A Fast Lightweight Approach to Origin-Destination IP Traffic Estimation Using Partial Measurements

Gang Liang, Nina Taft and Bin Yu

***Abstract*— In this paper, we propose a novel approach to estimating traffic matrices that incorporates lightweight Origin-Destination (OD) flow measurements coupled with a computationally lightweight algorithm for producing the OD estimates. There are two key ingredients in our method, called PamTram, for PArtial Measurement of TRAffic Matrices. The first is to actively select a small number of informative OD flows to measure in each estimation time interval. To avoid the heavy computation of an optimal selection, we use a heuristic based on intuition from game theory. Randomized selection rules are developed based on the goals of reducing errors and adapting to traffic changes. We provide an algorithm for selecting a good flow to measure that is fast because it avoids the computations, such as integrating over past intervals, that are needed for optimal selection. The second key aspect of our method is an explanation and proof that an Iterative Proportional Fitting (IPF) algorithm can be used to approximate the traffic matrix estimate when the goal is a minimum mean squared error and the optimization starts from a maximum entropy initial estimate.**

In addition, we provide a one-step average error bound for PamTram when the randomized selection rule is uniform and no link counts are used. This bounds the average error for the worst case selection rule. Finally, we validate our method using data from Sprint's European Tier-1 IP backbone network. Results show that our method generates average errors below the 10% carrier target error rate. Interestingly, we show that it suffices to measure a single OD flow in each estimation interval, which renders our partial measurement method very lightweight in terms of measurement overhead.

***Index Terms*— iterative proportional fitting, minimax, origin-destination traffic matrix, partial measurement, statistical game**

I. INTRODUCTION

Origin-destination (OD) traffic matrices are network profiles that quantify the volume of traffic flow between all pairs of nodes in a given network. Such matrices serve as important inputs for a variety of network traffic engineering tasks, including capacity planning, load balancing, and traffic provisioning; hence, the problem of estimating OD traffic matrices for backbone networks has recently attracted much interest from both service providers [1], [2], [3] and the network research community [4], [5], [6], [7], [8].

A general traffic matrix can be defined at any level of granularity: the traffic sources and destinations could be hosts, groups of hosts, routers or even PoPs (a large collection of co-located routers). The specification of a particular traffic matrix requires the selection of the level of aggregation. In a router-to-router traffic matrix, the traffic considered to be "sourced" at a given router includes all of the clients and peers attached to that router. Most research has focused on either router-to-router or PoP-to-PoP matrices, and we continue in the same vein, as these are the ones ISPs are primarily interested in. For a network with n_e edge (or access) nodes, the number of possible OD traffic flow pairs is n_e^2. The OD matrix also has a timescale associated with it - each entry gives an average volume level over some time interval (1 min, 1 hour, 1 day, etc.). Traffic matrices should be thought of as 3-dimensional matrices in which the third dimension is time. Each OD traffic flow is actually a time series, and thus the entire matrix evolves over time. It has been shown ([2], [9]) that traffic matrices are quite dynamic and exhibit strong diurnal patterns thus varying a great deal within a 24 hour period.

Current approaches for obtaining traffic matrices can be classified into two categories: direct and indirect. A direct approach is a pure measurement one in which the entire traffic matrix is repeatedly measured over time via monitoring technologies such as Netflow on Cisco routers. In [2], the authors explicitly calculated the overheads of direct measurement using state-of-the-art flow monitors. They showed that today's solutions, which essentially mandate a centralized solution, are prohibitive in terms of communication and computation costs. They also illustrated that by moving towards a more distributed approach, the computation costs fall but the communications cost of full measurement (albeit smaller) still remains high.

The indirect approach relies on alternative data that is more readily available in networks, yet is incomplete. In particular, the Simple Network Management Protocol (SNMP), supplies statistics on links (e.g., total bytes seen in a 5 minute window) and is widely deployed in today's ISP networks. This is only partial information because typically the number of internal link constraints is much smaller than the number of OD pairs, thus creating an ill-posed inverse problem. Vardi [5] was the first to investigate the problem of estimating OD matrix through link traffic counts, and coined the term "network tomography" to illustrate its similarities with medical tomography. The challenge of the indirect approach lies in its ill-posed nature. For a general network, the number of links is usually proportional to the number of edge nodes n_e, which grows much more slowly than the number of OD pairs n_e^2. The problem becomes severely under-constrained even for a modest n_e. For instance, in a backbone network, n_e is in the range of 20-40 at the PoP level, and is on the order of hundreds at the backbone router level.

Many approaches to tackle these problems try to find a simple model for OD flows, introduce constraints to ensure

Gang Liang is with the Department of Statistics, University of California at Irvine, Irvine, CA 92697, USA. (email: liang@uci.edu)

Nina Taft is with Intel Rearch Berkeley, Berkeley, CA 92710, USA. (email: nina.taft@intel.com)

Bin Yu is with the Department of Statistics, University of California at Berkeley, Berkeley, CA 92720, USA. (email: binyu@stat.Berkeley.EDU)

the identifiability of the model, and then employ some form of maximum likelihood estimation. Vardi [5] proposed a Poisson model assuming iid (independent identically distributed) Poisson distributions for the OD traffic byte counts. Based on LAN network data, Cao et. al. [4] revise the Poisson assumption to propose a Gaussian model coupled with an assumption of a power-law relationship between the mean and variance of an OD flow. Vaton and Gravey [6] propose an empirical Bayesian method and an iterative algorithm is used to learn the prior distribution. In [10] the authors proposed the use of gravity models for determining initial conditions for optimization methods (such as maximum likelihood estimation) to avoid local minima problems. In Zhang et. al. [1], a tomogravity model is proposed to regularize the gravity parameter estimate such that the final estimate is also faithful to the SNMP link counts. The computation of these methods is usually very high. Liang and Yu [11] propose a pseudo likelihood method to speed up the parameter estimation for general network tomography problems.

A key question regarding the indirect approaches is to what level of accuracy can the hidden OD traffic be recovered simply from aggregated link traffic counts? Most of the indirect methods achieve average relative errors in the range 20-30%. However carriers are hoping for error rates to fall below the 10% barrier. In order to achieve lower error rates, recent research seeks to obtain yet more data (referred to as *side information* in statistics) to bring into the problem. Nucci et. al. [9] propose to use routing changes to obtained more information about the underlying OD traffic. Zhang et. al. [10] use SNMP data not only from inter-router links (as in the traditional problem), but also from access and peering links in order to populate the gravity model.

In this paper we propose the approach of using partial OD flow measurements as a good type of side information to bring into the problem. The idea is to measure a small number of OD flows (e.g., one) directly using a flow monitor, in each measurement interval, and then to vary the flow(s) measured over the course of time. This idea was originally proposed in [12]; however in that short paper neither the theoretical foundation for this approach nor any validation using data was carried out. We do both of those herein. Three partial flow measurement approaches were proposed and evaluated in the comparative study done in [8]. The notion of partial flow measurement in those approaches is different because they all propose to turn flow monitors on at *all* routers, for a period of 24 hours to measure the traffic matrix throughout its diurnal cycles. This data is used to calibrate a smart model. All flow monitors are then turned off until sufficient change has been detected so as to require them to be activated again, for another period of 24 hours, in order to re-calibrate the underlying models. While these approaches proved useful, the one we include in this study is far more lightweight. The measurement overhead in [8] varied from 5-30% depending upon the particular scheme; using their same overhead metric, our approach yields a measurement overhead of either 1% or 5% (depending upon the implementation).

Our contributions in this paper are multiple. First we introduce a simple non-stationary model to capture the 1-step temporal transitions of a traffic matrix. The intention of the model is to utilize the temporal relationship of network traffic within a local time window. Although it does not match the full (or global) OD flow behavior, we illustrate that it is sufficient for the purposes of accurate traffic matrix estimation and enables the use of less intensive computations.

Second, we propose a lightweight algorithm called PamTram, for **PA**rtial **M**easurement of **TRA**ffic **M**atrices. The proposed algorithm is a two-step procedure. In the first step, we propose a mechanism to select informative OD flows that will be measured in each interval, and in the second step we compute an approximation to the minimum mean square error (MSE) estimate to populate the traffic matrix. To select which flow(s) to measure we employ a game theoretic randomization scheme to choose informative OD pairs: several randomization schemes are proposed, and we also compute a bound on the 1-step error (the error of each successive estimate) under a simplified scenario. In our approach, different OD flows will be measured in different time intervals and the choice of which flows to measure is based on the probability that an OD flow will generate large errors. The benefit of this approach is that it permits adaptation to dynamic changes in the traffic matrix. When changes in particular OD flows occur, those flows are likely to generate larger errors; as our method progresses in time, it eventually catches these changes. We contend that the original ill-posed problem can be substantially improved even if only a tiny fraction of OD pairs are measured in each time interval.

Third, we prove that the iterative proportional fitting (IPF) algorithm can be used for our two critical computational steps: (i) it approximates the minimum Kullback-Leibler divergence estimate (as used in [1]), and (ii) can also be used to implement our game for selecting which OD flow to measure. Because IPF can be used inside these two steps of our methodology, our overall procedure yields an efficient and fast algorithm that is thus practical to implement. Finally, our methods are evaluated on real data from a Tier-1 operational backbone network. We compare our scheme to an oracle-based scheme to assess how far we are from optimality. We also compare our scheme to the Kalman filtering based method in [8] because it most closely resembles parts of our approach. We show that while the Kalman method performs reasonably well, PamTram consistently outperforms it across a variety of performance metrics, and achieves that with far lower measurement overhead.

This paper is organized as follows. In Section II, we review the OD traffic estimation problem, and an iterative proportional fitting (IPF) algorithm. In Section III, we propose a dynamic state-space network traffic model. In Section IV, we explain our approach to partial measurement and introduce a few minimax randomization selection schemes for selecting those traffic matrix elements to measure. We explain our data, evaluation setup and the schemes we compare ours to, in Section V. The evaluation is carried out in Section VI. We conclude our paper in Section VII and provide proofs of the theorems in the Appendix.

