On the Convergence of Statistical Techniques for Inferring Network Traffic Demands

Alberto Medina[1], Kave Salamatian[2], Nina Taft[3], Ibrahim Matta[1], Yolanda Tsang[4], Christophe Diot[3]
[1] Boston University, Computer Science, USA [2] University Paris 6, France
[3] Sprint Advanced Technology Labs, Burlingame, CA, USA [4] Rice University, ECE, USA

February 6, 2003
Technical Report BUCS-2003-003

ABSTRACT

Accurate knowledge of traffic demands in a communication network enables or enhances a variety of traffic engineering and network management tasks of paramount importance for operational networks. Directly measuring a complete set of these demands is prohibitively expensive because of the huge amounts of data that must be collected and the performance impact that such measurements would impose on the regular behavior of the network. As a consequence, we must rely on statistical techniques to produce estimates of actual traffic demands from partial information. The performance of such techniques is however limited due to their reliance on limited information and the high amount of computations they incur, which limits their convergence behavior. In this paper we study strategies to improve the convergence of a powerful statistical technique based on an Expectation-Maximization iterative algorithm. First we analyze modeling approaches to generating starting points. We call these starting points *informed priors* since they are obtained using actual network information such as packet traces and SNMP link counts. Second we provide a very fast variant of the EM algorithm which extends its computation range, increasing its accuracy and decreasing its dependence on the quality of the starting point. Finally, we study the convergence characteristics of our EM algorithm and compare it against a recently proposed Weighted Least Squares approach.

1. INTRODUCTION

A traffic matrix (TM) reflects the volume of traffic that flows between source and destination nodes in a network. The nodes can refer to a variety of network elements such as POPs, routers or even address prefixes [8]. A POP-to-POP traffic matrix X captures the amount of traffic exchanged between two Points-of-Presence (POPs), where X_{ij} represents the volume of traffic traveling from ingress POP i to egress POP j. The value of X_{ij} usually represents a bandwidth value averaged over some time interval, although other types of elements are also possible.

There are a number of traffic engineering tasks that could be greatly improved with the knowledge provided by traffic matrices. Capacity planning, routing protocol configuration, definition of load balancing policies and fail-over strategies are tasks that would benefit from having information on the size and locality of traffic exchanges. An important example is the setting of OSPF or IS-IS routing weights. With knowledge of the TM, an algorithm for setting weights will select a routing that achieves a significantly better load balancing than one with an incorrect idea of the TM.

Obtaining a traffic matrix can be basically approached in two ways. We may directly measure it or we can rely on partial information to infer it. Measurement approaches have not been fully explored because they involve overcoming challenging engineering obstacles related to the deployment of a measurement infrastructure, and the storing and processing of large amounts of information. Furthermore, the monetary cost may be high.

Instead, previous work on obtaining traffic matrices has relied on statistical inference techniques that use partial information to estimate the TM. The term *Network Tomography* [13] has been coined for this problem when the partial data come from repeated measurements of the traffic flowing along directed links in the network. Such data are usually obtained from the Simple Network Management Protocol (SNMP [3]), which allows measuring the total amount of incoming and outgoing bytes on a link typically over five-minute intervals. The idea behind inference approaches is to use these *link* statistics to infer the characteristics of *end-to-end flows*. End-to-end flows are defined within a single domain and are usually referred to as origin-destination (OD) pairs. In a POP-to-POP topology, the origin and destination nodes are POPs. In addition to inference methods, it is also possible to formulate the traffic matrix estimation problem as a constrained optimization problem and use techniques such as Linear Programming [5].

Medina et al. conducted a comparative study of existing TM inference techniques [9]. The evaluated statistical techniques [13, 11, 2] are found to outperform an LP-based technique, still statistical techniques are significantly restricted in their ability to converge to the right solution. This is because they rely on scarce actual network information and they require intensive computation to reach reasonably accurate estimates. These restrictions impose a substantial burden on the quality of the starting point that should be provided to guide the estimation process.

Our Contribution:

In this paper we investigate two directions toward providing more efficient and more accurate estimations of network traffic demands for operational networks. First, we introduce a very fast variant of the Expectation Maximization algorithm for the network tomography problem. The improvements made are aimed at reducing the computation requirements of the algorithm, enabling it to expand the iterative horizon in search of global optima as solutions to the inference problem. Second, we investigate alternative modeling approaches to provide *reasonable* starting points for inference techniques. We call these starting points *informed priors* because they are obtained from models that incorporate substantial network information. Specifically we study the use of commonly used models, choice models as introduced by Medina et al. [9], a simple gravity model as introduced by Zang et al. [15], and an alternative but simpler formulation to the choice models we call linear-choice models.

We found that many of these approaches to modeling starting points for statistically inferring network traffic demands behave similarly in the sense of producing starting points within the same error range. We observe that the convergence speed of our underlying EM algorithm significantly improves as compared to standard EM implementations and its convergence behavior is substantially more robust as long as the provided prior is *reasonable*. Finally, an EM approach was found to outperform other statistical approaches in [9]. In this paper we compare our EM algorithm against a recently proposed alternative approach that uses quadratic programming, or more specifically, a weighted least squares (WLSE) algorithm [15].

The rest of the paper is organized as follows. In Section 2 we give the formal problem statement for TM inference. In Section 3 we review the main statistical techniques that have been proposed for inferring network traffic demands. In Section 4 we present one of our main contributions in the form of a fast variant of the EM algorithm. Section 5 describes the collection of packet traces and SNMP data we use in this study. In Section 6 we discuss the methodology we followed for the performance evaluation of the studied techniques. In Section 7, we discuss different alternative approaches to modeling starting points. In Section 8, we present and discuss the results of the performance evaluation. Finally, Section 9 concludes the paper.

2. PROBLEM STATEMENT

The network traffic demands inference problem can be formulated as follows. Let m be the number of origin-destination (OD) pairs. In a network with n nodes, $m = n \times (n-1)$. Rather than representing the amount of data transmitted from node i to node j as X_{ij}, it is usually more convenient to represent the list of OD pairs in vector form. We thus order the pairs and let X_j be the amount of data transmitted by OD pair j [1]. Let $Y = (y_1, ...y_L)$ be the vector of link counts where y_i gives the link count for link i, and L denotes the total number of links in the network. The vectors X and Y are related through an L by m routing matrix R. R is a $\{0, 1\}$ matrix where $r_{ij} = 1$ if link i belongs to the path associated with OD pair j, and $r_{ij} = 0$ otherwise. The OD flows are thus related to the link counts according to the following linear relation:

$$Y = RX \quad (1)$$

In IP networks, the routing matrix R can be obtained by gathering topological information, as well as OSPF or IS-IS link weights. Using this information we can compute the shortest-paths between all OD pairs. For simplicity, we assume the existence of a fixed single-path routing, that is, there is a single shortest path selected by all traffic flowing between any pair of end nodes in the network.[2] Link counts in Y are obtained from SNMP data. The problem is thus to compute X, that is, to find a set of OD flows that would reproduce the observed link counts as closely as possible. Notice that this formulation assumes that the components of Y come from a single measurement interval. A series of consecutive measurements of SNMP link counts, Y_i^k, can be considered, each one denoting the average load on link i in measurement period k. With such repeated measurements, the demands are as well modified to X_j^k, denoting the traffic demand for OD pair j in measurement interval k. The OD traffic demands and link counts are still related through R, as $Y^k = RX^k$.

The problem described by Equation (1) is highly under-determined because in almost any network, the number of OD pairs is much higher than the number of links in the network, that is, $L \ll m$. This means that there are an infinite number of feasible solutions for X. One approach to search through this large space is to use statistical inference methods to find the "most likely" solution given the observed partial network information.

We have additional information that may be incorporated into the problem statement. Specifically, the total amount of bytes leaving a node i corresponds to the sum of the SNMP link counts for all outgoing links from node i. Similarly, the total amount of bytes incoming into a node j corresponds to the sum of the SNMP counts over all links coming into node j. The amount of traffic traveling from i to j can be computed from the total amount of traffic exiting node i (denoted by O_i) multiplied by the fraction of this traffic headed toward node j. Let α_{ij} denote the fraction of the total traffic from node i traveling toward node j. With this notation, we can write X_{ij} as

$$X_{ij} = O_i \alpha_{ij} \quad (2)$$

The set of proportions, α_{ij}, $\forall j$ corresponds to what is often called the *fanout intensities* of node i. An alternative angle to look at the traffic estimation problem is to focus on the estimation of the *fanout intensities* of nodes in the network [9]. In other words, the problem now becomes that of estimating the proportionality factors, α_{ij}.

It is important to notice that if the fanout intensities can be accurately estimated, then the traffic matrix itself would consequently be accurately estimated from Equations (2), and there would not be any need for further inference or estimation procedures. The more likely scenario would be one in which the fanout intensities are estimated with certain errors. Nevertheless, these sub-optimal estimated fanout intensities would be very useful to provide good starting points for the estimation procedures of statistical techniques.

3. STATISTICAL INFERENCE TECHNIQUES

Statistical approaches for estimating network traffic demands have the general structure depicted in Figure 1. There are three main inputs. First, each statistical approach makes an assumption about the elements (entries) of the TM. Such an assumption is not actually an input but the foundation of the estimation engine used later is fundamentally influenced by such assumption. Second, statistical methods usually require some starting point (prior) information, aimed at conveying some *clues* about the traffic matrix being estimated. Such a starting point may correspond to an outdated version of the TM or be the output of some other mechanism aimed at obtaining a prior (as we shall see in Section 7). Finally, additional information is provided such as the link counts (the vector Y) and routing information used to construct the routing matrix for the studied network topology. The estimation part includes computing the parameters of the assumed probability distribution—parameters that maximize the likelihood of observing the measured link counts on the given routing matrix. Once these parameters are obtained, the output traffic matrix is populated with the average for each entry. A final step called *proportional fitting* adjusts the estimated average values to satisfy as close as possible the constraints imposed by the link counts.

