Inferring Invisible Traffic

Vineet Bharti[1], Pankaj Kankar[1], Lokesh Setia[1], Gonca Gürsun[2], Anukool Lakhina[1], and Mark Crovella[2]

[1]Guavus Network Systems
[2]Department of Computer Science, Boston University

ABSTRACT

A traffic matrix encompassing the entire Internet would be very valuable. Unfortunately, from any given vantage point in the network, most traffic is invisible. In this paper we describe results that hold some promise for this problem. First, we show a new characterization result: traffic matrices (TMs) typically show very low effective rank. This result refers to TMs that are purely spatial (have no temporal component), over a wide range of spatial granularities. Next, we define an inference problem whose solution allows one to infer invisible TM elements. This problem relies crucially on an atomicity property we define. Finally, we show example solutions of this inference problem via two different methods: regularized regression and matrix completion. The example consists of an AS inferring the amount of invisible traffic passing between other pairs of ASes. Using this example we illustrate the accuracy of the methods as a function of spatial granularity.

1. INTRODUCTION

In many situations it is desirable to form a volume estimate for traffic that is not directly observable. On a practical level, traffic measures are useful for capacity planning, performance analysis, and traffic engineering, and each of these tasks must sometimes be performed in the absence of directly measurable data. On a scientific level, improved knowledge of how traffic flows through the Internet as a whole can inform our understanding of how demand, topology, and economics interact to shape Internet evolution. Unfortunately for the latter problem, the very nature of the Internet's structure means that there is no single location (or even reasonably-sized set of locations) from which a comprehensive picture of Internet traffic can be obtained. A key
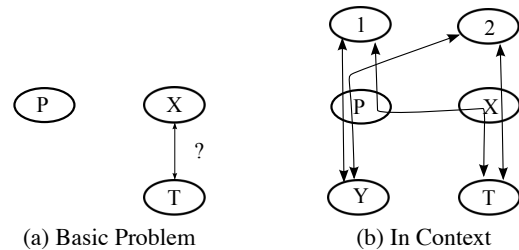
Figure 1: Invisible Traffic

aspect of the problem is that large-scale measurements of Internet traffic are only readily available to ISPs, and each ISP can only directly measure the traffic flowing through its own network.

In this paper we show that an ISP's view of Internet-wide traffic may not be as limited as it appears. We show that measurements of the traffic flowing through a given network additionally contain significant information about traffic that does *not* flow through the network. In some cases, this information is sufficient to form accurate estimates of the amount of traffic flowing through a *different* network.

An example of the kind of problem we are interested in is shown in Figure 1(a). Network P would like to estimate the amount of traffic flowing between networks T and X. However, this traffic does not flow through P and so P cannot directly measure it.

Of course, in the situation shown in Figure 1(a), it seems impossible for P to obtain the desired knowledge. However, consider the case in which some other traffic does flow between P and T, perhaps because P and X are peers. For example, traffic might flow according to the more detailed picture shown in Figure 1(b). In this figure, traffic is being sent through P and X to destinations 1 and 2. (Of course, although only two destinations are shown, in reality there can be many destinations; likewise P and X can each have many other customers.)

Note that Provider P has a Customer Y whose traffic to both 1 and 2 is visible to P. Perhaps by observing Customer Y's traffic, Provider P can learn how traffic "typically" flows toward destinations 1 and 2. And, because T's traffic for destination 1 also flows through P, traffic from T to 1 is vis-

ible to P as well. Intuition suggests that in a situation such as this, it may well be possible for Provider P to form an inference about the traffic flowing between X and T.

In this paper we show that this intuition holds in practice. To do so we need to show two things: first, that relationships between traffic flows exist that could form a basis for inference; and second, that statistical methods can extract this information to make accurate estimates of invisible traffic.

To that end, we start by setting up a framework that allows us to construct well-defined estimation problems. We show that the first challenge is the construction of *atomic indexings,* a notion that we define. We then identify a number of atomic indexings and use them to construct traffic matrices measured in two networks (P and T) over a 14-day period. Our first result is to show that the TMs we measure show *low effective rank* – i.e., there are strong correlations between columns (or rows), such that a measured TM can be approximated by a matrix having relatively small rank. We then identify matrix elements that are visible in T, but not in P, as targets for estimation by P. Next we identify statistical methods that are appropriate for estimating those invisible elements, given the low effective rank of TMs. We show the relative performance of the various methods, and their overall accuracy in estimating both individual matrix elements as well as the aggregate of all traffic passing through T. We also show that for the methods we use, accurate estimation requires a relatively high level of aggregation, which tends to conflict with the requirement of atomicity. However, we also show that P can form surprisingly accurate estimates of the *total* invisible traffic passing through T. We also illustrate situations where estimation tends to fail; these illustrations suggest a number of future challenges.