II. Background

A. Problem Statement

We denote the SNMP link counts as $Y = (Y_1, ..., Y_J)$ for a network with J links. Let $X = (X_1, \cdots, X_I)$ be the vectorized version of the traffic matrix where X_i denotes the i-th OD flow (for a total of I OD pairs). The OD traffic matrix X has been aligned into a vector for the convenience of mathematical manipulation. As in [5] and [4] there is a linear relationship between the unobserved X and observed Y:

$$Y = AX, \tag{1}$$

where A is an $J \times I$ routing matrix, determined by the network topology and the routing protocol. Mostly, elements of A take on the value of 0 or 1: $A_{j,i} = 1$ if OD pair i traverses link j, and $A_{j,i} = 0$ otherwise. The elements of A could take on fractional numbers when traffic splitting is allowed. Our proposed PamTram approach can deal with both cases. In this paper, we assume that the routing matrix A is known. One advantage of our approach is that the routing matrix is actually allowed to vary during the monitoring period as long as such information is obtained by the monitoring algorithm.

The goal of the traffic matrix estimation problem is to recover X from the observable Y and known A. In ISP and enterprise networks, we typically have $J \ll I$, and so A is not full rank. Thus the estimation of the distribution of X is an ill-posed inverse problem in the sense that the system equations $Y = AX$ have an infinite number of solutions, and hence constraints have to be introduced to ensure the identifiability of the model. There is a rich literature in statistics ([13], [14]) devoted to this topic from the point of view of regularization. In a broad sense, statistical modeling can be also viewed as introducing constraints by taking characteristics of network traffic dynamics into account.

B. I-projection and the IPF algorithm

Now we will review an important problem in information theory along with an iterative proportional fitting (IPF) algorithm for solving it. We do this because we will make use of the IPF solution as the work horse of our PamTram approach for solving the traffic matrix estimation problem. Further below we explain the connection between the two problems.

The minimum Kullback-Leibler (KL) divergence problem is one of the most fundamental questions in information theory. It can be stated as: given a probability function q, we would like to find a p inside a convex probability set $\mathcal{L}$ such that it minimizes the KL divergence between p and q, i.e.,

$$\hat{p} = \arg\min_{p \in \mathcal{L}} D(p||q). \tag{2}$$

Here q can be thought as the initial guess of the distribution to be estimated, and the convex set $\mathcal{L}$ represents the constraints we impose on the final feasible solutions. Such a formulation has been widely used in communication [15], econometrics [16], and many other areas.

I-projection, first studied by Csisźar [17], gives a geometric view to the above minimum KL divergence inference problems. The problem is viewed as projecting the initial guess q into the feasible convex set $\mathcal{L}$ where the KL divergence plays the role of squared Euclidean distance. Algorithmically, this geometric view suggests an alternating minimizing type of algorithm [17], which is useful for solving (2) if the constraint set $\mathcal{L}$ can be decomposed as the intersection of a series of convex constraint sets $\{\mathcal{L}_l : l = 1, \cdots, L\}$. In practice, many real problems have only linear constraints as special cases of the convex set. Several iterative algorithms ([18], [19], [20], [17], [21]) have been proposed to solve the KL divergence problems with only linear constraints, and among them, the iterative proportional fitting (IPF) ([18], [17]) is a simple algorithm for solving the problem.

The connection between the OD traffic matrix estimation problem and the I-projection (or the IPF algorithm) is that the OD flow X is a component-wise non-negative vector, so it can be converted into a probability function after scaling assuming the total traffic volume is known, and the observation equation (1) defines the constrained convex set naturally. Given a starting point (usually just the previouse OD estimate), below is the pseudo code of the IPF algorithm to populate the current traffic matrix estimate:

Algorithm 1 IPF Algorithm in Traffic Matrix Estimation

Given (1) a starting value μ,
  (2) the current routing matrix A at time t,
  (3) the observation Y which is $Y = AX$;
for $k = 1, \cdots, K$ or till converge **do**
  for $j = 1, \cdots, J$ **do**
    $\alpha = \sum_i A_{j,i}\mu_i / Y_j$
    $\mu_i = \mu_i / \alpha$ for all i with $A_{j,i} > 0$
  end for
end for
return μ.

The above algorithm works for both routing matrices with fractional and whole numbers. When applying the algorithm later, we will replace A and observation vector Y with their augmented versions to accommodate the active measurements.

III. Network Traffic Modelling

A. A Dynamic Network Traffic Model

To model the OD flows, some previous efforts have chosen to assume that an OD flow is either Poisson or Gaussian. In this work we focus on the conditional random variable $X^{(t+1)}|X^{(t)}$, which is assumed to be a Gaussian distribution (We define η shortly.)

$$X^{(t+1)}|X^{(t)} \sim N(X^{(t)}, \eta^{(t)}\text{diag}(|X^{(t)}|)), \tag{3}$$

where the absolute value in the covariance matrix is introduced to ensure mathematical accuracies. In reality, the OD flow $X^{(t)}$ is always non-negative; as will be seen below in our approach the estimate of $X^{(t)}$ based on the model is always guaranteed to be non-negative.

In the model, the unknown parameter $\eta^{(t)}$, characterizes the variability of the network traffic at time t. At a particular time t, the same $\eta^{(t)}$ applies to all OD flows; however $\eta^{(t)}$ varies vary over time and thus accounts for volatility of network traffic. We assume that these parameters are bounded by a constant $\eta > 0$, that is, $\eta^{(t)} < \eta$. In short, our model is spatially homogeneous but temporarily inhomogeneous. Empirical studies based on our dataset in Section V suggest that $\eta^{(t)}$ usually takes on small values (Fig. 2 (c)).

We assume a linear mean-variance relationship on the conditional distribution of $X^{(t+1)}$ given $X^{(t)}$. In previous work, [4] and [11], a power-law mean-variance relationship (with linear as a special case) was used to model the marginal distribution of $X^{(t)}$. We make our assumption for two reasons. First, this linear mean-variance relationship intuitively accounts for the phenomenon that large flows have large variations; we validate this assumption later in Section V. Second, the network traffic is non-stationary. It is our intention to use the conditional relationship, which essentially results in a non-stationary network traffic model, to capture certain aspects of non-stationarity of network traffic and to track network dynamics.

Another important assumption of the model is that the covariance matrix of this conditional distribution is diagonal, implying that all OD flows are independent. Intuitively, there are few reasons for OD flows to be correlated as traffic sources and sinks are independent (such as independent end users or web servers). Correlations can arise when some web servers are very popular, and thus many users send and receive packets from the popular nodes. Similarly, OD flows that share a source can be correlated. Incorporating such correlations would lead to a block diagonal autocovariance matrix. But any attempt to model such dependencies explicitly requires adding a large number of parameters in the traffic model. Two problems can arise when attempting parameter estimation for such a model: a) we can end up fitting the noise instead of the true signals; and b) a large amount of data is needed to do parameter estimation. An in-depth exposition of this topic can be found in [22]. We prefer to opt for a simple formulation, as specified in (3), since simple models are always preferable as long as they yield accurate estimates. The goal in modeling is to capture essential features of the traffic that lead to accurate estimates. A model need not incorporate all properties of the traffic, if they are not essential to the estimate being sought. Our results show that the model we have chosen leads to estimates that are well below the 10% target.

One motivation for this conditional model is to introduce a time series structure between consecutive time slots of an OD flow. This conditional model enables us to combine *past* traffic matrix estimates and the current link counts together to produce an estimate of a *current* traffic matrix. There are many ways to incorporate previous estimates, such as using it as an initial condition for an optimization procedure. To populate our traffic matrix we will use an estimate based on the expectation of the current random variable conditioned upon the link constraints and the additional measurements we obtain, given the immediate previous estimate.

In this approach, the transitions of the traffic matrix from one time interval to the next are controlled by the parameter η and small η's imply that these transitions are not excessive. Clearly the validity of non-excessive transitions depends upon the time scale the matrix intends to be used for. In our case, we make estimates of a traffic matrix every 10 minutes. Our model is intended to capture *local* behavior, that is, "local" in a temporal sense (over a short window of time). We realize that our model would not be an accurate description of traffic over long timescales such as many hours or days. However, our intent is to capture the transitional behavior of a traffic matrix from one (short) interval to the next.

This modeling assumption has an alternate interpretation as a state-space model, which is used to describe internal unobservable states that evolve over time. The relationship between the observable and unobservable variables is usually specified as linear functionals typically with noise terms. In terms of state-space system notations, our model can be rewritten as follows:

$$X^{(t+1)} = X^{(t)} + \sqrt{|X^{(t)}|}\epsilon^{(t)} \qquad (4a)$$

$$Y^{(t+1)} = AX^{(t+1)}, \qquad (4b)$$

where the observable link traffic $Y^{(t)} \in \mathcal{R}^J$ is a linear function of the unobservable OD traffic $X^{(t)} \in \mathcal{R}^I$ at time t. The routing matrix A, relating the unobservable states and observations together, is a known sparse matrix (i.e., with many zero entries). The errors $\epsilon^{(t)}$ are identical independent distributed normal random variables:

$$\epsilon^{(t)} \sim N(0, \eta^{(t)}), \qquad (5)$$

where, as discussed earlier, $\eta^{(t)}$ $(< \eta)$ is an unknown parameter quantifying the dynamics of the underlying OD traffic. We would like to comment that there is no need to estimate $\eta^{(t)}$'s in the OD estimation and flow selection. They are needed to capture variability, however when computing a traffic matrix estimate using $E(X|Y)$, the actual Y's become fixed and the $\eta^{(t)}$'s cancel out. This will become clear in the proof of Theorem 1 (see Appendix).

