problem with random routing (Vardi, 1996) (entries of A are fixed and inbetween 0 and 1). We believe MPLE would work there as well.
- The OD traffic matrix estimate is fed into different network optimization routines (e.g. to update the routing table or carry out dynamic routing). How does the uncertainty in the OD traffic estimate propagate through the down-stream optimization?