2. RELATED WORK

The term *traffic matrix* has been used for two notions in the literature. In the first case, a TM is a purely-spatial concept: a set of measurements of traffic flowing between some sources and some destinations during a fixed time period. This is the sense as used in [13, 20]. In the second case, a TM is in fact a time series of (purely-spatial) TMs. This is the sense as used in [12, 21]. The difference is significant because in temporally sequenced TMs, there is a very strong correlation structure along the time dimension. Past work has established the low effective rank of such data (e.g., [12]), without distinguishing between the effects of temporal and spatial correlation. In fact most of the work using these low rank results has implicitly depended strongly on this temporal correlation.

In contrast, in the problem discussed in this paper, the data has no time dimension. Hence our results depend only on spatial correlation. For this reason, documenting the spatial correlation present in the data is an important first step, and so forms a part of the results we report in this paper.

The previous work that is closest to ours is reported in [21]. That work also seeks to infer missing TM measurements, specifically those that are missing due to equipment malfunction, etc. Thus the approach taken in that paper can make use of temporal correlation, which improves accuracy considerably. While making use of temporal correlation may be helpful in our setting, we expect its contribution to be much more limited than in [21]. This is because generally, in our case a TM element to be estimated is rarely or never visible over time, and so past history is less useful than in [21]. Additionally, that work focuses on a network's internal measurements, rather than measurements of traffic flowing through other networks; hence it does not need to cope with the issues of atomicity that are central to the problems we deal with in this paper.

The general challenge we address in this paper is similar to the matrix completion problem, which has received considerable attention recently. We review the matrix completion problem in Section 6.2; here we note that our problem differs in certain ways from the standard matrix completion setting [4, 10]. In particular, the matrices we work with are not uniformly sampled; the usable entries correspond to visible elements and these are irregularly scattered, reflecting local routing. Furthermore, our TM entries tend to show high variability which has the potential to invalidate uniformity assumptions required by matrix completion methods.

Besides our work, other papers have looked at the problem of estimating interdomain traffic demands. The work described in [8] estimates only the portion of traffic that is Web related, and depends on the availability of server logs from a large content delivery network provider. The work described in [5] estimates properties of ASes such as population size and AS role and uses these to form estimates of interdomain traffic volume.

Finally, we note that the low effective rank of TMs within individual ASes has been implicitly or explicitly assumed in some previous work. The *gravity models* that have been used in TM modeling in the past (e.g., [19]) are rank-1 models. The authors in [6] find that a rank-2 model is a good fit to measured TMs, and the author in [16] shows that a rank-1 model is a good fit to measured TMs. However, these papers do not ask the question of what rank model is the best fit to measured TMs, which is a focus in this paper.

3. FRAMEWORK, GOALS, AND CHALLENGES

In order for traffic measurements collected in one network to be useful in estimating traffic passing through another network, the available data must be placed into a framework yielding a well-defined estimation problem. To set out that framework we start with some definitions.

In what follows, we use bold symbols to refer to matrices ($\mathbf{M}$) and vectors ($\mathbf{x}$). Vectors are column vectors, and $\|\mathbf{x}\|$ denotes the ℓ_2 norm of $\mathbf{x}$.

3.1 Basic Definitions

We are concerned with using traffic measurements made

in a *Predictor* network (P) to estimate traffic measurements that would be seen in a *Target* network (T) (e.g., as shown in Figure 1(b)). To do this, we need to distinguish between a *view*, a *traffic matrix*, and an *indexing*.

For a given network P, consider an interval during which there are no changes in external routing. Then there is a set of IP source-destination pairs (s, d) such that if s generates traffic and sends it to d, that traffic will pass through P. This set is called the network's *view* during that interval.

A *traffic matrix* (TM) is an $m \times n$ matrix $\mathbf{M}(\ell)$ in which $M_{ij}(\ell)$ is a measure of some subset of the traffic flowing from a set of addresses $\mathscr{S}_i$ to a set of addresses $\mathscr{D}_j$ during a specific time interval $\{t \mid t_\ell \leq t < t_{\ell+1}\}$. In this paper time intervals will be one day or one week; for brevity we will omit the time index ℓ in what follows. The elements $\{M_{ij}\}$ may be any traffic measure; in this paper we consider two: the number of bytes and the number of packets.

In general, a TM need not be specific to a particular network. However we will work with traffic matrices measured in particular networks. When necessary, we will use superscripts to distinguish TMs measured in different locations, so $\mathbf{M}^{(T)}$ is measured in the Target network, and $\mathbf{M}^{(P)}$ is measured in the Predictor network.

An *indexing* is a particular choice $\mathscr{I} = (\mathscr{S}, \mathscr{D})$ with $\mathscr{S} = \{\mathscr{S}_i \mid i \in 1..m\}$ and $\mathscr{D} = \{\mathscr{D}_j \mid j \in 1..n\}$. Each $\mathscr{S}_i$ and each $\mathscr{D}_j$ is a set of IP addresses. We assume that the sets $\mathscr{S}_i$, $i = 1..m$ are disjoint, as are the sets $\mathscr{D}_j$, $j = 1..n$. If $\mathscr{S}$ and $\mathscr{D}$ each form a partition over the entire routable address space, then a TM with indexing $\mathscr{I}$ is a *full* TM. Otherwise it is a *partial* TM.