B. Error Metric

Before proceeding to our methods, we first introduce our error metric. It will be used as the objective in the optimization problem for estimating OD traffic, and will also assist in selection of traffic flows to measure. We propose to use a variant of the mean square error (MSE) as the error metric to assess the performance of an estimator. Let $\hat{X}$ be an estimate of the unknown OD traffic X, then the MSE of $\hat{X}$ is defined as

$$\text{MSE}(\hat{X}, X) = ||\hat{X} - X||^2.$$

One drawback of the MSE metric is that it is not invariant to the traffic volume changes. As our network traffic model is non-stationary: the mean total traffic volume may increase or decrease over time; hence, it is not reasonable to compare estimation performance at different time locations. Model (4) postulates that the variance is proportional to the mean (conditioned on the traffic in the previous time slot). This relationship is used to devise the following scaled mean square error (sMSE) metric to mitigate the problem of MSE:

$$\text{sMSE}(\hat{X}, X) = \frac{||\hat{X} - X||^2}{||X||_1} = \frac{\sum_i (\hat{X}_i - X_i)^2}{\sum_i |X_i|}.$$

Since we will use this sMSE metric for assessing performance, it is also used as our objective function in searching for the OD flow estimate. It is important to note that the scaling factor is a quantity that does not involve $\hat{X}$; hence, in effect, minimizing the sMSE is identical to minimizing the MSE metric.

Another justification of the sMSE metric is based on the model we are using. Suppose at time t the true OD flow $X^{(t)}$ were known and no link measurements were available, then the most sensible estimate for $X^{(t+1)}$ would be just $X^{(t)}$ itself. We can then calculate that the expected sMSE error under such a scenario as

$$E\left(\text{sMSE}(X^{(t)}, X^{(t+1)})\right) \approx \eta^{(t)}.$$

This indicates that if we start from the previous true OD flow $X^{(t)}$, the expected sMSE error will approximately be $\eta^{(t)}$, which quantifies the variability of the traffic according to our model. So sMSE is meaningful to our model. In practice, of course the previous true traffic is unknown to us, so two factors are at play regarding the sMSE of the estimation. On one hand, we do not know $X^{(t)}$ but only its estimate $\hat{X}^{(t)}$. On the other hand, we can include (link or other extra active) measurements to better the OD traffic estimation. The final expected error metric will be influenced by both factors.

Other error metrics have been used in the past (e.g., [10]); a common one is the relative error defined as:

$$\text{Rel-Error}(\hat{X}_i, X_i) = \frac{|\hat{X}_i - X_i|}{|X_i|}.$$

It has been shown that in real networks, roughly 95% of the total load in the traffic matrix is carried by less than 1/2 or 1/3 of the flows [9]. Moreover, the volume of flow in these OD pairs can span several orders of magnitude. Hence, there are typically very small traffic flows that generate extremely large relative errors and others are essentially irrelevant. Our scaled MSE metric avoids this drawback, and works well as a performance metric for both large and small flows. We point out that in practice, the relative error is a useful measure to network operators as it is intuitively appealing; thus we also report on this metric in the results section.

IV. Partial Measurement Approach

A. Incorporating Measurements

One of the central ideas in our method is that of coupling the inference activity with the direct measurement of a small number (possibly just one) of OD flow. To do this, it would be necessary for flow monitors to be universally deployed throughout a network. One might ask, if flow monitors are deployed everywhere, why not just measure the traffic matrix entirely? In [2] the authors outline the overheads involved for both centralized and distributed versions of full direct measurement. In both cases, the communications cost (information being shipped to a central Network Operations location) remains very high. For these reasons, it is interesting to consider more lightweight uses of direct OD flow measurement.

A recent discovery illustrated that seemingly high dimensional network OD traffic actually resides in a space of much lower dimensional [23]. This provides compelling intuition for a partial measurement approach, since it implies that there is potential to learn a great deal about all the flows by only measuring a few of them. In practice, it is challenging to get a low rank representation because the network traffic is volatile; hence, the representation changes over time. Our proposed partial measurement approach is to use only a few active measurements to obtain some vital information to explore this low dimensional space dynamically. We contend that the original ill-posed problem becomes more well-posed even if only a tiny fraction of OD flows are measured directly at each time point, because the OD flows measured in recent time slots remain informative due to temporal correlations in OD traffic.

The partial measurements can be incorporated into our model as follows. Let $M^{(t)}$ be a $k \times I$ measurement matrix at time t. Each row of this matrix is a unit vector: it contains e_i' if $X_i^{(t)}$ measured. We append $M^{(t)}$ to obtain the augmented routing matrix $A^{(t)}$:

$$A^{(t)} = \begin{pmatrix} A \\ M^{(t)} \end{pmatrix}.$$

Then the augmented observation vector $Z^{(t)} = A^{(t)}X^{(t)}$ is the total observation available at time t. The first J entries in this vector contain the link counts while any additional entries contain the measured OD flows. In this paper, k, the rank of $M^{(t)}$ is preset, i.e., the number of OD pairs to be measured is determined. It is possible to treat it as a tuning parameter in different scenarios, however we find excellent performance when $k = 1$ and hence there is little motivation to explore other values (at least for the dataset we study).

Equation 4b is now replaced so that our new system equations, with the measurements incorporated, are given by

$$X^{(t+1)} = X^{(t)} + \sqrt{|X^{(t)}|}\epsilon^{(t)} \tag{6a}$$

$$Z^{(t+1)} = A^{(t)}X^{(t+1)}. \tag{6b}$$

Our proposed PamTram approach is shown in Algorithm 2. The initial traffic matrix $\hat{X}^{(0)}$ is set to be component-wise vector 1. This initial choice of traffic matrix is not very important as the algorithm will quickly adjust itself to the right region. The first step is to measure a small set of well selected OD flows using monitoring equipment. Step 2 of the procedure corresponds to the usual optimization problem for traffic matrix estimation, and many of the previous methods could possibly be applied here. We will provide a fast implementation of an existing method. The challenge in Step 3 is to determine which informative OD flows to measure. We will tackle these two questions separately in the following two subsections.

Algorithm 2 Summary of the PamTram approach

Initialization: Set $\hat{X}_0 = \mathbf{1}$
for each time interval t **do**
  1. Measure OD pairs selected at step $t-1$;
  2. Estimate $X^{(t)}$ based on data $Z^{(1)}, \cdots, Z^{(t)}$
  3. Determine OD pairs to measure at $t+1$.
end for

In the following section, we will show that the IPF algorithm can be used as an approximation method to solve the two optimization problems (Step 2 and Step 3), that is both very accurate and very practical. It is practical because the implementation of an IPF algorithm is much faster than indirect solutions since it avoids matrix inversions (typically needed in Step 2), and integrating over long periods (typical in Step 3).

The PamTram algorithm can be viewed as starting from a maximum entropy estimation in the following sense: after normalization by the total OD traffic (which is naturally done during IPF), the OD traffic problem is equivalent to finding the *I*-projection to the linear space of probability distributions from a uniform distribution. This is intuitively appealing because a maximum entropy estimate implies that we start knowing nothing and thus need no prior knowledge. Hence our choice of initial traffic matrix $\hat{X}^{(0)}$ is not important.

B. IPF and Minimum Mean Square Error Estimation

Now we state our workhorse algorithm, and its properties for estimating OD traffic when both link traffic counts and some direct measurement information are gathered (Step 2 in Algorithm 2). This IPF algorithm was used first by Cao et. al. [4] as a post-processing step in their OD estimation algorithm based on a Gaussian OD traffic model.

Since our goal is to be able to estimate the traffic matrix on the timescale of minutes (e.g., 5 minutes for SNMP reporting, or 10 minutes as in our measurement) , we seek a fast online solution. The IPF solution is an appealing option because: 1) it is easy to implement; 2) it converges in exponential rates (cf. Liang et. al. [12]), and is thus very fast in practice. The IPF algorithm can be run satisfactorily in the order of $O(IJ)$ with a preset finite number of iterations. The starting point at time t is determined by $\hat{X}^{(t-1)}$, the estimate obtained from the previous step. It is reasonable to expect the starting value to be in a small neighborhood of the OD traffic $X^{(t)}$ to be estimated; this further speeds up the convergence rate.

The following theorem justifies the use of IPF for the OD flow estimation under the dynamic traffic model from a statistical viewpoint.

Theorem 1: For the network dynamic model (4), conditioning on $X^{(t-1)}$, if the mean vector μ (i.e., $X^{(t-1)}$) is assumed known, then the IPF estimate of $\hat{X}^{(t)}$ is approximately the minimum MSE estimate.

The IPF algorithm was first proposed in [18]. in the context of fitting contingency tables with fixed margins. It was conjectured there that IPF approximately minimizes a (weighted) least square objective function (but there was no proof). [17] showed later that IPF minimizes actually the KL-divergence which is only an approximation to the (weighted) least square function as seen in the above theorem. Our proof is quite similar to the proof used in [1], but extends their result to a conditional Gaussian model. This theorem implies that the iterative proportional fitting (the *I*-projection estimate) approximately gives the minimum MSE estimate when $\mu = X^{(t-1)}$ is known. In a real problem, $X^{(t-1)}$ is unknown hence replaced by the previous estimate $\hat{X}^{(t-1)}$. Another advantage of the IPF algorithm that the resulting OD flow estimate is positive, which is not guaranteed by the minimum MSE error estimate.