Few statistical inference approaches have been proposed to date [13, 11, 2, 15]. The basic idea behind the first three approaches is to first define a probabilistic model describing the bandwidth of

[1] In this subsection we use X defined this way as a vector for mathematical convenience. In the rest of the paper we let X be indexed by $_{ij}$ to identify the origin and destination indices.

[2] It is straightforward to relax this assumption to deal with other routing schemes, e.g. multi-path (ECMP) routing.

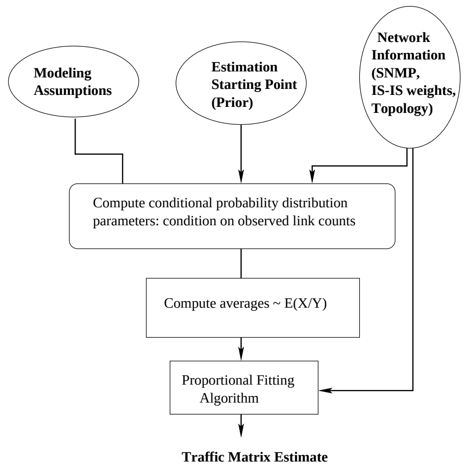

Figure 1: General diagram of statistical techniques

OD pair flows. First, estimation techniques, such as maximum likelihood estimators, are used to estimate all model parameters. Then, the traffic matrix is populated with a conditional expectation capturing the mean bandwidth of the flow between two end nodes, conditioned on the observed SNMP link counts. For example, Vardi [13], and Tebaldi and West [11] define a probabilistic model that assumes origin-destination flows follow a Poisson distribution. Cao et al. [2] assume instead that origin-destination flows follow a Gaussian distribution. To estimate the model parameters, Tebaldi and West [11] use a Bayesian approach, combining Gibbs sampling with Monte Carlo simulations, while Cao et al. [2] use an Expectation Maximization (EM) algorithm to compute maximum likelihood estimates.

The focus of this paper is on an EM approach and its convergence properties. For comparative purposes we contrast the performance of our EM algorithm to that of a quadratic programming approach recently proposed in the literature [15]. One of the main contributions of our work is the derivation of a very fast variant of the standard EM algorithm, which is discussed in Section 4. We refer to the approach in [15] as the Weighted Least Squares Estimation (WLSE) method. This method was proposed as part of an estimation method coined by the authors as *tomogravity*. Tomogravity consists of obtaining a starting point using a gravity model (see Section 7.2), and then reducing the error in the starting point by using *quadratic programming*. The error-reduction step seeks to find a solution that minimizes the distance to the starting point while at the same time satisfying the restrictions imposed by the system $RX = Y$.

4. A FAST EM ALGORITHM

We use the framework established by Cao et al. [2]. Let $Y_t = (Y_t^1, \ldots, Y_t^L)$ be a vector of observed traffic counts at time t on L links, and let $\lambda = (\lambda_1, ...\lambda_m)$ be the vector of mean rates, where m is the number of OD pairs. It is common in these kinds of problem to assume some kind of relationship between the mean and the variance. Without such an assumption the variances, and possibly covariances, would also need to be estimated. This may drive the number of variables to estimate very high. We therefore assume that the variance and the mean of traffic rates can be related by $\sigma_i^2 = \phi\lambda_i^c$. The value of c can be fixed to a known value or estimated over empirical data.

The parameters to be estimated in this framework are $\theta = (\lambda, \phi)$. We wish to estimate θ by a maximum likelihood criteria. The log-likelihood of the observed traffic values $(Y_1, ..., Y_T)$ can be calculated as:

$$\begin{aligned} l(\theta|Y_1,...,Y_T) &= -\frac{T}{2}\log|R\Sigma R^{'}| \\ &\quad -\frac{1}{2}\sum_{t=1}^{T}(Y_t - R\lambda)^{'}(R\Sigma R^{'})^{-1}(Y_t - R\lambda) \end{aligned} \quad (3)$$

where Σ is the covariance matrix.

The maximum likelihood estimate $\hat{\theta}$ is defined as:

$$\hat{\theta} = \arg\max_{\theta} l(\theta|Y_1,...,Y_T)$$

As Σ is related to λ there is no analytic solution to the above optimization problem. Even if it remains possible to do a brute force resolution, however as the inversion of $(R\Sigma R^{'})$ is inside the optimization, it might be hazardous and difficult. We therefore choose to use an EM approach to do the optimization. The EM method replaces the previous optimization problem by an iterative procedure where at each step a conditional expectation function Q is optimized.

In the problem under study the complete data log-likelihood can be obtained from:

$$l(\theta|X_1,...,X_T) = -\frac{T}{2}\log|\Sigma| - \frac{1}{2}\sum_{t=1}^{T}(X_t-\lambda)^{'}\Sigma^{-1}(X_t-\lambda)$$

The EM conditional expectation function is defined as follows:

$$\begin{aligned} Q(\theta,\theta^k) &= E(l(\theta|X)|Y,\theta^k) \\ &= -\frac{T}{2}(\log|\Sigma| + \mathrm{Tr}(\Sigma^{-1}W^{(k)})) \\ &\quad -\frac{1}{2}\sum_{t=1}^{T}(u_t^{(k)}-\lambda)'\Sigma^{-1}(u_t^{(k)}-\lambda) \end{aligned} \quad (4)$$

where

$$\begin{aligned} u_t^{(k)} &= \lambda^{(k)} + \Sigma^{(k)}R'(R\Sigma^{(k)}R')^{-1}(Y_t - R\lambda^{(k)}) \\ W^{(k)} &= \Sigma^{(k)} - \Sigma^{(k)}R'(R\Sigma^{(k)}R')^{-1}R\Sigma^{(k)} \end{aligned}$$

where the terms $u_t^{(k)}$ and $W^{(k)}$ are the conditional mean and variance of X given both Y and the current estimate θ^k. Tr(.) denotes the trace of a matrix, i.e. the sum of the diagonal elements.

Each iteration of the EM method consists of two steps: one expectation step (usually called the E-step) and one maximization step (called the M-step). The E-step consists of calculating the conditional expectation function $Q(\theta, \theta^k)$ as per Equation (4), by using the k^{th} estimate of θ, namely θ^k. In the M-step, the new value $\theta^{(k+1)}$ is obtained by maximizing the conditional expectation function:

$$\theta^{(k+1)} = \arg\max_{\theta} Q(\theta, \theta^k)$$

It can be shown that θ^k converges to a minima of the likelihood function.

4.1 Implementation of EM Algorithm

The optimization problem involved in the M-step can be solved by finding the value that drives the gradient of the function Q to zero, that is, $\frac{\delta Q}{\delta\theta}|_{\theta=\theta^{(k+1)}} = 0$. It was shown in [2] that this is equivalent to solving the following nonlinear equation:

$$\begin{aligned} 0 &= c\phi\lambda_i^c + (2-c)\lambda_i^2 - 2(1-c)\lambda_i b_i^{(k)} - ca_i^{(k)},\ i = 1,\cdots,m \\ 0 &= \Sigma_{i=1}^{m}\lambda^{-c+1}(\lambda_i - b_i^{(k)}) \end{aligned} \quad (5)$$

where

$$b_i^{(k)} = \frac{1}{T}\sum_{t=1}^{T} m_{t,i}^{(k)}$$

$$a_i^{(k)} = w_{ii}^{(k)} + \frac{1}{T}\sum_{t=1}^{T} (m_{t,i}^{(k)})^2$$

The authors in [2] replace the classical EM method by a modified EM method where at each step $\theta^{(k+1)}$ is updated using a Newton-Raphson or a second order method. The convergence of this modified EM method is reported to be slow, and singularity problems appear frequently when inverting the $(R\Sigma^{(k)}R')^{-1}$ term at each iteration. This is mandatory for calculating $u_t^{(k)}$ as well as $W^{(k)}$. Because the number of iterations of the EM algorithm could be very large in this approach, the problematic matrix inversion step would be carried out several times, which in turn leads to large complexity.

In our implementation of the EM method we modify the algorithm to obtain a fast version of the EM algorithm for this problem. A fast version of an EM algorithm is important for the scalability of TM estimation. As TM estimation gets applied to networks with larger number of nodes (such as router-to-router TMs), scalable and fast EM algorithms become essential. We have included three improvements that greatly speed up the run time of such algorithms:

- We introduce additional linear constraints into the system. This often results in increasing the rank of the R matrix, which has two advantages: it reduces the search space and it makes $(R\Sigma^{(k)}R')^{-1}$ inversion more stable.
- We convert the routing matrix R to one which is a linear transform of the original matrix and looks as close to an identity matrix as possible. We do so by transforming the matrix to a reduced echelon form. Having an R matrix in this form enables the optimization procedure to run much more quickly.
- As suggested in [2], we transform the optimization problem involved in the M-step of the EM method to solving a non-linear equation. However we solve this equation using sophisticated numerical techniques suited to large scale problems and we follow closely the EM algorithm, *i.e.* we set exactly $\theta^{(k+1)} = \arg\max_\theta Q(\theta, \theta^k)$.

The combination of these three ingredients greatly speed up the optimization steps which now take less than a couple of minutes to run on a standard laptop computer. Next, we describe in more detail the three steps involved in our implementation.

Additional Constraints

One of the measures of interest in traffic engineering is the amount of traffic flowing into (from) a POP from (toward) the backbone, respectively. These values correspond to the sum of columns and the sum of rows of the traffic matrix (not to be confused with the routing matrix). We add these values as additional constraints into our linear system.

More specifically, the $\sum_j X_{ij}$ for $j = 1, \cdots, n$ gives the total amount of traffic node X_i sends into the backbone, and corresponds to the sum of row i in the traffix matrix. This amount should be equal to the sum of the SNMP counts on all links exiting PoP node i. For a network with n PoPs or nodes, this adds an extra n constraints. Similarly, the column constraints are obtained from $\sum_i X_{ij}$ for $i = 1, \cdots, n$ that denotes all the traffic received by PoP j from the backbone. This includes the traffic from all OD pairs that terminate at PoP j. The value of this sum is computed from the sum of the SNMP link counts on all links entering PoP j from the backbone. The row sum and column sum constraints each add one block of equations in the system. These constraints strongly correlate the variables and make the spectral structure of the R matrix more stable, which in turn leads to more stability in the $(R\Sigma^{(k)}R')^{-1}$ inversion.