The way these definitions work together is as follows. At any moment, the state of the interdomain routing system determines the network's view. The traffic actually flowing through the network consists of whatever traffic happens to flow between pairs of IP addresses in the network's view. Applying an indexing to the traffic flowing through the network yields a traffic matrix for the interval.

Consider a particular network N in which we measure a TM $\mathbf{M}$ having indexing $\mathscr{I} = (\mathscr{S}, \mathscr{D})$. Element M_{ij} is *fully visible in N* if, for all $(s, d) \in \{\mathscr{S}_i \times \mathscr{D}_j\}$, (s, d) is contained in N's view. Likewise, element M_{ij} is *invisible in N* if no $(s, d) \in \{\mathscr{S}_i \times \mathscr{D}_j\}$ is contained in N's view. Elements that are neither fully visible nor invisible in N are *partially visible*.

3.2 Atomicity

In order to fix the estimation problem, it is necessary to distinguish between TM values that are known and those that are unknown. Only known TM values are useful as input to the estimation problem.

To be more precise, if a TM element is fully visible in the predictor network, then it is useful as an input. On the other hand, we do not want to use partially-visible elements as inputs. Partially-visible elements may contain some useful information, but they introduce complexity that we want to

avoid. Of course invisible elements are not useful as inputs.

Ideally, if the view for the predictor network were known, then one could use any indexing and mark the fully-visible elements as known and the rest as unknown. However this is difficult in practice. The reason has to do with the organization of the interdomain routing system, and where knowledge of the state of the routing system actually resides. Consider Figure 1(b). If T changes its routing so that traffic to 1 goes through some provider other than X (and hence no longer passes through P), BGP does not require that fact to be announced to P. Thus it is actually difficult for P to know what its view is, based on the information provided to it by the routing system. This makes it a challenge to identify the visibility status in P (fully visible, invisible, or neither) of any given element in an arbitrary indexing.

In practice, there is a way around this problem – but only partially. The idea is to define an indexing for P's TM such that each element is either invisible, or fully visible. Such an indexing is *atomic for P* – it has no elements that are partially visible in P. Formally, given a particular network P, an indexing $\mathscr{I}$ is atomic for P if, in a TM $\mathbf{M}$ using indexing $\mathscr{I}$, each element M_{ij} is either fully visible in P, or invisible in P.

When using an atomic indexing, P still does not know *a priori* the visibility status of any given element. However when P observes traffic, it learns that some elements are in fact fully visible. So for TM $\mathbf{M}$ using an indexing that is atomic for P, if $M_{ij} \neq 0$, then it is fully visible in P. In other words, if $M_{ij} \neq 0$, then M_{ij} represents *all* the traffic flowing from addresses in $\mathscr{S}_i$ to addresses in $\mathscr{D}_j$. Note though that if $M_{ij} = 0$, its visibility status in P is still unknown.

We also note that even if an indexing is atomic for network P, the above argument will not hold in general if P's view changes during the measurement interval. For example, if element (i, j) changes from fully visible to invisible during the measurement period, we can have $M_{ij} \neq 0$ and yet M_{ij} may not represent all corresponding traffic for that interval. In everything that follows, we assume that each network's view does not change during the measurement period. This assumption is undoubtedly violated for our measurements, but quantifying its impact is outside our scope.

3.3 Problem Definition

Using the notion of atomicity, we can define two versions of our estimation problem.

> PROBLEM 1: TRAFFIC MATRIX COMPLETION. Given an indexing $\mathscr{I}$ that is atomic for P, and given a TM $\mathbf{M}$ with indexing $\mathscr{I}$ measured in P, estimate the elements of $\mathbf{M}$ that are invisible in P.

> PROBLEM 2: ALTERNATE VIEW CONSTRUCTION. Given an indexing $\mathscr{I}$ that is atomic for two networks, P and T, and given a TM $\mathbf{M}$ with indexing $\mathscr{I}$ measured in P, estimate the elements of $\mathbf{M}$ that are invisible in P and

visible in T.

Problem 1 is more general, and if **M** were a full TM, then solving Problem 1 would be tantamount to estimating a TM for the entire Internet. However, by its nature, validating a solution to Problem 1 is difficult. Hence we focus on Problem 2, which nonetheless has considerable practical significance (as described in Section 1), and whose solution is easier to validate. Further, our hope is that by exploring solutions to Problem 2 we can form a foundation for solving Problem 1.

Our focus in this paper is on statistical estimation of individual traffic matrix elements. However, one complication in addressing Problem 2 is that we need to know *which* TM elements to estimate – that is, we need to know T's view. As discussed in Section 3.2, knowing an AS's view requires learning some of the state of the BGP system. This can be addressed by various methods that are outside our scope. For example, in the case where T is a "stub" AS, T's view can be learned through analysis of T's BGP announcements. In other cases, it may be possible to learn T's view through analysis of collected BGP tables (e.g., from Routeviews) at least in an approximate manner. In this paper, we use measurements from T to approximately infer its view, but it is important to emphasize that is simply one of many possible methods for inferring T's view, and that inferring T's view is not our focus.