Alternatively, the IPF algorithm can be justified as an approximation to the true minimal MSE estimate given all past observations. The true minimum MSE estimate of $X^{(t)}$ is the conditional expectation of $X^{(t)}$ given all observations:

$$E^{(t)} = E\left(X^{(t)} \,\middle|\, Z^{(1)}, \cdots, Z^{(t)}\right).$$

The computation of such a quantity is very high: it involves an integration over all past data points. To avoid this cost, we can consider the following one-step approximation instead:

$$\begin{aligned} E^{(t)} &= E\left(E\left(X^{(t)}|X^{(t-1)}, Z^{(t)}\right) \middle| Z^{(1)}, \cdots Z^{(t)}\right) \\ &\approx E(X^{(t)}|\hat{X}^{(t-1)}, Z^{(t)}). \end{aligned} \tag{7}$$

This approximation is valid if the last parameter estimation $\hat{X}^{(t-1)}$ is in the neighborhood of the true traffic $X^{(t-1)}$; then the IPF algorithm can be used to compute this conditional expectation approximately by starting from $\hat{X}^{(t-1)}$.

C. Measurement Selection Scheme

1) Motivation: We now address the issue of how to select the OD flows to measure in each time interval (Step 3 in Alg. 2). The idea is to choose a scheme that will select the most informative of the unobservable flows. Clearly, the choice has to be made based solely on the observable variables. We focus on selecting a single OD flow because even just measuring one OD flow per interval provides excellent performance. Our ideas here could be generalized to selecting a few flows, driving errors yet further down.

First, let us consider what an optimal solution would suggest and entail. Suppose X is a multivariate random variable (not necessarily normal) with: $E(X) = \mu$, and $Var(X) = \Sigma$, where both μ and Σ are known (or can be estimated). Then the minimum MSE predictor for X is just μ with the MSE error

$$E||X - \mu||^2 = \text{trace}(\Sigma).$$

Hence ideally, we would like to select an OD pair such that the resulting conditional covariance matrix given all observations

$$\Sigma^{(t)} = \text{Var}\left(X^{(t)} \,\middle|\, Z^{(1)}, \cdots, Z^{(t)}\right) \tag{8}$$

has the smallest trace. In other words, our task is to select the observation matrix $M^{(t)}$ such that the trace of the conditional variance is minimized,

$$M^{(t)} = \arg\min_{M^{(t)}} \text{trace}\Sigma^{(t)}.$$

Intuitively this means we want to select the OD flows such that our estimation error was minimal. However, this approach is not attractive because the computation of $\Sigma^{(t)}$, involving integration over all past observations, is too costly.

Similarly to the approach for producing an approximation that we mentioned in (7), we can also imagine using the same type of approximation here for the conditional covariance in (8):

$$\text{Var}\left(X^{(t)} \,\middle|\, X^{(t-1)} = \hat{X}^{(t-1)}, Z^{(t)}\right)$$

However this remains difficult to compute because in general, for any random variables C and D, we have $\mathrm{Var}(C) = \mathrm{Var}(E(C|D)) + E(\mathrm{Var}(C|D))$, that in our case translates to,

$$\mathrm{Var}\left(E\left(X^{(t)} \middle| Z^{(t)}, X^{(t-1)} = \hat{X}^{(t-1)} \right) \middle| Z^{(1)}, \cdots, Z^{(t)} \right).$$

which remains computationally difficult to approximate.

2) Randomized Decision Rules: Since using $\Sigma^{(t)}$ to choose the optimal OD flow to measure is too computationally intensive, we develop instead heuristic randomization schemes motivated by game theory. Consider for a moment a uniform randomization scheme in which, at each time step, each OD flow is selected for measurement with equal probability $1/I$. The following theorem bounds the one-step error performance (the error made from one interval to the next) assuming uniform random sampling of flows.

Theorem 2: Let $\omega^{(t-1)}$ be the sMSE error at step $t-1$,

$$\omega^{(t-1)} = \mathrm{sMSE}\left(\hat{X}^{(t-1)}, X^{(t-1)} \right).$$

Assume no link measurements Y are made, and only one OD pair is selected for measurement by uniform random sampling, then the expected value of $\omega^{(t)}$, the error metric of $\hat{X}^{(t)}$, is approximately bounded by

$$E(\omega^{(t)}) \leq \frac{I-1}{I}\omega^{(t-1)} + \eta^{(t)} \leq \frac{I-1}{I}\omega^{(t-1)} + \eta,$$

where I is the number of total OD pairs. When t goes to infinity, $I\eta$ is an upper bound of the expected error.

In the theorem, the expected value of the next step error metric is bounded by the sum of two parts: the first is the reduction in previous error thanks to the additional measurement; and the second part (η) comes from the intrinsic variability (uncertainty) of the traffic itself. The theorem implies that when t grows, the expected error will be bounded regardless of where we start.

This theorem is a comforting result in the sense that using a uniform randomization scheme is not going to lead to an error metric that can grow without bound. Since this case corresponds to picking a flow arbitrarily, it can be viewed as a worst case bound. Indeed, as we will see, all of our alternative randomization schemes produce smaller errors than the uniform randomization scheme.

In practice, link measurements $Y^{(t)}$ are obtained, then the residual of the parameter estimate at time t is

$$R^{(t)} = X^{(t)} - E\left(X^{(t)} \middle| X^{(t-1)} = \hat{X}^{(t-1)}, Y^{(t)} \right). \quad (9)$$

In order to reduce the sMSE (or equivalently the MSE), one should measure the OD pairs with the largest absolute residual(s). Note $\hat{X}^{(t-1)}$ is only an estimate. We can view the task of picking the OD flow with the largest residual as a two person game. One player is the traffic generator and the other is the operator who is trying to guess the traffic volumes. At each move, the traffic generator changes the traffic volumes, and the operator's move is to guess the traffic. We assume that the traffic generator knows the strategy of the operator and tries to select traffic volume levels so as to confuse the operator as much as possible. Suppose the operator has a 0-1 loss function (the loss is 1 if it guesses correctly and 0 otherwise). The operator's goal is to maximize the probability of picking the largest residual, i.e.,

$$L(X^{(t-1)}, i) = 1\left(R_i^{(t)} = \max_j R_j^{(t)} \right).$$

This corresponds to picking the largest random number amongst a set, and hence we call our game a *pick-largest-random-number* game. The next theorem shows that random guessing, i.e., the *uniform* randomization scheme above, is in fact the minimax rule, and is thus the best option in this game scenario.

Theorem 3: The uniform random sampling ($p(i) = 1/I$) is the minimax decision rule of the pick-largest-random-number game with a 0-1 payoff (loss) function.

In reality, the "traffic generator" player is not really an intelligent adversary. Although there is variability in traffic, there is also a good deal of temporal correlation, and many flows vary slowly over short time scales. Hence, choosing OD pairs uniformly is likely to give poor results since the information in the previous traffic estimate is not exploited. In fact, since $\hat{X}^{(t-1)}$ is likely to be close to the true traffic state $X^{(t-1)}$, we should be able to guess reasonably well the moves of the traffic generator player, by relying on previous estimates. If we assume $X^{(t-1)} = \hat{X}^{(t-1)}$, then $R^{(t)}$ is a mean zero normal random variable with variance (independent of $Y^{(t)}$)

$$\Lambda^{(t)} = \Sigma^{(t)} - \Sigma^{(t)}A'(A\Sigma^{(t)}A')^{-1}A\Sigma^{(t)}, \quad (10)$$

where $\Sigma^{(t)} = \eta^{(t)}\mathrm{diag}(\hat{X}^{(t-1)})$, and the probability of $R_i^{(t)}$ being the largest residual in absolute value is

$$Q(i) = P\left(|R_i^{(t)}| = \max_j |R_j^{(t)}| \right). \quad (11)$$

In this alternate game, the operator now knows the strategy of the traffic generator in the sense that is has a model of the traffic. A good strategy for the operator would be to pick an OD flow whose probability of generating the largest residual is highest. Let $P_{maxen}(i)$ denote our strategy, i.e., the probability of picking flow i. We should choose $P_{maxen}(i) = Q(i)$ for the following reason. If the operator has a negative log loss function, and the distribution Q is assumed to be known, then it is well known that the solution that minimizes this loss function is given by the maximum entropy solution, i.e.,

$$P_{maxen} = \arg\min_P - \sum_i \log P(i) \log\left(\frac{Q(i)}{P(i)} \right) = Q.$$

Hence, we call this randomization scheme *maxen*.

The uniform and maxen randomization schemes approach the measurement selection from two opposing points of view. On one hand, the uniform scheme ignores the knowledge about the network from previous time intervals. On the other hand, the maxen randomization scheme is based on the rationale that the system will not change much. Real networks exhibit both behaviors: sudden changes and sustained smooth transitions. Hence we combine these two schemes to produce a scheme that sometimes allows departures from the base model. Let $\alpha \in (0, 1)$. A weighted minimax randomization is defined as:

$$P_{wMaxen} = \alpha P_{uniform} + (1 - \alpha) P_{maxen}.$$

Here we assume that the parameter α is preselected, and that it could be tuned for particular networks. We usually set it as a relatively small number, such as 0.2, to favor the existing estimated models more.

3) Implementation of Decision Rules: For these randomization schemes, the uniform is easy to realize, but the implementation of the maxen randomization rule is difficult because the probabilities defined in (11) are hard to obtain. Instead of computing these probabilities explicitly, the maxen scheme can be implemented by generating multivariate normal random numbers whose covariance matrix is that specified in (10). Let $\mu = \hat{X}^{(t-1)}$, and $X^{(t)} \sim N(\mu, \eta^{(t)} \text{diag}(\mu))$, then

$$X^{(t)} - \Sigma A'(A \Sigma A')^{-1}(AX^{(t)} - A\mu) - \mu \qquad (12)$$

is a mean zero multivariate normal random variable with covariance matrix $\Lambda^{(t)}$. Our method is thus to use this distribution to generate I random numbers (recall I is the number of OD flows), and then find the index of the largest one. But (12) requires the inversion of the matrix $A\Sigma^{(t)}A'$, which is computationally expensive. Again, the result from Theorem 1 shows that the IPF algorithm can be used to approximate

$$X^{(t)} - \Sigma A'(A \Sigma A')^{-1}(AX^{(t)} - A\mu).$$

This can be solved approximately by using $X^{(t)}$ as a starting point and applying IPF to find a solution that fits the link constraint $Y = A\mu$. Thus a maxen randomization algorithm can be devised as follows:

Algorithm 3 Maxen Randomization Algorithm

Let $\mu = \hat{X}^{(t)}$ and $y = A\mu$;
1. Generate $\tilde{X} \sim N(\mu, \eta^{(t)} \text{diag}(\mu))$;
2. Project $\tilde{X}$ onto $\{X | y = AX\}$ to get $\check{X}$ using IPF;
3. Pick the jth OD flow if $j = \arg\max_i |\check{X}_i - \mu_i|$.