The sample routing matrix we consider in most of our examples has an initial rank of 40. After adding additional 28 constraints (for a 14-POP network), the rank increased to 46. This indicates that while many of these constraints are redundant with existing information, some new independent equations can be found as well. Increasing the rank of the R matrix is important as it makes the system of equations less underconstrained. There may be other techniques for increasing the rank of the R matrix but those would involve taking additional or different kinds of measurements. Such a line of thought is worthy of research, but is not our goal here. We sought merely to add as much information as possible given the assumed set of measurements.

Echelon Forms

The goal here is to transform the extended routing matrix R into a format more suitable for the optimization step. For this purpose we rewrite the R matrix in a reduced echelon form. Computing the reduced echelon form is merely taking a linear transform of the R matrix and thus does not change the solution sought.

There are two reasons to do this. The result of this step may yield some rows in which all elements are zero except for one element that is a one. The corresponding column in which this 'one' is located identifies an OD pair that in fact is explicitly known and does not need to be estimated. This OD pair can be removed from the estimation process and we thus reduce the dimension of the problem and the number of parameters that need to be estimated.

The second reason to do this step has to do with making the optimization run more quickly. Feeding an EM algorithm a matrix that has large component of it that resembles an identity matrix is a numerical advantage, as it will lead to a more sparse matrix and less error propagation.

EM steps

This last improvement provides a good deal of the speedup obtained in our method. In place of obtaining $\theta^{(k+1)}$ as suggested in [2], by a Newton-Raphson or second order method, we assign $\theta^{(k+1)}$ such that $\theta^{(k+1)} = \arg\max_\theta Q(\theta, \theta^k)$. This optimization problem is carried out by solving a set of nonlinear equations using a procedure based on least squares estimation that uses a trust region method and an interior-reflective Newton method. This was implemented using the optimization toolbox of Matlab [7]. With this approach we follow precisely the EM method whereas the method proposed by [2] is a modified approach.

Generally we found that our EM method converges in about 10 steps, because during the optimization of $Q(\theta, \theta^k)$, the values of $u_t^{(k)}$ and $W^{(k)}$ do not change, thus we only need to carry out the costly matrix inversion operation once in every step, whereas the modified approach proposed in [2] needs to do the matrix inversion hundreds (and sometimes thousands) of times.

5. MEASUREMENTS USED

The work presented in this paper was done in the context of a Tier-1, continental-US, backbone network. We use packet traces from several monitored POPs, as well as SNMP data collected for all backbone links. We use information computed from the packet traces, together with SNMP data, for calibrating and validating the studied starting point models, and for testing the performance of our EM algorithm.

5.1 Packet Traces

We used two sets of full packet traces, which were collected on September 5, 2001 for a time interval of 12 hours, and on November 21, 2002 for an interval of 10 hours. These two sets contain packet traces for 3 POPs and 2 POPs, respectively. The collection of these packet traces was performed by monitoring sets of links at each monitored POP (about 10 links per POP) in the studied backbone network. Specifically, we monitored aggregated access links (customers), which connect access routers to core routers, peering links and inter-POP backbone links. The collected packet traces provide us with *measured* estimates of actual rows of the corresponding POP-to-POP traffic matrix.

In order to compute actual rows of a TM from packet traces we apply a mapping procedure that takes as input the destination address of an incoming packet and outputs the egress POP through which the packet will leave the network. The implementation of such a mapping mainly uses BGP routing information and, for some cases in which BGP information is not enough to establish the mapping, traceroutes [12] are used. Using our mapping procedure we are able to map more than 99% of the monitored packets. We can then compute the fraction of all packets that were sent from a monitored (ingress) POP to every egress POP, i.e., the fanout intensities α_{ij}.

5.2 SNMP Data

The Simple Network Management Protocol (SNMP) provides per-link information regarding the number of bytes flowing through each link in the network over some interval of time (e.g., 5 minutes). This information is systematically collected from all links in the backbone network and we use it at different aggregation levels for computing POP attributes as well as for evaluating performance improvements gained by the combinations of starting point and estimation techniques we have studied. Specifically, from SNMP data we draw information about aggregated customer, peering, inter-POP, and intra-POP link utilization levels in the network. For each of these link types we determine the average used capacity over a certain interval of time. SNMP provides per-link byte-count information at a minimum granularity of 5 minutes.

Note that the SNMP data used to compute the link-utilization statistics were collected during the same period as the packet traces, that is, 12 hours on September 5, 2001, and 10 hours on November 21, 2002.

5.3 Time Scales

The characteristics, availability and applications of measured or estimated network traffic demands depend to a large extent on the time granularity used to collect the data. On one hand, the collected packet traces in the studied backbone network are gathered at the time granularity of packet arrivals. For this work, we pre-process the packet traces to compute a basic aggregation level capturing the number of packets and bytes per second arriving to the measured links. Such minimal level of aggregation can be further increased as needed. On the other hand, the SNMP link utilization data is collected at a time granularity of 5 minutes. As with packet traces, higher levels of aggregation, always in multiples of 5 minutes, are obtained as needed. For example, if we want to estimate a TM over a one-hour time period, the SNMP link counts would be aggregated by summarizing 12 5–minute measurements with an average value. In our study, we are interested in aggregation levels of at least one hour since we are targeting traffic engineering and network management tasks for which changes in POP-to-POP traffic exchanges over finer timecales are not of interest.

6. EVALUATION METHODOLOGY

One of the challenges that must be tackled when investigating inference mechanisms to estimate network traffic demands is the issue of how to validate the results. Ideally, we would have complete accurately measured network traffic demands to compare the results of the inference process against them. However, if we had an effective and efficient mechanism to obtain such accurate measurements we would not need to rely on statistical inference! Alternatively, we would like to obtain substantial information about network traffic demands using mechanisms such as Netflow or BGP Policy Accounting. Doing so, however, is difficult since these mechanisms may impose a significant burden on routers and consequently may degrade the performance of the network. In this section we describe the approach we adopted for validating our EM algorithm and for assessing its convergence behavior.

6.1 Empirical Model for Synthetic TMs

In general, previous studies and comparative evaluations have relied on limited actual network information and on synthetically generated traffic matrices based on seemingly strong assumptions regarding the underlying distributions of the actual traffic exchanges between origin-destination (OD) pairs [13, 2, 11, 9]. For example, a common approach has been to assume that OD demands are distributed according to a Gaussian or Poisson distribution. Alternatively, more skewed distributions (e.g. Bimodal) have been proposed for testing purposes as well. Although making such assumptions may be useful in terms of agreeing with the intrinsic assumptions made by the statistical technique used, they may not be representative of the actual characteristics of OD traffic exchanges [9].

The validation approach we use in this paper makes use of what we call an *empirical model* for synthetic traffic matrix generation. This very simple empirical model consists of two steps which use the measurement data described in Section 5. Specifically, we use packet traces collected at a Tier-1 backbone network to determine an empirical distribution of the POP-to-POP fanouts, and use SNMP utilization information to establish a hierarchy of *importance* among egress POPs. The procedure is as follows:

(1) Determine empirical distribution of fanouts: As described in Section 5, we have access, on different dates, to information regarding actual POP-to-POP traffic exchanges for up to three POPs. Despite the very large amount of data collected for each of the measured POPs, we are capturing only a fraction of the total traffic flowing through the POPs. However, we believe that by carefully choosing the POPs and links from which packet traces are collected, the traffic demand information gathered in the process would capture an important component of the behavior of traffic exchanges. Using an empirically derived distribution of POP fanouts for, say, three POPs, we generate random fanouts. Figure 2 shows an example empirical complementary cumulative distribution function of fanouts and the associated fit with a simple single-exponential function.

(2) Define egress POP ranking: Building on the premise that POPs are engineered in correlation with the amounts of data they would need to handle, we establish a POP ranking based on utilization information about the POPs as given by SNMP data. Specifically, for each egress POP, we rank it according to its individual attributes, such as utilization levels for incoming and outgoing customer, peering and inter-POP links. Then an overall ranking is de-

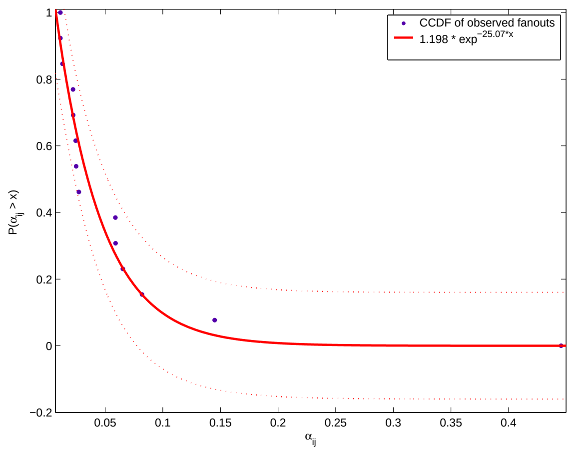

Figure 2: Fit of empirical CCDF for fanouts distribution

termined by summing the individual rank values for each egress POP. If $Rank_k^j$ is the rank of egress POP j with respect to attribute A_k, then we compute the overall rank of POP j as $\sum_k Rank_k^j$.

(3) Match random fanouts to ranked egress POPs: the last step consists of sorting the random fanouts obtained in step one for each ingress POP, and assigning them to the egress POPs in order according to their rank established in step two.

Although this empirical model is very simple, it is aimed at providing synthetic target traffic matrices that are in some sense more realistic and can provide more meaningful evaluation test cases.

6.2 Synthetic–data Experiments

Synthetic data is very useful to evaluate the performance of traffic matrix estimation techniques since it enables us to assess their behavior with respect to whole matrices rather than partially measured TMs. By performing synthetic–data experiments we can better assess the errors yielded by the evaluated techniques, determine the distribution of errors among the estimated OD traffic demands, etc.