We refer to the elements of **M** that are visible in P as *predictors* and the elements to be estimated as *targets*.

3.4 Atomicity Versus Aggregation

An input to our problem is $\mathscr{I}$, an indexing that is atomic for both P and T. In practice, we need to find such an indexing, and the choice of indexing can have a significant effect on the tractability of the problem.

Finding indexings that are atomic for two networks seems hard without knowledge of the global routing system state. However, one way around this problem is to define indexings that are atomic for *all* networks. In fact, there are multiple such *globally-atomic* indexings, and one of our goals is to study their relative merits.

We can define four indexings that are (at least approximately) globally-atomic. These differ in the degree of *aggregation* (i.e., spatial granularity) — the number of source-destination pairs that are assigned to each TM element. There are two main benefits to increasing aggregation. First, it reduces data size; coarser-grained indexings produce smaller TMs, which reduces computational demands. Second, as we will show later, increasing the level of aggregation generally improves the quality of the solutions we can obtain.

IP-IP. Consider the indexing $(\mathscr{S}, \mathscr{D})$ in which each routable IP address constitutes a distinct element $\mathscr{S}_i$ and a distinct element $\mathscr{D}_j$. Then clearly, this indexing is atomic for any network, because traffic for each IP address pair $(\mathscr{S}_i, \mathscr{D}_j)$ either passes through a given network, or not.

As already mentioned, there are two difficulties with this simple approach that immediately present themselves: *noise* and *scale*. The nonzero elements of **M** will typically be very small (for reasonable timescales ℓ, e.g., a few days or weeks), and will show extremely high variability. Furthermore, **M** is immense – on the order of 500 million $\times$ 500 million elements [1]. Together these mean that the resulting estimation problem will be difficult to solve.

Prefix-Prefix. At a coarser level, one can group addresses according to the longest matching network prefixes that are advertised in the BGP system. In this case each $\mathscr{S}_i$ and $\mathscr{D}_j$ correspond to all of the addresses matching a particular prefix. The nature of interdomain routing via BGP dictates that indexing at the prefix-prefix level will be atomic for any network. This is because all the addresses matching a prefix are routed the same way in any given router. Aggregation to this level reduces the maximum size of **M** to the order of 300K $\times$ 300K [15].[1]

Atom-Atom. A further coarsening is possible through empirical analysis of BGP tables. Observations show that there are many cases in which a collection of prefixes is routed the same way in any given router. Such collections are called *BGP policy atoms* ('atoms' for short) [2]. We constructed atoms using the code made available by CAIDA [3] and applied to BGP tables from the Route Views Project [15]. While an atom-atom indexing will usually be atomic, it is not guaranteed since atoms are defined based only on forward BGP paths, and using a necessarily-incomplete set of BGP tables. Nonetheless, we speculated that atom-atom indexings are close enough to being atomic to be useful. Using an atom-atom indexing reduces the maximum size of **M** to approximately 66K $\times$ 66K.

AS-Atom. Finally, we made use of a further coarsening of the indexing. We note that in many cases, an AS only uses a single-next hop AS for any given prefix. Although there are notable violations of this pattern [14], we again speculated that they were rare enough in our data to make AS-atom indexing approximately atomic. Aggregation to this level reduces the maximum size of **M** to about 32K $\times$ 66K.

4. DATA USED

The data we use in this paper is taken from two networks, denoted P and T. The general relationship between these networks and the rest of the Internet is shown in Figure 2. Network T is a customer of P as well as other networks. Network T has a number of customers, some of which also have customers. Also, network P has a number of customers, and so on. All of the links in the figure are customer-provider links, except the link between T and O, which is a peering

---

[1]For this as well as the other aggregation levels, we used prefixes as collected by the Route Views Project; these are subject to prefix aggregation which can interfere with atomicity if our measured networks use less-aggregated prefixes internally. Experiments we performed with prefixes taken from the RIBs in our networks suggested that this was not a significant source of error.

Indexing	Size	Density of P	Density of T	Density of Targets
Prefix-Prefix	7376×7431	0.0509	0.0070	0.0059
Atom-Atom	3183×3116	0.0428	0.0031	0.0025
AS-Atom	2156×3116	0.0517	0.0039	0.0030

Table 1: TM Sizes Used. Density is for Jan. 5 (Day 1).
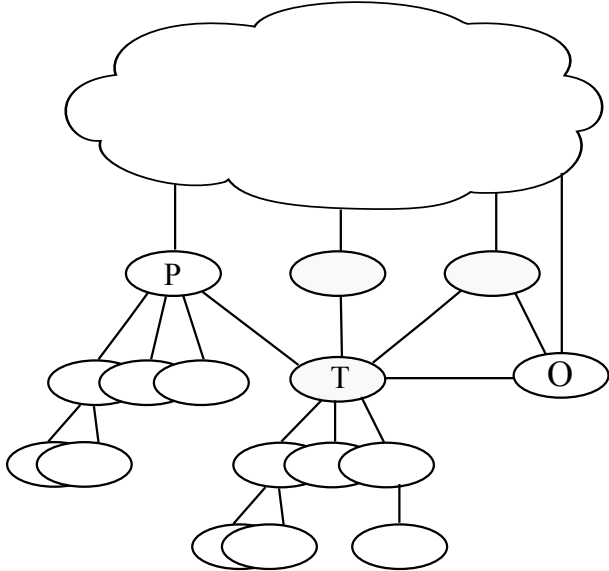
Figure 2: Topological relationships of networks used.