Note that we needn't be concerned about our choice for the parameter $\eta^{(t)}$ in this procedure. As mentioned earlier, during the computation of $E(X|Y)$, the $\eta^{(t)}$'s cancel out (see proof of Theorem 1).

In summary, the total computational cost of PamTram is at most two IPF algorithm costs. If uniform sampling is used, we don't need Algorithm 3, and the IPF gets executed once. For the *M*axen solution we use IPF twice. Since the IPF computation is light, we believe our solutions can scale to larger networks.

D. Practical Issues of Flow Collection

There are some issues related to the practically of our proposed partial monitoring scheme. We realize that because in practice the flow monitor is attached to a link, when we turn it on, we will in fact capture all the flows traversing that link. However, in this paper we study the case of measuring only a single OD flow to understand the impact of this idea. Our goal is to understand, in general, how much flow measurement is needed to obtain accurate traffic matrices. Since in practice we have more than one OD flow, the errors will be lower than what we calculate using only one OD flow.

The other practical problem is that when the flow monitor resides on a router, activating and deactivating flow measurements requires updating router configurations (e.g.,, once every 10 minutes). This could be viewed as high management overhead by network operators. There are two ways to avoid this problem. First, it is possible to imagine alternate implementation scenarios in which, for example, flow monitors are left on all the time, and the results are either stored at the monitor or in a collection server residing in a PoP. Then the network operations center could selectively pull flow records, a few at a time, as directed by our selection schemes.

A second option is to select the measurement schedule a few hours in advance thus providing the network ample time to disseminate and schedule the monitoring activities, that could be loaded into monitors in batches. We consider a variation of our randomization schemes is which the OD flows to measure are selected 24 hours in advance. The idea is that a flow selected for measurement at 2:10pm on one day, is actually measured at 2:10pm the next day. The rational for such an approach comes from both the observation of strong daily periodicity (as in Fig. 1) which shows that traffic is generally similar from one day to the next at a particular time of the day, and from [2] in which the authors illustrate this notion more precisely using fanouts. We call this a *Latent* scheme. Note a latent scheme is merely a scheduling approach that needs to be combined with a randomization rule. We evaluate both *Latent(maxen)* and *Latent(wMaxen)*.

V. Experimental Environment

A. The Data

Our data comes from Sprint's European backbone that is made up of 12 Points of Presence (PoPs) and 18 inter-PoP links. The network OD traffic information was collected by turning on Netflow (version 8) on all the Cisco routers. This version of Netflow uses a sampling scheme of monitoring 1 out of every 250 packets. The data was aggregated into PoP level flows at a time granularity of 10 minutes (i.e., average number of bytes sent between PoP pairs during each 10 minute window). The data collection interval of 10 minutes was chosen to mitigate possible measurement errors. This is the same dataset used in [8] hence some of our error statistics can be directly compared to those in [8]. To avoid inconsistencies between the link traffic and OD traffic, the link measurement data are derived from the flow level measurements X; this guarantees that the traffic matrix X, the routing matrix A and the link traffic counts Y are all in agreement with each other. This approach is well justified in [10].

We now show some behaviors of this OD traffic data that, although they have been pointed out before, are included here for completeness. Fig. 1 shows two time series plots of OD traffic flows selected because they represent common

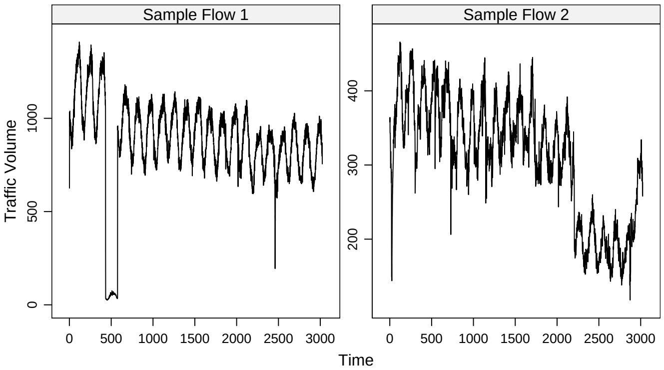


Fig. 1. Two sample PoP level OD traffic flows.

behaviors. [1] The first one shows strong periodicity (very common in OD pairs [23]). The strong periodicity of OD traffic also induces strong periodicity in observed link traffic. The period of the traffic is exactly one day, while a weekly period can also be seen over a longer time frame. Both samples illustrate that sharp changes, different from the diurnal cycles and from the local noise, can occur. These can occur for reasons such as router failures, the addition of new customers, or the removal of previous customers.

We now use this data to validate some of the assumptions used in our model. The conditional linear mean-variance relationship implies that we have

$$U_{t,i} = \left(X_i^{(t)} - X_i^{(t-1)}\right) \Big/ \sqrt{X_i^{(t-1)}} \sim N(0, \eta^{(t)}).$$

Even though $\eta^{(t)}$ varies over time, it is reasonable to assume that it is continuous. Then we can estimate its value within each small moving window. Fig. 2 shows two QQ-plots at two different time points, and a time plot of the $\eta^{(t)}$ estimation. We use only the upper 90% of the traffic load to generate all three figures, since we care less about faithfully representing the very small flows than the larger ones constituting the majority of the traffic. The Q-Q plots are produced based on all $U_{t,i}$ within a 50-minute window, i.e., 5 intervals. These two Q-Q plots are chosen because of their representativeness; data in other time windows show similar features. Fig. 2(a) is drawn based on data points in the time window 1-5, and (b) is in time window 2000-2005, which is the region with the highest spike (Fig. 2(c)). From both plots, we can see that the $U_{t,i}$ is very close to a normal distribution but with a longer tail. Fig. 2 shows the estimated $\eta^{(t)}$ over time. Because $U_{t,i}$'s have a longer tail than normal, a robust estimate of $\eta^{(t)}$ based on absolute moment ([24]) is used:

$$\hat{\eta}^{(t)} = \left(\frac{\sum_i |U_{t,i}|}{0.799 \times I} \right)^2,$$

where $E(|V|) = 0.799$ for $V \sim N(0,1)$. From the plot, we can see that the values of $\eta^{(t)}$'s mostly oscillate around 1, which is very small given that a medium traffic flow may take a value of several hundreds or thousands. There are occasional spikes in the figure – the most obvious one corresponds to the sudden traffic changes occurring around time slot 2000. Overall, the $\eta^{(t)}$ is well bounded except a few spike points. The plot shows that the conditional linear mean-variance relationship is a good approximation to the raw data.

[1]The traffic volume has been multiplied by a randomly selected number for reasons of confidentiality.

B. Partial Measurement Schemes

We tested PamTram using a number of partial measurement schemes to the Sprint PoP network data, including the *uniform*, *maxen*, *wMaxen*, *Latent(maxen)*, and *Latent(wmaxen)* schemes. For the *wMaxen* and *Latent(wMaxen)* versions, the weight parameter α is set as 0.2 to favor the *maxen* randomization scheme. Our experiments show that the scheme is not very sensitive to the choice of α. In order to better evaluate the performance of these randomization schemes, we also implemented an *oracle* scheme. The oracle has full knowledge of the true OD traffic and thus the largest residual can be precisely selected. In other words, we select the flow that results in the smallest scaled MSE error in the next parameter estimate. We can do this since we have the measured traffic matrix at our disposal. Although this cannot be done in practice, it provides a means of assessing how far our schemes are from a sort of optimal (full knowledge) behavior. We have not included the results for *Latent(maxen)* because they are very similar to those of *Latent(wMaxen)* and due to lack of space. In almost all of our evaluations, we measure only one OD flow in each 10 minute measurement interval.

C. Comparison to Kalman Filter method

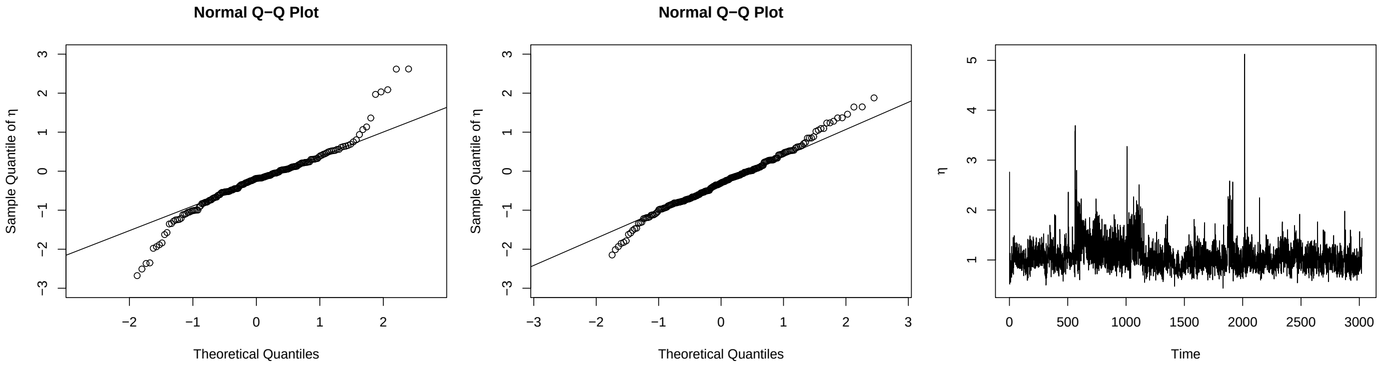
We compare PamTram to the Kalman filter based method that was proposed in [25] and fully evaluated in [8]. We briefly summarize the essential ideas here for completeness. We choose to compare our solution to the Kalman method because that method resembles our solution in that it also uses a state space model coupled with partial flow measurements.