In this step of the evaluation process, we use the empirical model described in Section 6.1 to generate a target synthetic traffic matrix. As depicted in Figure 3, we route this target matrix onto the topology of the studied network to obtain a set of (synthetic) link counts equivalent to the set of link counts that would be provided by SNMP data. Then, we generate a starting point for the estimation procedure according to any of the models described in Section 7. We pass the link counts and the starting point to the chosen estimation technique to obtain an estimated TM. Finally, we compare the output of the estimation to the target TM to assess the error incurred by the estimation procedure.

Both the synthetic target TM and the chosen starting point are generated consistently using packet traces and SNMP data corresponding to the same period of time. Once the synthetic fanouts ($\hat{\alpha}_{ij}$) have been defined (cf. Section 6.1), the synthetic target TM is populated using actual SNMP data to determine the total amount of bytes leaving POP i via inter-POP links as follows:

$$X_{ij} = O_i \times \hat{\alpha}_{ij} \tag{6}$$

6.3 Real–data Experiments

The next step is to evaluate estimated network traffic demands with respect to their *goodness-of-fit* or closeness to measured traffic demands. The approach is similar to the one described in Section

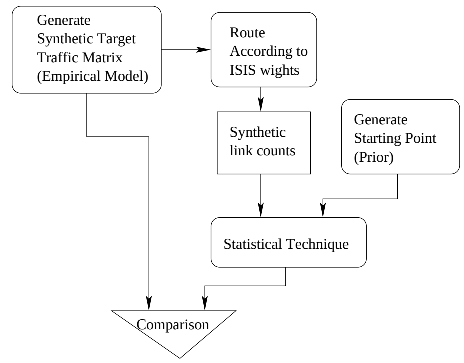

Figure 3: Performance evaluation for synthetic cases

6.2 with two differences. First, we do not have a full target traffic matrix which would be used as before to generate a set of consistent link counts. Second, after the estimation procedure finishes, the comparison is not done against a full synthetic TM. Instead, we feed the given estimation technique with a set of actual SNMP link counts and a starting point generated in the same way as before. We then take the output estimated TM and compare the rows that correspond to the actual measured rows to assess the goodness-of-fit of the estimation.

As an example and following the diagram in Figure 4, suppose we have measured the third row of the actual traffic matrix for a given date, say November 21, 2002. Then, we will extract from the SNMP data repository, link utilization information for the same time at, say, 1-hour aggregation intervals. Then we generate a starting point according to, say, a choice model (cf. Section 7), and feed these into the EM algorithm. We then take the third row of the estimated TM and compare it against the measured row we have from the beginning.

For all experiments, the starting points are calibrated and populated with data (SNMP and packet traces) corresponding to the input data fed into the estimation technique used.

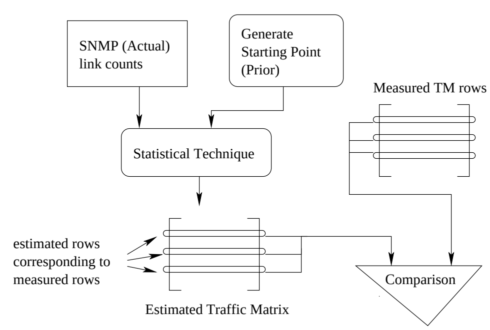

Figure 4: Performance evaluation for real-data test cases

To quantitatively compare the results, we plot entries of the estimated traffic matrix versus the target traffic matrix. The closer such a plot follows a linear trend the better is the mean quality of the estimated traffic matrix. Furthermore, we need to evaluate the dispersion of the estimation points around the mean. We compute this dispersion using the well-known Pearson's coefficient R [6]. The closer R is to one the better the estimation is.

7. MODELING REASONABLE STARTING POINTS

Although in general we may generate starting points arbitrarily or according to any *standard* distribution (e.g. Gaussian, Poisson, etc.), the convergence behavior of statistical techniques may be significantly influenced by the characteristics of the provided starting point [9]. In this section we describe different approaches to the modeling and population of reasonable starting points to be provided as input to statistical inference techniques for the traffic matrix estimation problem.

7.1 Mlogit and Linear-choice Models

Medina el al. [9] introduced an approach to modeling the fanouts of nodes using a *choice model framework* derived from Economic Consumer Theory. In this approach, the engineering characteristics of nodes in the network determine the likelihood that a byte will be transferred from node i to node j. Some degree of uncertainty in the process is also allowed by incorporating a random component into the choice models. More specifically [9], the utility U_j^i that a given ingress POP i gains from choosing to send a packet to POP j, is the sum of a deterministic component, V_j^i, and a random component, ϵ_j^i. Since it includes a random component modeling uncertainty, the utility function becomes a random variable. Therefore, the probability that POP i selects POP j from a set of egress POPs, representing the fanout intensities α_{ij}, equals the probability that the random variable U_j^i has the largest value among the utilities of all alternatives.

In general, given K attributes for each POP and let $f(A_k^i)$ $(g(A_k^j))$ denote a function of the k^{th} attribute of ingress POP i (egress POP j), V_j^i is given by:

$$V_j^i = \sum_{k=1}^{K} \beta_k f(A_k^i) + \sum_{k=1}^{K} \beta_{K+k} g(A_k^j) + \gamma_j \qquad (7)$$

where β_k defines the relative importance of attribute k with respect to the others, and γ_j is a scaling term.

Many different choice models can be defined based upon how many and which combination of attributes are included in the deterministic component. Assuming Gaussian random uncertainty, the so-called *multinomial logit* or *mlogit* model is derived in which the probability of POP i choosing a given egress POP j is given by [9]:

$$\alpha_{ij} = \frac{e^{V_j^i}}{\sum_{k \in C} e^{V_k^i}} \qquad (8)$$

where C is the set of egress POPs. Therefore, the traffic between a pair of POPs can be modeled by:

$$X_{ij} = O_i \alpha_{ij} \qquad (9)$$

where O_i represents the total outgoing bytes sent into the network by POP i. Intuitively, the mlogit function captures behavior in which a few traffic exchanges are large and dominate the overall characteristics of the traffic matrix, and in which there can be great differences between small and large traffic exchanges.

In this paper,we also consider a variant of choice models we call *Linear Choice models*, in which the form of the mlogit function is simplified by eliminating the exponential function at both the numerator and denominator of Equations (8) as follows:

$$\alpha_{ij} = \frac{V_j^i}{\sum_{k \in C} V_k^i} \qquad (10)$$

For the linear-choice models we set the weights of the V_j^i function to 1, yielding α_{ij} values that are linearly correlated with the attributes of the POPs.

7.2 Gravity Models

Gravity models are trip distribution models that have been widely used in transportation applications for estimating traffic demands between urban areas [4, 1, 14, 10]. Basically, a gravity model says that the trip interchange between zones in an urban area is directly proportional to the relative *attraction* of each of the zones and inversely proportional to some function of the *separation* between zones. In the context of the traffic estimation problem, we want to relate the amount of data exchanged between two nodes to the attraction, the ability of attracting data sent by other nodes, and some *friction* factor that influences how much data actually flows between the two nodes.

A general formulation of a gravity model may be given by the following equation:

$$X_{ij} = \frac{f(R_i, A_j)}{g_{ij}} \qquad (11)$$

where $f(.)$ is a non-decreasing function, X_{ij} is the traffic volume from i to j, R_i is a parameter representing *repulsive* factors which are associated with "leaving" i, A_j is a parameter representing *attractive* factors related to "going" to j, and g_{ij} represents the *friction* factor between i and j.

Since X_{ij} is a fraction of the total amount of traffic coming out of POP i, a simple gravity model formulation is given by rewriting the general Equation (11) as $X_{ij} = O_i \alpha_{ij}$. Note that this formulation is identical to the choice model formulation, leaving the fanout intensity factor, α_{ij} as a variable to be defined. In this model O_i is the repulsion factor, and it reflects the amount of traffic POP i dumps into the network.

In [15], the authors propose two simple and elegant gravity models for generating starting points for traffic matrix estimation. Their first model is called a "simple gravity model" while their second model is called a "generalized gravity model." In this paper we consider the simple gravity model for our comparative purposes. In this model, the friction factors in Equation (11) are assumed to be constant. Despite of such assumption being the simplest form for the friction factors, the formulated model does a good job at producing reasonable starting points to be input to a statistical approach. At the POP-to-POP level, the main idea is that the traffic exchanges between POPs in the network should be proportional to the volumes of traffic entering and exiting the end nodes in any OD pair. In a nutshell, the gravity model at the POP level is given by:

$$X_{ij} = O_i \frac{T_j^{out}}{\sum_k T_k^{out}} \qquad (12)$$

where O_i is defined as above, and T_j^{out} is the total amount of bytes leaving the network through POP j. Note that this gravity-based formulation is similar to the linear-choice formulation.

Approaching the generation of priors by using choice or gravity models has the goal of avoiding making statistical assumptions that may not be representative of the actual characteristics of actual network traffic demands. Models like these enable us to capture correlations between traffic characteristics and the properties that actually characterize the underlying network.

7.3 Common Models

A common approach to the generation of starting points for the estimation procedure has been to assume any underlying *standard* distribution for the elements of the traffic matrix and then synthetically populate starting points by generating random values according to the chosen distribution. For example, the technique proposed in the pioneering work of Vardi [13] assumes a Poisson distribution for the underlying traffic matrix. Therefore we may generate starting points for such a technique by populating synthetic traffic matrices according to a Poisson distribution. The EM approach proposed in [2] is developed based on the assumption that elements of the underlying traffic matrix are distributed according to a Gaussian distribution. In [2], a simple mechanism for generating starting points is also proposed. That mechanism generates constant starting points where the constant value of each entry in the TM is a weighted sum of average link utilization levels where the weights are set according to the number of OD pairs traversing each link on the OD-pair path. We experimented as well with such constant starting points.

We included in our experimental framework more *extreme* distributions such as multi-modal and skewed distributions. Investigating these distributions is important since they should expose the behavior of the studied statistical techniques in the presence of "unreasonable" starting points. Note that by *reasonable starting point* we mean starting points that are not radically different from the actual distributional shape of the underlying traffic matrix we are seeking to estimate.