Figure 3: Total Traffic in P and T (in Packets).
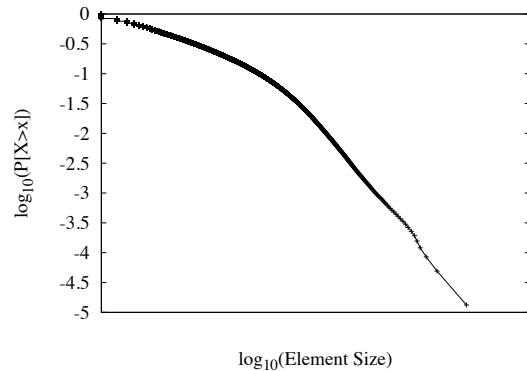
Figure 4: Log-Log Complementary Distribution of Sizes of Elements of $\mathbf{M}^{(P)}$, Jan. 5.

link. The significance of this link and the T-O relationship will be discussed in Section 7.3.

Figure 3 shows the total amount of traffic passing through P, T, and the amount of traffic that passes through both, for each day of our measurement period. The figure shows that about 5% of P's traffic consisted of traffic that passed through T, and about 25% of T's traffic passed through P during our measurement period. Note that in order to protect proprietary information, we do not label traffic volumes in our results.

We collected complete `netflow` logs from all edge routers in both networks over the 14 day period January 5, 2010 – January 19, 2010. Measurements in P were based on 1:100 sampling; in T some routers sampled at 1:1, some at 1:100 and some at 1:200. All results reported here are based on inverting sampling in the straightforward way by multiplying each flow's stated volume by the sampling rate of the router reporting the flow. Multiple counting of flows was eliminated, so that what remained was a record of each sampled flow that passed through P or T. Analysis (not shown) indicates that the uncertainty in traffic measures due to the inversion of sampling is small compared with the overall accuracy of our methods.

For each of the 14 days in our measurement period, we constructed traffic matrices using 3 globally-atomic indexings: prefix-prefix, atom-atom, and AS-atom. We then chose
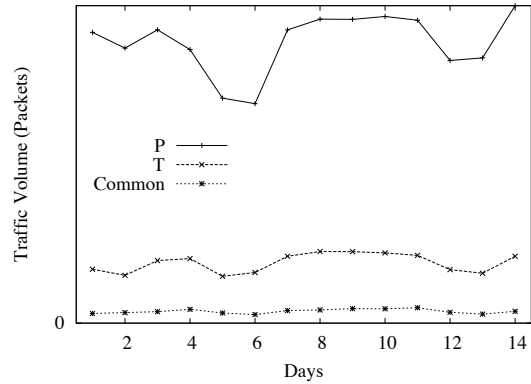
a subset of rows and columns that captured most of the traffic present in each TM (86-90% for T, and 82-83% for P). We used the same set of rows and columns for each day's TM. The final TM sizes and density (number of nonzero entries) are shown in Table 1.

As has been noted in previous studies, most traffic is concentrated in a small number of matrix elements [11, 7, 17]. The sizes of matrix elements show high variability characterized by a long distributional tail. The tail of the size distribution of elements for the Prefix-Prefix matrix measured in P on January 5 is shown in Figure 4.

5. EFFECTIVE RANK

In this section we present our primary characterization results, which are that the spatial traffic matrices we use appear
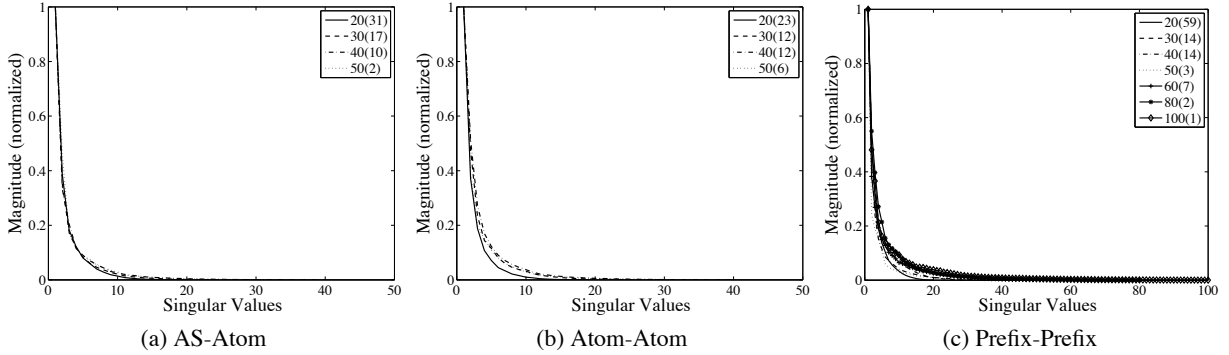
| | (a) AS-Atom | (b) Atom-Atom | (c) Prefix-Prefix |

Figure 5: Spectral Plots of TMs as a Function of Size

to show low effective rank.