The state space model in the Kalman method is

$$\begin{cases} X_{t+1} &= CX_t + W_t \\ Y_t &= AX_t + V_t \end{cases} \tag{13}$$

where C is the state transition matrix, W_t is the dynamic traffic noise process and V_t is the measurement noise process. The diagonal elements of C capture temporal correlations for individual OD flows, while the off diagonal elements of C capture spatial correlations across different OD flows. The Kalman filter is a two-step method that iterates each time interval. It first computes both a prediction for X at time $t+1$ given all the data seen up to time t, namely $\hat{X}_{t+1|t}$, and then computes a modified update when the new set of link counts arrive at time $t+1$, denoted $\hat{X}_{t+1|t+1}$. This latter step provides some of the adaptability of the Kalman method because it estimates the TM using a model capturing temporal and spatial correlations, but modifies it to be in line with the link counts.

To make these computations, the Kalman method needs to calibrate the matrices C, Σ_W and Σ_V. To do this all the flow monitors throughout a network are turned on for a period of



(a) (b) (c)

Fig. 2. Model Validation. (a) and (b): Q-Q plots of $U_{t,i}$ within two chosen 5-point time windows: the first one is based on the time window 1-5 and the second time window 2000-2005; (c) Estimated $\eta^{(t)}$ over time.

24 hours. The 3 matrices are computed using this data via an Expectation Maximization algorithm. Once the model is calibrated, the filter alone is used for producing estimates. In addition, the method uses a background procedure to check when sufficient change has occurred so as to require recalibration of these 3 model parameters. When this happen the flow monitors are turned back on for another 24 hours.

We now point out a critical difference between the Kalman method and PamTram. In the Kalman solution, their rich model is well calibrated at the time of measurement collection. If the model drifts, the out-of-date parameters are not updated until the drift is large enough to be detected. In the PamTram approach, because we measure 1 (or a few) flows at each time interval, the model is essentially continuously being updated. At no single moment is the entire TM collected; however since the OD flows exhibit strong temporal correlations one could hypothesize that this isn't necessary. As we will see below, it appears that continuous update of a small portion of the TM itself, is more powerful than discrete model updates that are complete but spaced far apart in time.

VI. Results

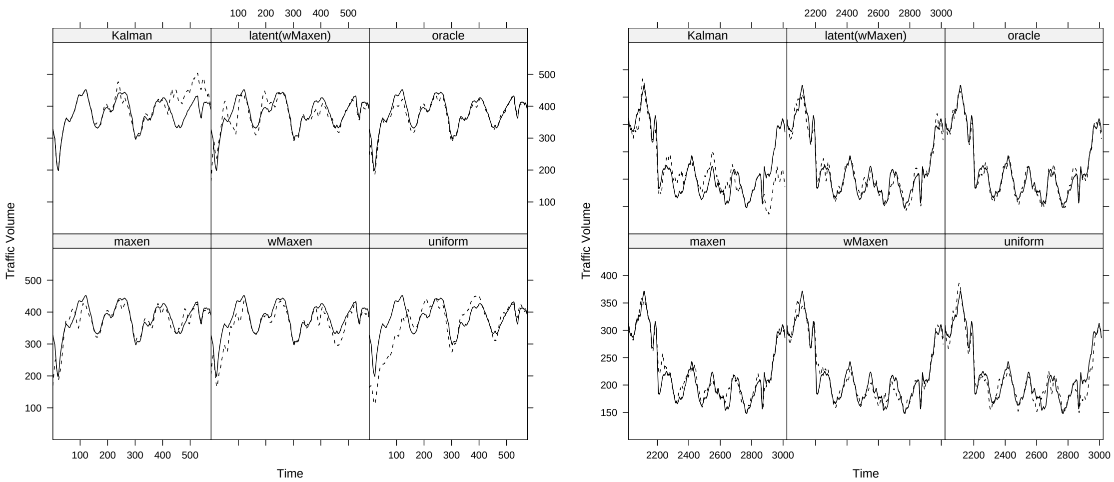
We now evaluate all of our partial measurement methods along with the Kalman based method, with respect to a number of performance metrics including temporal errors, spatial errors, relative errors, overhead and adaptability.

Time plots. We start by viewing some sample time plots of OD flow estimates as shown in Fig. 3. Because the network traffic is very volatile and hard to visualize, the smoothing spline method is used to remove unnecessary spikes of the true traffic while keeping the trend faithful. The same method is applied to estimated OD flows as well. In Fig. 3, we show a sample OD flow along with its estimate (Sample flow 2 in Fig. 1): the first panel shows the starting region, and the second is a region where a large traffic changes occurs. We see that all the methods track the OD flow well. The Kalman has the most difficulty with the changes around time slot 500 (Fig. 3(a)), and around time slot 2900 (Fig. 3(b)). The uniform scheme has the most difficulty in the initial phases of estimation (starting from time 0). All PamTram approaches adapt to the true OD traffic quickly during these two change episodes.

Scaled MSE errors. First we consider temporal errors. By temporal errors we mean that at each moment in time, we compute our error metric over all the flows giving a representative error for that time-point. We see how this scaled MSE evolves over time by viewing it at consecutive time-points, as in Fig. 4. All variants of PamTram drive the estimation error very low, even when only one OD flow is measured per time interval. We see that the majority of the errors are below 5%. This breaks new ground in terms of low error rates. By examining the initial time period of this figure, we can see that the PamTram method converges. That is, regardless of our arbitrary starting point, the errors of PamTram reduce over time and stabilize roughly around 5%. Of course there is variation, but the errors hover around 5%, the noise is small and larger deviations are short-lived. In this sense, we can say the errors ”converge” or ”stabilize”. The Kalman estimates appear more noisy, occasionally producing larger error spikes. The increase in error just before time slot 500 reflects the delay the method incurred until it detected the large change requiring model re-calibration. The Kalman method does reasonably well overall, however the PamTram variants yield lower errors more *consistently* over time (i.e., the errors are less variable). We found similar behavior for many other flows we examined.

Adaptability. To measure adaptability over a number of episodes of change we use the following metric. We check in each flow, if it has more than a 30% jump from one time slot to the next. If it does, we flag this as the start of a jump, and compute the relative errors during the next 5 time slots. We do this for all methods and for most flows (those constituting the top 90% of the load). We can them compute the average relative error during these jump regions for each method. We found that the *uniform* scheme had a 13% error, the *wMaxen* scheme a 10% error, the oracle experienced 6% error and the Kalman method incurred a 16% average relative error rate.

Spatial Errors. The spatial errors give a different view of the errors in OD flow estimation. By spatial errors we mean that one error is computed per flow since the summing



(a) (b)

Fig. 3. A zoomed-in sample OD flow: true data (solid line) and estimates (dotted line). (a) the initial stage; (b) a region where large traffic changes occur.

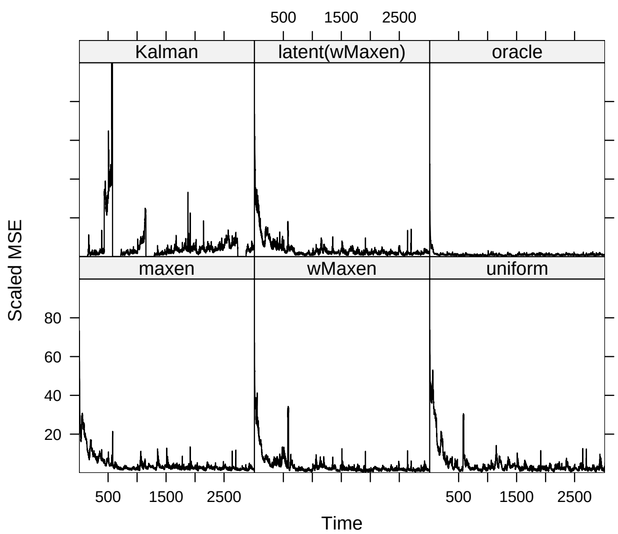


Fig. 4. Evolution of scaled MSE over time.