7.4 Calibration Mechanisms

A starting point model may need to be calibrated, for example, to assign weight values to its parameters and create specific instances. Once the calibration has been performed when needed, we must then populate traffic matrices to be used as starting points.

7.4.1 Calibrating choice models

Choice models need to be calibrated so as to specify the coefficients β_k in Equations (7). To that end, packet traces and SNMP data are used in the calibration process. Packet traces, aggregated at the POP level, enable us to compute individual TM rows for the ingress POPs at which the packet traces were collected. These measured TM rows are used as the equivalent of sample surveys of the decisions made at the ingress POP as to where to send the bytes it generates, and they are provided to the calibration procedure. From SNMP data we extract POP-to-POP information regarding the capacity and utilization information for incoming and outgoing customer and peering links, as well as for inter-POP links in the studied tier-1 backbone network. To discuss the use of this information we use the following notation. Let D_j denote the total amount of traffic received by egress POP j from the backbone, which is computed by summing the SNMP link counts of all inter-POP links entering POP j. Let O_i denote the total traffic leaving POP i, which is computed by summing the SNMP link counts of all inter-POP links exiting POP i. Let C_i^{in} (C_i^{out}) denote the used capacity for incoming (outgoing) customer links at POP i. Finally, let P_i^{in} (P_i^{out}) denote the used bandwidth of incoming (outgoing) peering links for POP i.

Intuitively, the six most useful attributes should be O_i, D_j, C_j^{out}, C_i^{in}, P_j^{out} and P_i^{in}, for ingress POP i and egress POP j. We want to include attributes in our choice-models that are as uncorrelated as possible, since otherwise we may have co-linearity problems. To assess the correlation among different POP attributes, we calculated the correlation coefficient between all pairs of attributes (see Table 1) [3]. Only the pairs (O_i, C_i^{in}), (D_j, C_j^{out}) and (C_i^{in}, P_i^{in}), have correlation coefficients higher than 0.65. This implies that a

	O_i	D_j	C_i^{in}	C_j^{out}	P_i^{in}	P_i^{out}
O_i	1.0000	0.5992	0.9217	0.6032	0.5587	0.2167
D_j	-	1.0000	0.4316	0.7961	0.0767	0.3341
C_i^{in}	-	-	1.0000	0.5261	0.8366	0.3182
C_j^{out}	-	-	-	1.0000	0.2730	0.5386
P_i^{in}	-	-	-	-	1.0000	0.3744
P_i^{out}	-	-	-	-	-	1.0000

Table 1: Correlation coefficient of POP attributes

model should not include both the members of these pairs. Note that the relatively high correlation level for these pairs is expected. In the first case, (O_i, C_i^{in}), it is intuitive that the volume of data on the incoming customer links at an ingress POP is correlated to the amount of traffic the POP dumps onto the inter-POP backbone links (assuming that most of the customer traffic wants to cross the backbone and not exit immediately at the same POP). Similarly for the pair (D_j, C_j^{out}), there must be a strong correlation between the amount of traffic entering an egress POP j from the backbone and exiting the POP on its customer links. The correlation between (C_i^{in}, P_i^{in}) is a bit more surprising. Perhaps this indicates that if an ingress POP is small (large) it will have similarly small (large) numbers of customer and peering links, respectively.

Table 2 describes, in terms of the included attributes, the three choice models we have included in the results of this paper. These models behave best with respect to yielding lowest errors and producing reasonable starting points for the TM estimation procedure. Model I uses only two POP attributes given by the total amount of bytes entering and exiting a POP. Model II uses instead the volume of traffic leaving the network at POP j via customer and peering links. Finally, Model III replaces the use of O_i by the total volume of data coming into the network at POP i via customer and peering links.

Model	**Attributes**
I	(O_i, D_j)
II	$(O_i, C_j^{out}, P_j^{out})$
III	$(C_i^{in}, P_i^{in}, C_j^{out}, P_j^{out})$

Table 2: Attributes included in each model

The actual calibration of the choice model requires the calculation of the coefficients β_k in Equations (7) so as to match the α_{ij} for the measured POPs i. This is done by curve-fitting to the mlogit function using a maximum likelihood estimation implemented in the Econometrics toolbox of Matlab [7]. Once the model is calibrated, we compute the remaining fanout values α_{ij} using (8), and the full prior TM is then populated using (9).

7.4.2 Linear-choice and Gravity model Calibration

The linear-choice and gravity models do not need to undergo a calibration procedure since they do not have coefficient values.[4] These models need to be populated by extracting from the SNMP archives the information they require. We can then generate the starting point using Equations (9) and (12), respectively.

7.5 Comparative Analysis

Figures 5, 6, and 7, show a comparison between three different starting points, generated according to the gravity model, mlogit-choice model and a skewed distribution model, and the target TM which we seek to estimate. The target TMs used throughout most

[3] Since this matrix is symmetric, we only include half the values for ease of readability.

[4] In these models, the coefficients of the POP attributes are all set to 1.

of our experimental scenarios were generated using the empirical model described in Section 6.1. As can be observed, the gravity and mlogit-choice models produce priors that are scattered around the values of the target TM. The gravity model starting point is slightly more variable but very similar to the choice-model case. Skewed starting points are generated such that, for a given ingress POP, most of the egress POPs would have a low fanout value while a few will have significantly larger fanout values. As can be observed in Figure 7, starting points like this are not reasonable in the sense that they are significantly different from what the target TM we seek to estimate looks like. We incorporate this type of priors to evaluate the convergence behavior of the studied techniques when provided with *unreasonable* starting points.

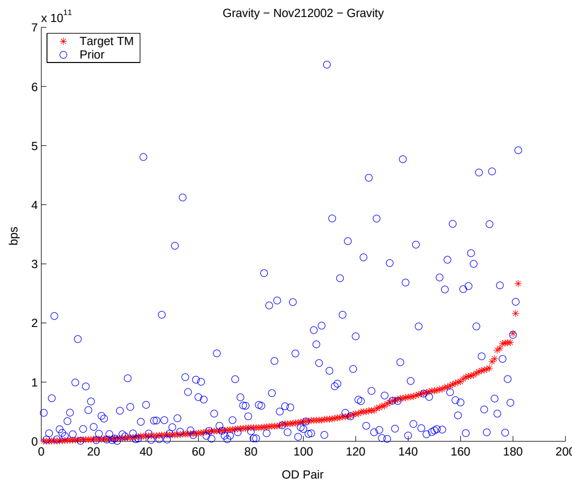

Figure 5: Gravity model prior vs. empirically modeled synthetic TM

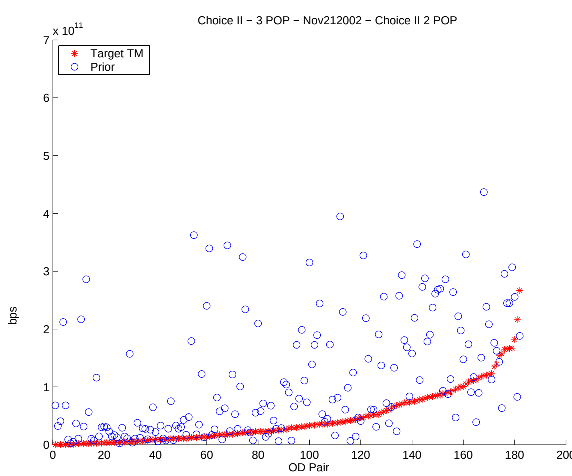

Figure 6: Choice model prior (II - calibrated using 3 measured rows) vs. empirically modeled synthetic TM

For each type of generated starting point, we compute the fanout values from the resulting starting TM. We then compare the observed fanouts, obtained form the actual measurements described in Section 5.1 against the corresponding starting-point fanouts. Figures 8 and 9 depict the results of this comparison. We observe that all studied ("reasonable") starting-point models produce similar results when compared to observed fanouts for two measured POPs. Therefore, a plausible conclusion to make is that any of these mechanisms may be used to produce starting points for satistical techniques and similar results should be expected. Furthermore, the more powerful the statistical technique is, the more resilient it will be, i.e. it will be more capable of recovering from even unreasonable starting points to produce reasonably accurate estimations. We further explore this issue in Section 8.

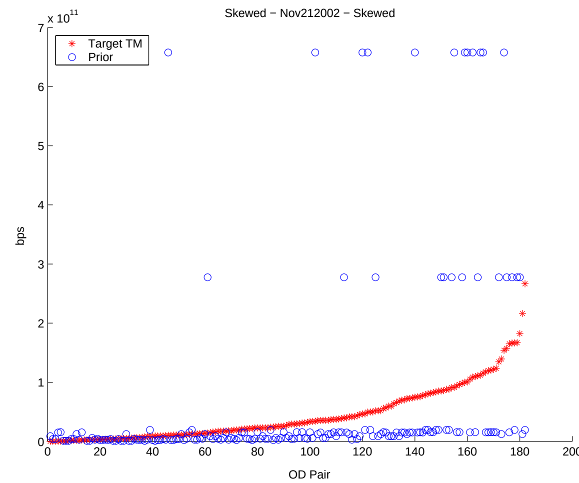

Figure 7: Skewed prior vs. empirically modeled synthetic TM

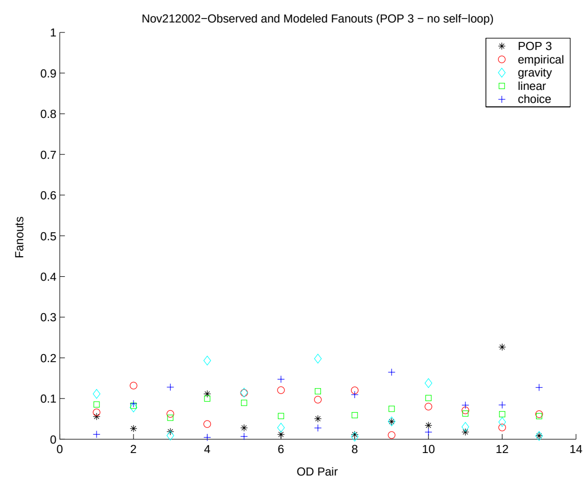

Figure 8: Observed vs. predicted fanouts for POP3 TM row

8. CONVERGENCE EVALUATION

In this section we discuss experimental results regarding the convergence behavior of our EM algorithm and the WLSE approach described in Section 3.