A matrix $\mathbf{M}$ has *effective rank r* if $\mathbf{M}$ can be approximated by a rank-r matrix $\mathbf{M}'$, that is, if there exists a rank-r matrix $\mathbf{M}'$ such that $\sum(\mathbf{M}_{ij} - \mathbf{M}'_{ij})^2$ is suitably small. A direct way to test this is to form the singular value decomposition of $\mathbf{M} = \mathbf{U\Sigma V^T}$, and extract the singular values from the diagonal of Σ (the eigenspectrum of $\mathbf{M}$). These give a measure of how much each additional increase in rank improves an optimal approximation of $\mathbf{M}$. If it is the case that beyond the first r singular values the remaining singular values are all small, we can conclude that $\mathbf{M}$ has effective rank r.

To assess the effective rank for spatial traffic matrices such as ours, we developed a simple method to extract dense square submatrices from our sparse matrix $\mathbf{M}^{(P)}$.[2] We then extracted a large set of non-overlapping submatrices from the prefix-prefix, atom-atom, and AS-atom matrices for Jan. 5. The matrices we extracted vary in size from 20×20 up to 100×100. We then computed the eigenspectrum of each extracted matrix, and normalized each spectrum to set its largest singular value to 1.

We are particularly interested in the relationship between matrix size and effective rank, so we then averaged the eigenspectra for each matrix size. The resulting averaged spectra are shown in Figure 5. The figure label shows the size (number of rows or columns) of each matrix as well as the number of non-overlapping matrices whose spectra were averaged into the curve shown.

The figure compares the averaged eigenspectra for different-sized matrices. There are two striking characteristics: first, the effective rank does not seem to vary appreciably as a function of matrix size. In most cases we can conclude that the effective rank is approximately 5 or less. Second, the effective rank does not seem to vary appreciably with the degree of aggregation: prefix-prefix, atom-atom, and AS-atom matrices all show similar eigenspectra.

Hence we find that dense submatrices of $\mathbf{M}$ typically show very low effective rank. While this does not directly inform

---

[2]This simply consists of finding a set of rows and columns of $\mathbf{M}^{(P)}$ such that the resulting submatrix has no zero elements. No elements are inferred in this process.
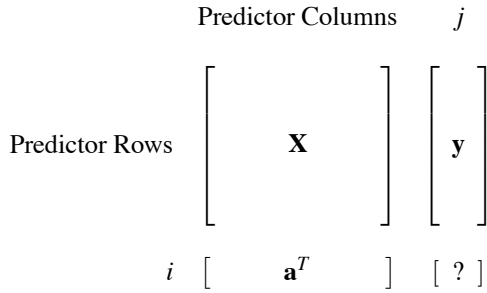
us about the effective rank of $\mathbf{M}$ as a whole, the consistency of our results with respect to size and aggregation level suggest that other submatrices of $\mathbf{M}$ are also likely to show low effective rank.

6. METHODS

The previous section showed that there is typically considerable structure in a TM that may be used for prediction purposes. We took two approaches to exploiting that structure: element-by-element linear estimation, and matrix completion.

6.1 Elementwise Linear Estimation

The fact that TMs show low effective rank means that a typical column can be expressed as a linear combination of a small number of orthogonal vectors. This suggests using linear estimators. A linear estimator takes the form

$$\hat{\mathbf{t}} = \mathbf{A}\beta$$

where the elements of $\mathbf{A}$ are model inputs, the elements of β are the model parameters, and elements of $\hat{\mathbf{t}}$ are the model outputs (the estimations).

To train such an estimator we need to find parameters β, and to use the estimator we need inputs $\mathbf{A}$. Both of these need to be derived from the elements of $\mathbf{M}$ that are visible in P. Because the visible elements are irregularly scattered in $\mathbf{M}$, the simplest approach is to construct a separate linear estimator for each target. The estimator then is

$$\hat{t} = \mathbf{a}^T \beta \qquad (1)$$

where t is a scalar, and $\mathbf{a}$ and β are column vectors.

To train the model (estimate β) we need training data. Thus, for each target t having position (i, j) in $\mathbf{M}$, we need to construct the tableau as shown in Figure 6. That is, we need to find predictor rows and columns in $\mathbf{M}$ such that $\mathbf{X}$, $\mathbf{y}$, and $\mathbf{a}$ are all visible in P. Having done that, we could then estimate $\hat{\beta}$ as the least-squares solution to $\mathbf{y} = \mathbf{X}\beta$, and then estimate the target element as $\hat{t} = \mathbf{a}^T \hat{\beta}$.