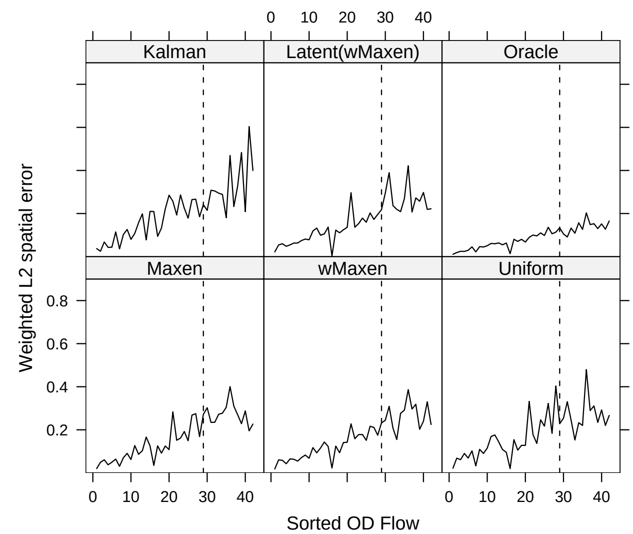


Fig. 5. Spatial error metric versus average flow size (decreasing order). The verticle line indicates the 80% quantile.

operation is done over time. This gives a summary error per flow over its lifetime. The ensemble of these errors illustrates the different errors experienced by different flows. We use the weighted L2 norm proposed in [8] to measure the spatial error:

$$d_{\text{spatial}}(i) = \sqrt{\sum_{t=1}^{T} \left(\hat{X}_i^{(t)} - X_i^{(t)}\right)^2 \Big/ \sum_{t=1}^{T} \left(X_i^{(t)}\right)^2}.$$

Fig. 5 shows the ensemble of spatial errors across large OD flows. The OD flows in the plot are sorted in decreasing order according to their total traffic volume. The OD flows included in the plot constitute 90% of the total load. The vertical bar represents the 80% cut-off point (i.e., all the flows to the left of the bar constitute 80% of the load). We see in this plot that OD flows with smaller average size tend to have larger errors. This is a well known phenomenon and is consistent with results in almost all previous traffic matrix estimation papers. More importantly, we observe very small spatial errors for the majority of the traffic. All of these partial measurement schemes perform reasonably closely to the oracle one. For comparison, we compute the average spatial errors for 90% of the total load. The Kalman method had the highest average spatial error: 20.9%, uniform was 18.5%, Latent(wMaxen) 16.4%, maxen 16.8%, and wMaxen around 16.5%. Since we include 90% of the traffic in the plot, there are numerous small flows still included in the set of flows presented here. These are often disregarded in traffic matrix estimation because they are less important and hard to estimate. This thus shows that our methods can handle some of the small flows as well. It appears that the PamTram variants handle the smaller flows better than the Kalman method.

Relative Errors. We now look at the instantaneous relative errors, summing neither over time nor over space, but instead

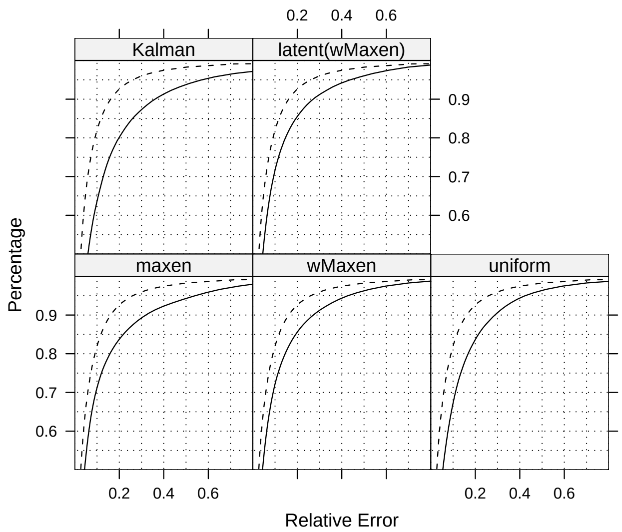


Fig. 6. Cumulative distribution plots of relative errors. The dash line in each panel is that of the oracle scheme.

TABLE I
AVERAGE RELATIVE ERROR, SMSE AND RUNNING TIME

	Avg(relError)	Avg(sMSE)	Time (sec)
Oracle	5.0%	1.13	-
Maxen	9.5%	4.43	0.08
wMaxen	9.0%	3.78	0.06
Latent(wMaxen)	9.1%	3.82	0.06
Uniform	10.5%	4.88	0.04
Kalman	16.9%	6.11	-

just assembling the errors achieved at each interval for each flow. Fig. 6 shows the cumulative distribution function (CDF) of the absolute values of relative errors. In each panel, the dashed line is the oracle and the solid line is the method indicated. The x-axis is the relative error rate, and y-axis the percentage of traffic volume. Note that the y-axis on these plots begins at 50%. We see that none of the schemes are too far off from the oracle, although the Kalman is farthest. For all the PamTram methods, about 75% of OD flows experience an error less than 10%. Since roughly 30% of the flows constitute 95% of the total traffic load, it is likely that the remaining 25% of the flows with errors over 10% are the small ones. We can make this assessment with the help of plots like those in Fig. 5 showing increased errors for smaller flow sizes.

Error Summary. Table I reports the average relative errors, and average scaled MSE for all partial approaches. Again, we report the average relative errors for the largest OD traffic flows that constitute the top 90% of the total traffic load. The average scaled MSE is the simple average of all the scaled MSEs at each time interval. We see that most of the PamTram variants keep their average errors below the 10% target. The main PamTram methods experience an instantaneous average relative error of 9% or 9.5% while the Kalman method sees a 16.9% average error.

Overheads. Table I also reports the run-time or computation time for our proposed randomization methods. The computation of the PamTram is very light; it is a very appealing property of the proposed approach, especially important for the implementation of such online algorithms. The Sprint dataset was processed on a 3.2GHZ computer using the R package [26]. It takes *maxen* approximately 0.08 seconds to generate one traffic matrix estimate per 10 minute window. This includes two iterations of the IPF algorithm. The *uniform* scheme further cuts the running time by roughly half because only one round of the IPF procedure is needed in each time interval (as discussed in Sec. IV-C.3). PamTram is fast because the complexity of an IPF algorithm is $O(IJ)$, and it avoids matrix inversion as is needed by many maximum likelihood estimation or regularization approaches.

The PamTram is lightweight not only computation-wise, but also in terms of measurement overhead. To assess the tradeoff between performance gain and measurement overhead, we use the measurement overhead metric proposed in [8]. Their metric is defined as $\sum_{i=1}^{I} D(i)/(\text{NumDays}*\text{NumLinks})$, where $D(i)$ is the number of days that link i was turned on for flow measurement. This metric, with units of *link-days*, made sense in their context because each time a flow monitor is turned on it remains on for 24 hours. The idea was to count the amount of time a flow monitor is on over many links and days, and to create a ratio so as to compare it to the case of full measurement when all flow monitors are on all the time. The measurement overheads in [8] ranged from 5-30% depending upon the scheme. The Kalman scheme incurred an overhead of 20%. Our scheme is equivalent to the case when one flow monitor is on all the time because at any moment we have one flow being monitored. Hence in terms of this metric, the overhead of PamTram is $1/J$ or roughly 5% since we have a network with $J = 18$ links. This is the overhead when a flow monitor is turned on and collects everything on the link to which it is attached. If the flow monitor could be configured to monitor only a single OD flow, then the measurement overhead would drop to $1/132$ (one over the number of OD pairs) which is less than 1%. This is so lightweight that the tradeoff of measurement versus performance gain is immaterial.

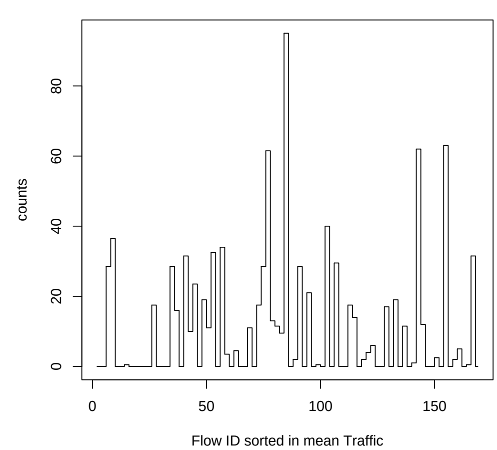


Fig. 7. Frequency of flows selected for measurement for *maxen* scheme. Flow IDs sorted by decreasing mean on x-axis, and number of time flow selected on y-axis.

Flows Selected for Measurement. It is interesting to ask: which flows are being selected for measurement? We counted the number of times each flow was selected over the three week period of our dataset. We plot this data in Fig. 7 for the *maxen* scheme. The flows, whose IDs are on the x-axis, are sorted in order of decreasing mean value. This was done to check if any correlation between frequency of flow selection and mean size exists. Clearly, it is not true that only the largest flows are being selected. We tried this on PamTram variants, and found the same result. We also examined plots of frequency of flow selection versus the standard deviation of a flow to test whether the most variable flows are being selected. Again, there was no clear correlation. It is possible that there are a number of factors influencing flow selection and thus plotting the frequency against a single factor is not revealing. The plot indicates that small flows can also be informative to measure. This is because the information content of each flow is not determined only by the flow volume, but is also impacted by the linear constraints $Y = AX$. The A matrix influences many factors, such as how many links an OD flow traverses, and how many flows share a link. The residual quantity defined in (9) is a function of A and thus indirectly takes these effects into account, when determining which flows are more informative to measure. We leave further investigation of this problem as future work.

VII. Discussions and Future Work

In this paper, we proposed a partial measurement approach for OD traffic matrix estimation based on two key ideas. The first is to use partial flow measurement in a lightweight fashion by only measuring one flow per estimation time interval. We couple this with a dynamic traffic model that allows us to incorporate past information into the current estimate. Such an approach is successful in achieving excellent performance with minimal measurement cost. Measuring one OD flow per time interval brings only a small amount of extra information; however the measurements accumulate over time via the dynamic network model, thus facilitating estimation. Because our model updates continuously over time, no special steps are required when traffic changes occur. Our second key contribution is a scheme for selecting which flow to measure plus the illustration that an IPF algorithm can be used both for approximately this flow selection algorithm and for approximating an MSE error. Because the IPF algorithm is fast and we only measure one OD flow per time interval, PamTram is lightweight both in computation time and in measurement overhead. We thus believe that PamTram has potential to be considered for deployment in operational networks.

We found that while the Kalman method (one of the best methods proposed to date) performs well, all of our PamTram variants consistently outperformed the Kalman method with respect to numerous errors metrics (spatial errors, temporal errors, relative errors and adaptability). In addition, this improved performance comes with a much lower measurement and computational cost. Because the PamTram method uses a nonstationary traffic model it has the potential to adapt to nonstationarity in the traffic more rapidly that previous approaches like [25] or [4] since they rely on local stationarity on timescales of days or hours.