8.1 Constant-target Experiments

The *constant-target* experiments consist of attempting to estimate a constant traffic matrix while providing constant starting points with varying quality levels (i.e. errors). The target traffic matrix is chosen to have a component value of 200. At each run, we degrade the quality of the prior by 50% until a maximum error of 500%.

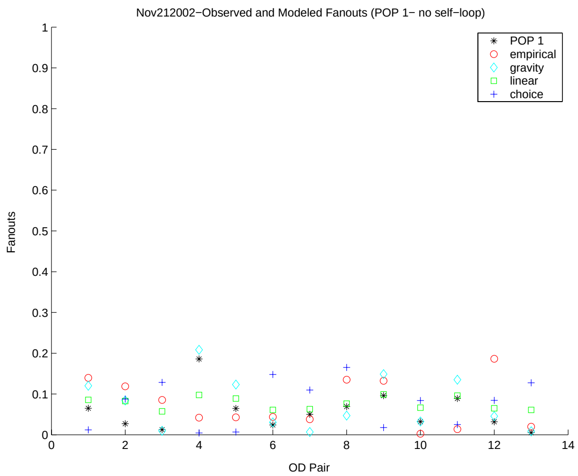

Figure 9: Observed vs. predicted fanouts for POP1 TM row

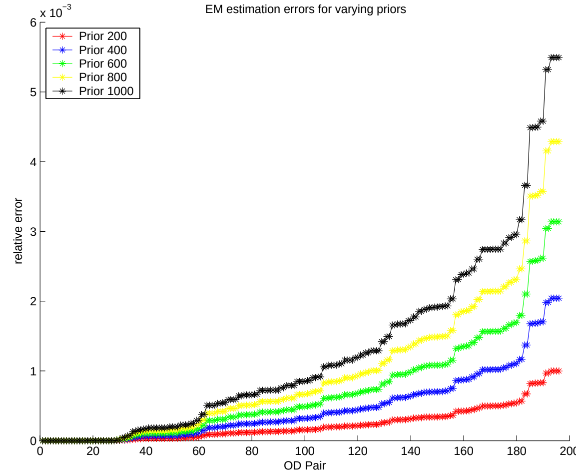

Figure 11: EM convergence with increasing prior errors

The formulation of the problem is given in terms of the under-determined system $RX = Y$. The WLSE method finds the solution that have the smallest norm or, equivalently, the solution that minimizes a quadratic criteria XX' subject to the constraints given by $RX = Y$. Therefore, the WLSE approach, as proposed in [15], finds the closest value to the provided starting point while satisfying the constraints. We expect if the starting point is far from the target, the closest value to it will also be far.

The design of these experiments was aimed at showing two things. First, we want to know how much error would be introduced by the application of a statistical technique if the provided starting point had 0% error. Second, we want to observe the response of the studied statistical techniques to increasingly degraded starting points. Figures 10 and 11 show the results for both the WLSE and EM algorithms.

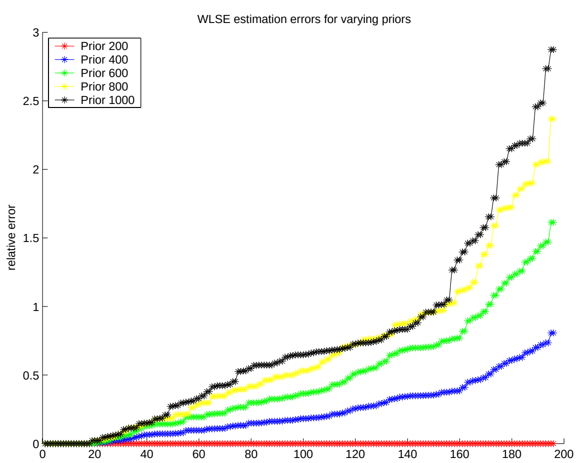

Figure 10: WLSE convergence with increasing prior errors

Note that the scales of the Y-axis in Figures 10 and 11 are not the same. The difference between the Y-ranges of the two figures is too large, preventing us from plotting them with the same Y-ranges. Otherwise, one of the plots would be rendered illegible. We observe that on one hand, for 0%–error starting points, the WLSE method does not add any error to the estimated TM, producing equivalently a 0%-error estimation. In contrast, the EM adds a little bit of noise of about 3% to the estimated TM. This small noise is the result of the EM method trying to fit its parameters to a Gaussian model while the target TM is constant. On the other hand, we observe how the WLSE quickly and substantially decreases the quality of the estimation as the error in the starting points increases. In contrast, the EM exhibits significant robustness to the degradation in the quality of the priors and keeps the errors in the estimation at very low levels.

8.2 Empirical Target Experiments

This set of experiments was designed according to the methodology described in Section 6.2. Specifically, two different target TMs were generated using archived SNMP for September 5^{th}, 2001, and November 21^{st}, 2002. Recall from Section 7, that the calibration procedure for the choice models makes use of available measured rows of the actual traffic matrix. Therefore, in the synthetic case, we calibrate the choice models varying the number of rows (one to six rows) from the synthetic target TM used for their calibration. We denote the number of rows used in the calibration by adding this number next to the number of the choice model used. As we will see, varying the number of rows used in the calibration did not affect significantly the estimation results for the choice model cases. Figures 12 and 13 depict plots showing, for both the WLSE and EM methods, the CDF of errors less than 50% for the estimation results obtained from various starting points other than choice-based (cf. Section 7). Similarly, Figures 14 and 15 depict another set of plots for the choice-modeled starting points. In Figure 12 we observe that for the EM method the slope of the CDF is similar for all starting points except for the skewed and bimodal cases where it is lower. The latter are the starting points we have labeled as *unreasonable* and we expect statistical techniques to yield higher errors when this type of starting points are used. In contrast, we observe in Figure 13 the slope of all curves is lower and the WLSE method behaves similarly for all cases—the performance is similar to that of the EM for the skewed and bimodal cases.

When choice models are used to generate the starting points, the performance of the EM exhibits little variability independently of the specific choice model used (Figure 14). For the WLSE method (Figure 15), the results vary slightly more but still the results are more *stable* than for the other starting point models.

The previous plots show only the lower end of the errors for both

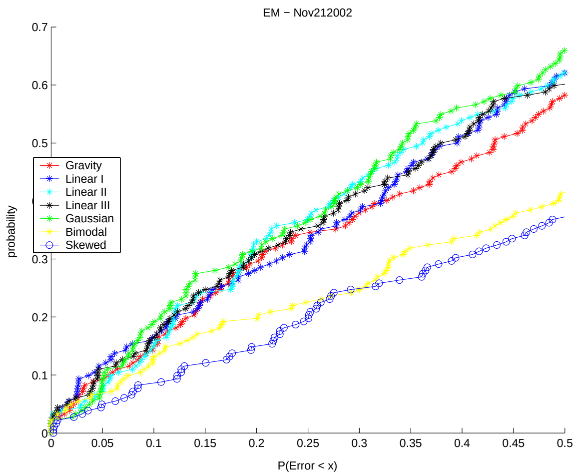

Figure 12: Convergence of EM method (prior set I)

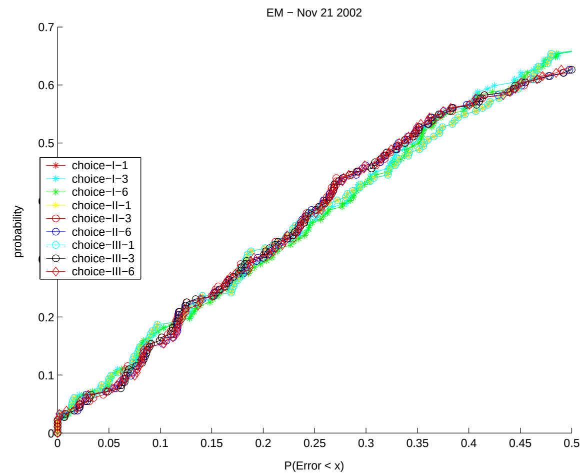

Figure 14: Convergence of EM method (prior set II)

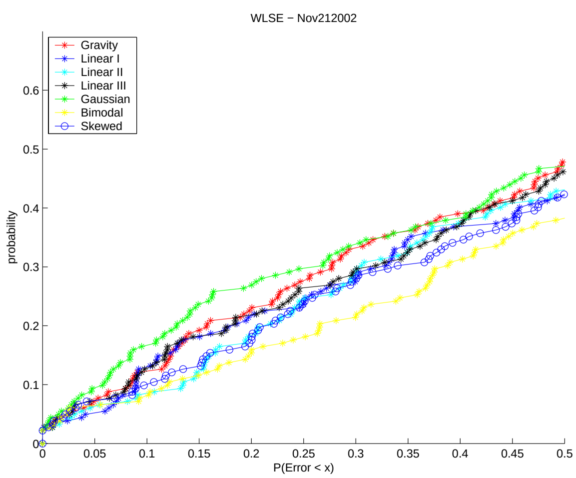

Figure 13: Convergence of WLSE method (prior set I)

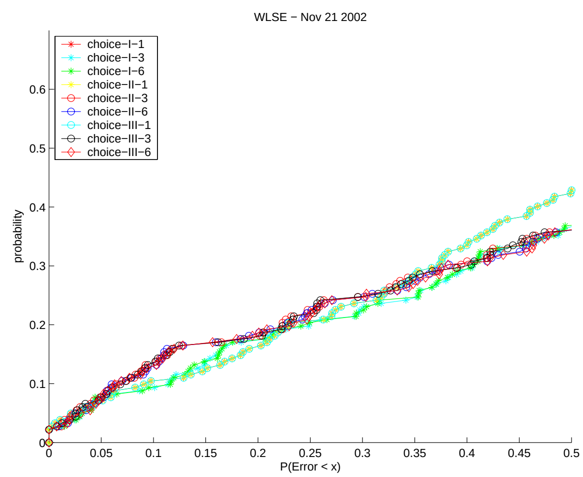

Figure 15: Convergence of WLSE method (priors set II)

the EM and WLSE methods. Figures 16 through 25 show the behavior of the methods for all estimated OD pairs. These figures contain both the target TM and the resulting estimated TM in the same plot. The target TM is sorted by OD pair size, and the estimated TM is plotted according to the corresponding order.