In our problem however, it is not a good idea to estimate β in the straightforward manner of least-squares solution of

$$\text{Predictor Columns} \qquad j$$

$$\text{Predictor Rows} \quad \begin{bmatrix} & & \\ & \mathbf{X} & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} \\ \mathbf{y} \\ \\ \end{bmatrix}$$

$$i \; \begin{bmatrix} \mathbf{a}^T \end{bmatrix} \; \begin{bmatrix} ? \end{bmatrix}$$

Figure 6: Tableau for Elementwise Linear Estimation

$\mathbf{y} = \mathbf{X}\beta$. The reason has to do with the nature of $\mathbf{X}$. Typically the columns of $\mathbf{X}$ are strongly correlated (indeed, that is the main implication of the results in Section 5). As a result, the value of β that minimizes $||\mathbf{X}\beta - \mathbf{y}||$ will be very unstable. Because there are many predictors that are carrying similar information, $\hat{\beta}$ will tend to be over-fit to the small differences between the predictors. The result is that $\hat{\beta}$, while an unbiased estimate of β, will have high variance. In our problem, this error arising from estimation variance makes the results obtained using straightforward least squares highly inaccurate.

There are a variety of methods for dealing with this situation [9, Ch. 3]. The general approach is to constrain $\hat{\beta}$ in some manner that makes it slightly biased, but with significantly lower variance. In doing so, the total error (bias squared plus variance) is reduced. We used two methods: principal components regression, and ridge regression.

6.1.1 Principal Components Regression

A first approach to reducing the variance of $\hat{\beta}$ comes from the realization that $\mathbf{X}$ has low effective rank: instead of regressing $\mathbf{y}$ against the columns of $\mathbf{X}$, we regress $\mathbf{y}$ against the few important (column) eigenvectors of $\mathbf{X}$. This is called *principal components regression* (PC regression).

To apply PC regression one chooses a value k and discards the $s - k$ least significant eigenvectors, where $s = \min(m, n)$ for $\mathbf{X}$ having size $m \times n$. This can be accomplished by singular value decomposition of $\mathbf{X}$: $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{X}$. Setting $\tilde{\mathbf{X}}$ to be the k columns of $\mathbf{U}$ with largest singular values (entries on the diagonal of $\mathbf{\Sigma}$), one then forms the PC regression estimate of β by least squares:

$$\hat{\beta} = \arg\min_{\beta} ||\tilde{\mathbf{X}}\beta - \mathbf{y}||$$

Thus, for any given value of k, PC regression is equivalent to setting the $s - k$ smallest singular values of $\mathbf{X}$ to zero before estimating β.

PC regression can be seen as a kind of smoothing or noise reduction in $\mathbf{X}$. This smoothing retains most of the predictive information in $\mathbf{X}$ but transforms that information into a set of orthogonal predictors $\tilde{\mathbf{X}}$ which, by nature of their lack of correlation, yield a more stable estimate of β.

6.1.2 Ridge Regression

A drawback to PC regression is that it works in discrete steps, corresponding to different choices of k. Ridge regression is a method that is similar in spirit to PC regression, but with a continuously-variable tuning parameter.

The idea behind ridge regression is that when β is unstable, its individual elements will typically be very large. Ridge regression imposes a penalty on large values of β:

$$\hat{\beta} = \arg\min_{\beta} ||\mathbf{X}\beta - \mathbf{y}|| + \lambda ||\beta||$$

with $\lambda > 0$. It can be shown that ridge regression is equivalent to the least-squares solution of

$$\hat{\beta} = \arg\min_{\beta} ||\tilde{\mathbf{X}}\beta - \mathbf{y}||$$

where $\tilde{\mathbf{X}}$ is formed from $\mathbf{X}$ as follows. Starting again from the SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, a new $\tilde{\mathbf{\Sigma}}$ is formed by *shrinking* the singular values:

$$\tilde{\sigma}_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda} \sigma_i \qquad (2)$$

and $\tilde{\mathbf{X}}$ is then constructed from $\tilde{\mathbf{X}} = \mathbf{U}\tilde{\mathbf{\Sigma}}\mathbf{V}^T$.

The transformation (2) has a greater effect on small singular values than on large singular values. Hence, ridge regression can be seen as similar to PC regression, but with an important difference. While PC regression sets the $s - k$ smallest singular values of $\mathbf{X}$ to zero, ridge regression shrinks all the singular values of $\mathbf{X}$, with a greater amount of shrinkage applied to the smaller singular values.

In this paper we plot results in terms of the *effective degrees of freedom* of the ridge regression fit. This is defined as:

$$\text{df}(\lambda) = \sum_{j=1}^{s} \frac{\sigma_j^2}{\sigma_j^2 + \lambda}$$

Note that $\text{df}(0) = s$, and when $\lambda \to \infty$, $\text{df}(\lambda) \to 0$.

Both PC regression and ridge regression introduce the additional need to determine the proper value of a tuning parameter (k or λ). Determining the best value of the tuning parameter to use in general can be approached via cross-validation: using $\hat{\beta}$ derived from $\tilde{\mathbf{X}}$ and $\mathbf{y}$, one estimates other, known values of $\mathbf{M}$. One then chooses the tuning parameter value that minimizes the resulting cross-validation error. We show examples in Section 7 showing that cross-validation can be effective for finding a good value of λ.