Our interesting and encouraging finding is that for this particular network, low errors could be achieved simply by measuring only one flow each measurement interval. This gives confidence that for general ISP-like networks, it may be sufficient to measure only a very small portion of the TM in order to accurately estimate the entire TM. For large networks with many OD flows, we contend that $k > 1$ may be useful; however we suspect that a good k would still be a very small fraction of the total number of flows. We will explore the scalability of this result in future work.

Other interesting directions for future research include using our dynamic traffic profiles for security purposes. One can detect anomalies by looking for outliers based on models of normal traffic. Such traffic profiles may also be useful for providing enhanced performance for subsets of the total traffic belonging to specific applications (such as VoIP) that may have its own performance and robustness requirements.

Acknowledgments

We would like to thank Sprint Advanced Laboratory for allowing us the access to the PoP network dataset. We would also like to thank Antonio Nucci, Dina Papagianaki, and Anukool Lakhina for preparing the dataset and many helpful discussions. Partial support to Bin Yu and Gang Liang is gratefully acknowledged from the National Science Foundation (CCR-0106656 and FD01-12731) and the Army Rearch Office grant (DAAD19-01-1-0643). Bin Yu was also supported by a Miller Research Professorship from the Miller Institute at UC Berkeley in Spring, 2004.

VIII. Appendix

A. *Proof of Theorem 1*

It is easy to show that the conditional expectation

$$E(X|Y) = \mu - \Sigma A'(A\Sigma A')^{-1}(A\mu - Y).$$

In the above equation, η, the coefficient of A, cancels out and does not appear in the final solution. This conditional expectation is also the solution to the weighted least-square estimate with square root weights in Zhang et. al.. [1]

$$\min \sum_i (X_i - \mu_i)^2/\mu_i \ \text{ subject to } AX = Y.$$

Then similarly, we may borrow the argument pointed out by Zhang et. al.[1] that

$$\begin{aligned} D(X/N||\mu/N) &\approx \sum_i (X_i/N)\,(X_i/\mu_i - 1) \\ &\approx 1/N \sum_i (X_i - \mu_i)^2/\mu_i, \end{aligned}$$

where N is the total traffic. The first approximation is a linear expansion of the logarithmic function, and the second approximation is due to our assumption that $\sum_i \mu_i \approx N$.

B. Proof of Theorem 2

Let K denote the index of the OD pair to be measured; hence, we have $P(K=k)=1/I$. Under the assumption that no any link measurement is obtained, we have

$$\hat{X}_k^{(t)} = \begin{cases} X_k^{(t)} & \text{if } K=k \\ \hat{X}_k^{(t-1)} & \text{otherwise.} \end{cases}$$

Similarly, we define $\tilde{X}^{(t+1)}$ as

$$\tilde{X}_k^{(t)} = \begin{cases} X_k^{(t)} & \text{if } K=k \\ X_k^{(t-1)} & \text{otherwise,} \end{cases}$$

which is the parameter estimate if we start from the true value.

Fix $X^{(t-1)}$ and $\hat{X}^{(t-1)}$ at first, then the expected value of the scaled MSE is

$$E\left(\frac{||\hat{X}^{(t)}-\tilde{X}^{(t)}||^2}{\sum_i \tilde{X}_i^{(t)}}\right) \approx \frac{E||\hat{X}^{(t)}-X^{(t)}||^2}{\sum_i X_i^{(t-1)}}$$
$$= \frac{E||\hat{X}^{(t)}-\tilde{X}^{(t)}||^2 + E||\tilde{X}^{(t)}-X^{(t)}||^2}{\sum_i X_i^{(t-1)}} \quad (14)$$

The first approximation is obtained by the delta method [24], and the second equality holds because

$$E\left(||(\hat{X}^{(t)}-\tilde{X}^{(t)})+(\tilde{X}^{(t)}-X^{(t)})||^2\right)$$
$$= E\left(E\left(||(\hat{X}^{(t)}-\tilde{X}^{(t)})+(\tilde{X}^{(t)}-X^{(t)})||^2 \middle| K=k, X_k^{(t)}\right)\right).$$

Note given $K=k$ and $X_k^{(t)}$, $\hat{X}^{(t)}-\tilde{X}^{(t)}$ are determined, and $\tilde{X}^{(t)}-X^{(t)}$ is a mean 0 multivariate normal random variable. The cross terms disappear after expanding the square term.

For each term in (14), we have

$$\frac{E||\hat{X}^{(t)}-\tilde{X}^{(t)}||^2}{\sum_i X_i^{(t-1)}}$$
$$= \frac{||\hat{X}^{(t)}-\tilde{X}^{(t)}||^2 - \sum_k P(k)(\hat{X}_k^{(t-1)}-X_k^{(t-1)})^2}{\sum_i X_i^{(t-1)}}$$
$$\leq \frac{I-1}{I}\omega^{(t-1)},$$

and

$$\frac{E||\tilde{X}^{(t)}-X^{(t)}||^2}{\sum_i X_i^{(t-1)}} \leq \eta.$$

So in summary, we have

$$E(\omega^{(t)}) \leq \frac{I-1}{I}\omega^{(t-1)} + \eta^{(t)} \leq \frac{I-1}{I}\omega^{(t-1)} + \eta.$$

Note that the above bound actually does not depend on the value of $X^{(t)}$, implying the inequality holds generally.

Let $\gamma^{(t)} = E\left(\omega^{(t)}\right)$. Taking an expectation over both sides of the above inequality, we have

$$\gamma^{(t)} \leq \frac{I-1}{I}\gamma^{(t-1)} + \eta. \quad (15)$$

Based on the above inequality, we can easily get the inequality:

$$\gamma^{(t)} \leq a^{t-1}\gamma^{(0)} + \frac{1-a^{t-1}}{1-a}\eta \leq a^{t-1}\gamma^{(0)} + I\eta,$$

where $a = 1-1/I$ and $\gamma^{(0)}$ is the initial error. As t tends to infinity, a^{t-1} tends to zero so asymptotically the influence of $\gamma^{(0)}$ diminishes exponentially. It implies that $I\eta$ is the upper bound of the expected error metric in the long run.

C. Proof of Theorem 3

If only we can show that the uniform selection rule $P(i) = 1/I$ is an equalizer decision rule. First note that

$$E_P\left(L(R^{(t)}, i)\right) = 1/I,$$

independent of the distribution of $R^{(t)}$ as long as the i is chosen independent of $R^{(t)}$. It implies that the such a decision rule is actually an equalizer for the game:

$$E_p\left(\max_{X^{(t-1)}} L(R^{(t)}, i)\right) = 1/I.$$

So the uniform rule is minimax.

References

[1] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An information-theoretic approach to traffic matrix estimation," in *ACM SIGCOMM*, 2003.

[2] K. Papagiannaki, N. Taft, and A. Lakhina, "A distributed approach to measure ip traffic matrices"," in *ACM Internet Measurement Conference*, October 2004.

[3] A. Gunnar, M. Johansson, and T. Telkamp, "Traffic matrix estimation on a large ip backbone - a study on real data," in *Proc. ACM IMC*, October 2004.

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

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

[6] S. Vaton and A. Gravey, "Network tomography : an iterative bayesian analysis," in *Proc. ITC 18*, August 2003.

[7] G. Liang and B. Yu, "Pseudo likelihood estimation in network tomography," in *IEEE Infocom 2003*, San Francisco, April 2003.

[8] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot, "Traffice matrices: Balancing measurement, modeling and inference," in *ACM Sigmetrics*, June 2005.

[9] A. Soule, A. Nucci, E. Leonardi, R. Cruz, and N. Taft, "How to identify and estimate the largest traffic matrix elements in a dynamic environment," in *ACM Sigmetrics*, June 2004.

[10] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale ip traffic matrices from link loads," in *ACM Sigmetrics*, San Diego, USA, June 2003.

[11] G. Liang and B. Yu, "Maximum pseudo-likelihood estimation in network tomography," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2043–2053, August 2003.

[12] G. Liang, B. Yu, and N. Taft, "Maximum entorpy models: convergence rates and application in dynamic system monitoring." in *International Symposium on Information Theory*, 2004.

[13] M. Hanke and P. Hansen, "Regularization methods for large-scale problems," *Surveys on Mathematics for industry*, vol. 3, pp. 253–315, 1993.

[14] G. Wahba, "Spline models for observational data," in *SIAM*, Philadelphia, 1990.

[15] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 623–656, July, Oct 1948.

[16] R. McDougall, "Entropy theory and ras are friends," Center for Global Trade Analysis, Department of Agricultural Economics, Purdue University, GTAP Working Papers 300, 1999.

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

[18] W. Deming and F. Stephan, "On a least square adjustment of a sampled frequency table when the expected marginal totals are known," *Ann. Math. Statist.*, no. 11, pp. 427–444, 1940.

[19] J. Darroch and D. Ratcliff, "Generalized iterative scaling for log-linear models," *The Annals of Mathematical Statistics*, vol. 43, no. 5, pp. 1470–1480, 1972.

[20] S. D. Pietra, V. D. Pietra, and J. Lafferty, "Induce features of random fields," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 380–393, 1997.

[21] M. Schneider and S. Zenios, "A comparative study of algorithms for matrix balancing," *Operational Research*, vol. 38, no. 3, pp. 439–455, 1990.

[22] P. Bickel and E. Levina, "Some theory for Fisher's linear discriminant function, 'naive bayes', and some alternatives when there are many more variables than observations," *Bernoulli*, vol. 10, no. 6, p. 9891010, 2004.

[23] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," in *ACM Sigmetrics*, June 2004.

[24] J. Rice, *Mathematical Statistics and Data Analysis*. Duxbury Press, 1988.

[25] A. Soule, K. Salamatian, and N. Taft, "Traffic matrix tracking using kalman filters," *Large Scale Network Inference Workshop at ACM Sigmetrics*, June 2005.

[26] 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.