Figures 16 and 17 show the estimation results when a skewed starting point is used. The WLSE estimated points loosely follow the trend exhibited by the target TM although there is a lot of variability among them. In contrast, the EM estimated points are closer to the target curve and the error is much less variable. Figures 18 and 19 show the estimation results when a starting point generated according to a gravity model is provided. For WLSE the estimation results do not vary qualitatively. From the constant-target experiments we observed that the EM method showed more robustness to variations in the type of starting point provided. Consistently, we observe how the estimation results for the EM improve slightly with respect to the skewed prior but with no significant difference. Figures 20 and 21 show similar results for the case when choice-model starting points are provided. We observe in this case that WLSE produces better estimates and their variability is reduced. The EM results remain consistent with a slight observed improvement.

An alternative angle to look at the evaluation results consist of plotting the values of the estimated TM versus the values of the target traffic matrix. The more aligned are the plotted points to a 45-degree line, the better is the estimation. The span of the estimated points around such line is an indication of the estimation errors. Figures 22 through 25 show these kind of plots for the gravity-model and choice-model priors. We can observe how for the gravity-model starting point, the WLSE estimation points are more scattered and show a lower linear trend as compared to the EM estimation points.

Recall we use the Pearson's correlation coefficient, R, for assessing the quality of the estimations. For the WLSE method, the value of R is around 0.3 when gravity-based starting points were used, around 0.2 for choice-based starting points, and around 0.1 when *unreasonable* starting points were used. In contrast, for the EM method the value of R was around 0.8 independently of the starting point used.

8.3 Real-data Evaluation

In this section we describe the results of experiments performed with limited but actual network traffic demands we measured directly from the studied network. These experiments were designed

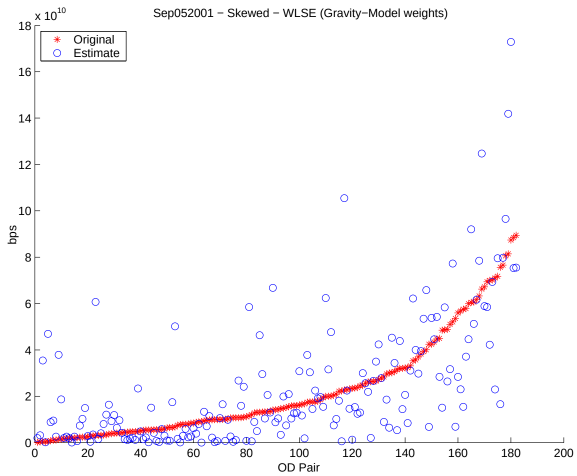

Figure 16: WLSE method with skewed prior

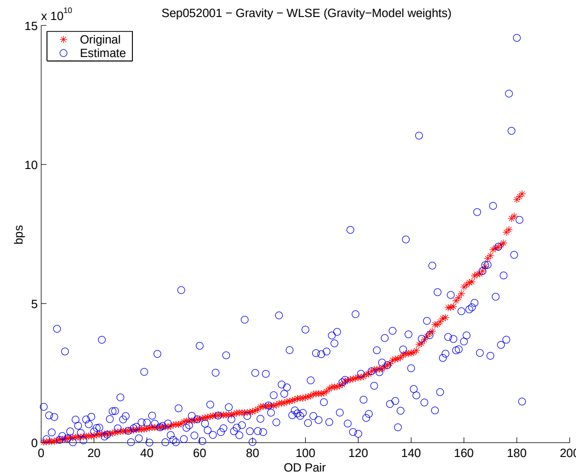

Figure 18: WLSE method with gravity prior

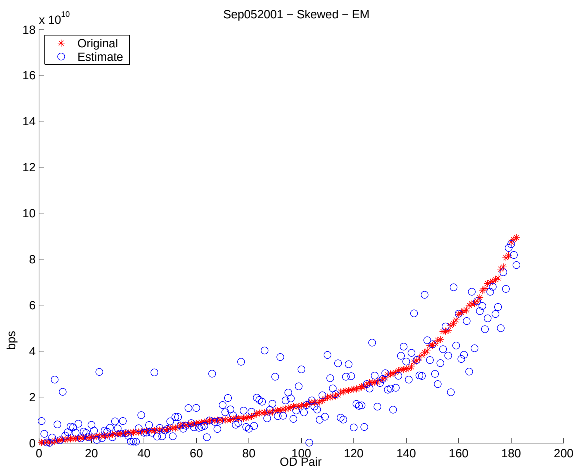

Figure 17: EM method with skewed prior

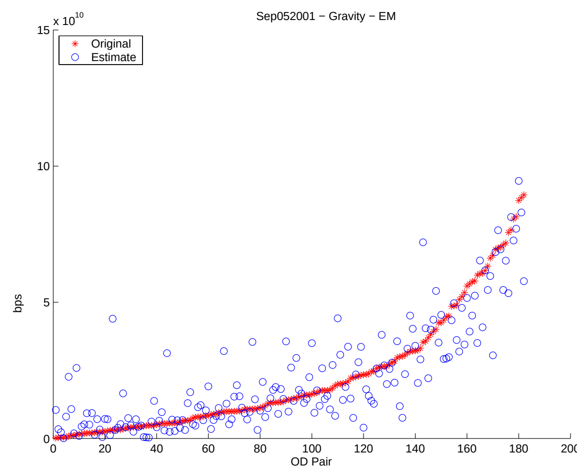

Figure 19: EM method with gravity prior

as explained in Section 6.3. We run the statistical techniques providing actual SNMP counts and a starting point generated from a model calibrated and populated with actual data for the same time period as the time during which packet traces were collected. Specifically, we use packet traces as described in Section 5.1 to compute measured TM rows for two POPs (November 21^{st} data) or three POPs (September 5^{th} data) — starting point models are calibrated using 1 or 2 POP rows (corresponding to POPs in POP1, POP2, or POP3), leaving the remaining measured row for assessing the error after the TM estimation. Figures 26 through 29 show the results of these experiments. The results show EM yielding better results than WLSE specially at the right end of the curve, that is, at higher volumes of traffic demand, but in general the difference between both methods is less pronounced than in the case of full synthetic TMs (cf. Section 8.2). One reason is that the quality of the estimation is assessed here on only one row of the TM. We expect as we become able to obtain more complete measurements, the advantage of our EM algorithm would become more pronounced.

9. CONCLUSIONS AND FUTURE WORK

For estimating network traffic demands, our contribution in this paper is two-fold. First, we found that most starting-point TM models (e.g. choice and gravity-based) produce initial estimates that are within a reasonable error range from the target TM being estimated. Although a choice-based prior TM produces slightly better estimates than a gravity-based one, the latter is simpler in that no calibration with real data is required. It will be up to carriers to decide on their individual tradeoffs between accuracy and simplicity. Unlike arbitrary models (e.g. skewed and bimodal), these starting-point models are informed by partial SNMP data. Such informed starting-point TM models are crucial for the success of statistical estimation techniques such as EM and WLSE.

Second, we introduced a new EM algorithm, which is much faster than conventional implementations. This expands its iteration range in search for global optima, and makes the algorithm less sensitive to the quality of the starting point. Our EM algorithm consistently produces estimates which outperform that of the WLSE approach by about 25%.

In this paper we compared choice starting-point models to an alternative simple gravity model for PoP-to-POP TMs. In the future we intend to compare against the extended gravity model proposed in [15]. This extended gravity model allows one to isolate separate traffic matrices such as peer-to-customer, customer-to-customer, and customer-to-peer TMs. We intend to study router-to-

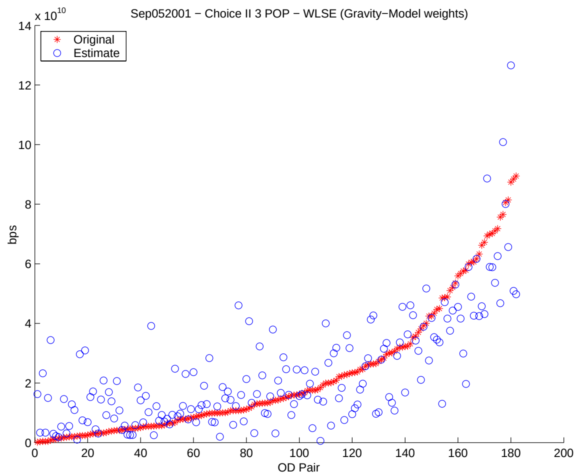

Figure 20: WLSE method with choice-model prior

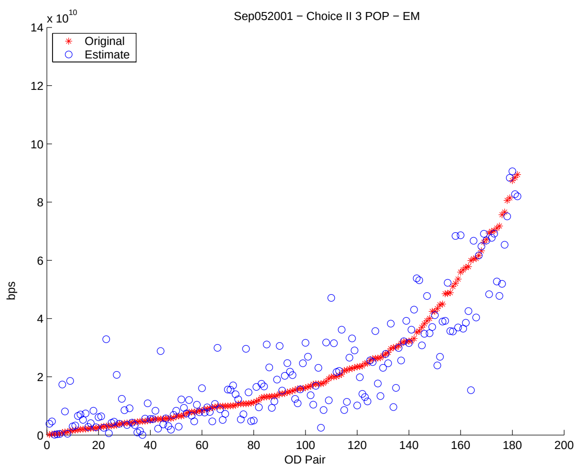

Figure 21: EM method with choice-model prior

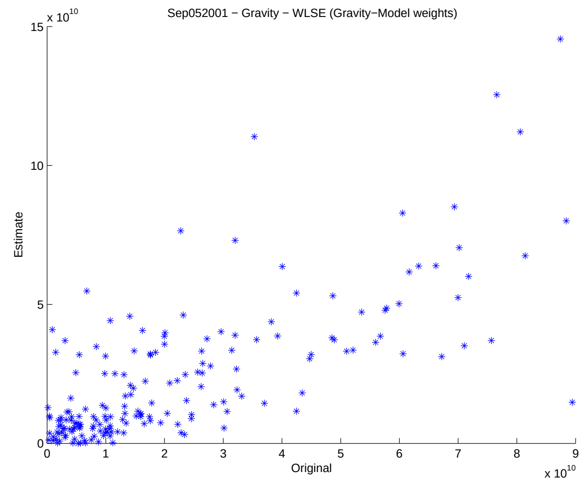

Figure 22: WLSE method with gravity prior

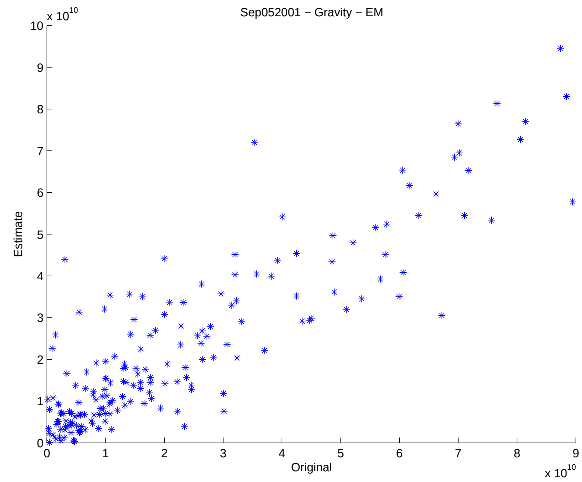

Figure 23: EM method with gravity prior

router level TMs. This is now possible given that both our fast EM algorithm and the WLSE algorithm would scale to work on much larger TMs.

10. REFERENCES

[1] R.J. Bouchard and C.E. Pyers. The use of the Gravity Model for Forecasting Urban Travel: An Analysis and Critique. In *43rd Anuual Meeting of the Highway Research Board*, January 1964.

[2] J. Cao, D. Davis, S. Vander Weil, and B. Yu. Time-Varying Network Tomography. *J. of the American Statistical Association.*, 2000.

[3] CISCO. Simple Network Management Protocol. Technical report, Cisco, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm.

[4] S.C. Dodd. The Interactance Hypothesis: A Gravity Model Fitting Physical Masses and Human Groups. *American Sociological Review*, pages 245–256, April 1950.

[5] O. Goldschmidt. ISP Backbone Traffic Inference Methods to Support Traffic Engineering . In *Internet Statistics and Metrics Analysis (ISMA) Workshop*, San Diego, CA, December 2000.

[6] Peter Kennedy. *A Guide to Econometrics*. MIT Press, Cambridge, MA. USA, 1998.

[7] J. P. LeSage. Econometrics Toolbox. Technical report, University of Toledo, 2001.

[8] A. Medina, C. Fraleigh, N. Taft, S. Bhattacharyya, and C. Diot. A Taxonomy of IP Traffic Matrices. In *SPIE Workshop on Scalability and Traffic Control in IP Networks II*, Boston, MA, July 2002.

[9] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic Matrix Estimation: Existing Techniques Compared and New Directions. In *ACM SIGCOMM*, Pitsburgh, PA, August 2002.

[10] W.J. Reilly. *The Law of Retail Gravitation*. Fillsburg Publishers, New York, 1953.

[11] C. Tebaldi and M. West. Bayesian Inference of Network Traffic Using Link Count Data. *J. of the American Statistical Association.*, pages 557–573, June 1998.

[12] Traceroute. Network utility to trace paths taken by packets. Technical report, Berkeley Lab, ftp://ftp.ee.lbl.gov/.

[13] Y. Vardi. Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data. *J. of the American Statistical Association.*, pages 365–377, 1996.

[14] R.E. Whitmore. Graphical and Mathematical Investigation of the Differences in Traveltime Factors for the Gravity Model Trip Distribution Formula in Several Specific Urban Areas. Technical report, Civil Engineering Department, University of Tennessee, 1965.

[15] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Loads. *To appear in ACM SIGMETRICS*, 2003.

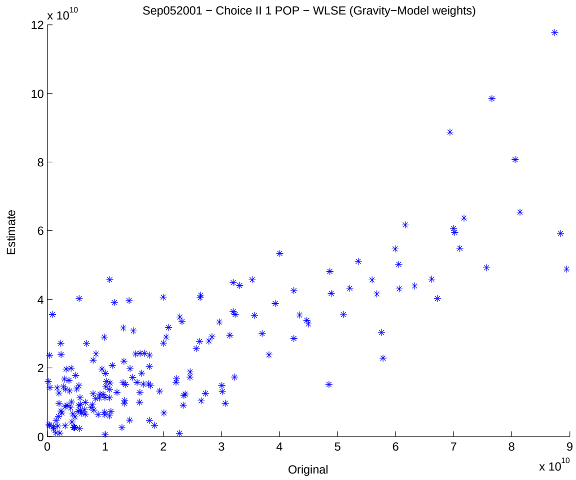

Figure 24: WLSE method with choice-model prior

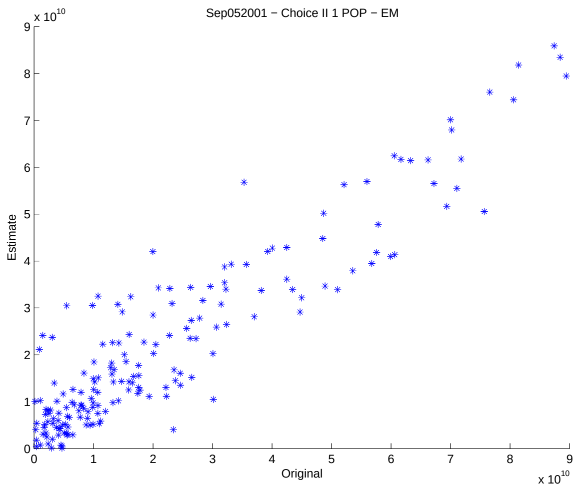

Figure 25: EM method with choice-model prior

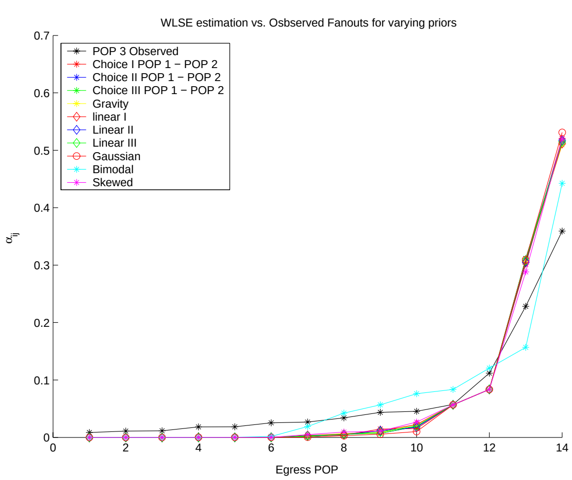

Figure 26: WLSE estimation of measured POP3 TM row

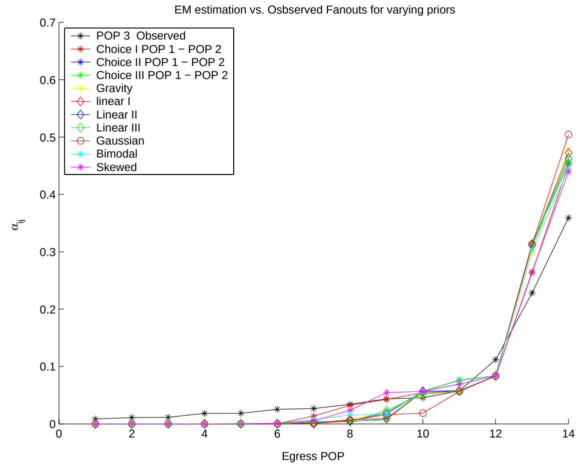

Figure 27: EM estimation of measured POP3 TM row

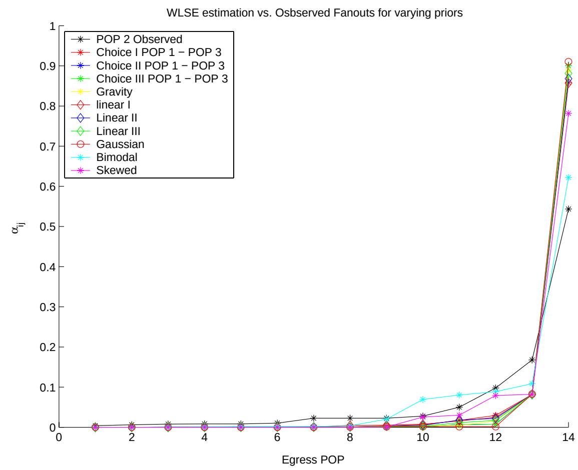

Figure 28: WLSE estimation of measured POP2 TM row

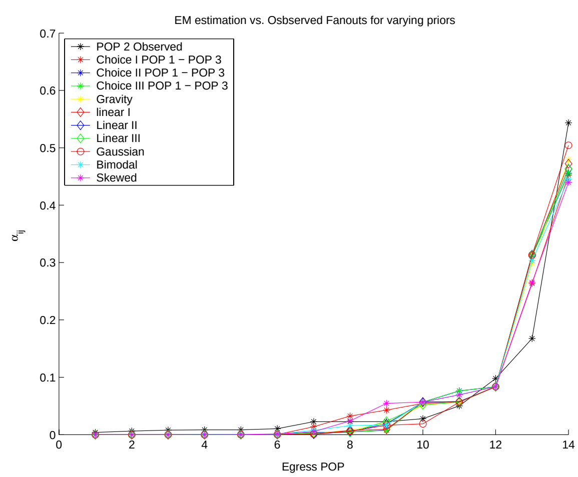

Figure 29: EM estimation of measured POP2 TM row