6.2 Matrix Completion

Matrix completion [4, 10] addresses the problem of recovering a low-rank matrix from a subset of its entries. Suppose $\mathbf{M}$ is an $m \times n$ matrix that has rank $r \ll \min(m, n)$ or that can be approximated by a rank r matrix. Assume that only a subset of $\mathbf{M}$'s elements $\Omega = \{(i, j)\}$ are known. If the set Ω contains enough information, and $\mathbf{M}$ meets a condition called *incoherence*, then there is a unique rank-r matrix that is consistent with the observed entries.
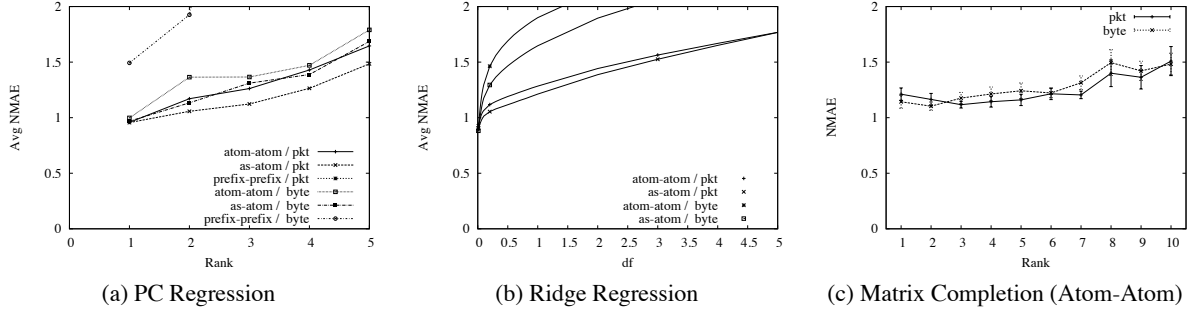
| (a) PC Regression | (b) Ridge Regression | (c) Matrix Completion (Atom-Atom) |

Figure 7: Estimating Individual Elements (NMAE).

Incoherence means that singular vectors of **M** are spread across all coordinates, i.e. they are not correlated with the standard basis vectors. In essence this means that the singular vectors are not 'spiky,' as would occur when a few entries of **M** are much larger than the others. The condition of sufficient information in Ω is met when the set Ω is sampled from the entries of **M** uniformly at random, and with sufficient density (of the order $O(r(m+n)\,\text{polylog}(m+n))$).

A variety of algorithms have been proposed for recovering **M**. Most rely on a convex optimization, namely, minimizing the nuclear norm (sum of the singular values) of a matrix **W** such that $W_{ij} = M_{ij}, (i,j) \in \Omega$. These approaches tend to be computationally demanding for large matrices. We used LMaFit [18] which does not rely on nuclear norm minimization, and which we found to be fast and robust.

Matrix completion methods assume that **M** and Ω are given. However in our case **M** may not meet the requirements of the method. In particular, the density of any given row or column of **M** is not guaranteed to be sufficient to allow estimation of its missing entries. Hence for each **M** we selected a submatrix **S** such that there were no less than c entries in each row and column of **S**. Increasing c represents a trade-off between increasing the information available for estimation and decreasing the number of entries that can be estimated. We found that the size of **S** dropped off sharply for $c > 50$; at $c = 50$, the fraction of traffic in **M** that was contained in **S** was 91-99% for prefix-prefix matrices, and 80-88% for atom-atom and AS-atom matrices. Hence we used $c = 50$ in the experiments we report here.

Besides the condition of sufficient information in Ω, our matrices may violate the assumptions of matrix completion methods in other ways. Matrix elements tend to show high variability, which may violate the incoherence assumption; and in our case Ω is not uniformly sampled, but rather is determined by P's view.

6.3 Error Metrics

We measure two kinds of error: error in estimating individual invisible elements, and error in estimating the total amount of invisible traffic flowing through T.

We would like to estimate the elements that are visible in T and invisible in P. However, as discussed in Section 3.3,

we do not know the exact views corresponding to T or P, so we approximate the set of targets τ as:

$$\tau = \{(i,j)\,|\,M_{ij}^{(T)} \neq 0 \text{ and } M_{ij}^{(P)} = 0\}$$

To measure error in estimating individual elements of τ we use Normalized Mean Absolute Error (NMAE):

$$NMAE = \frac{\sum_{(i,j)\in\tau}|M_{ij}^{(T)} - \hat{M}_{ij}|}{\sum_{(i,j)\in\tau}M_{ij}^{(T)}}$$

where the $\hat{M}_{ij}$ are estimated using information only from P. To measure error in estimating the total invisible traffic, we use Absolute Relative Error (of the total):

$$RE = \frac{|\sum_{(i,j)\in\tau}\hat{M}_{ij} - \sum_{(i,j)\in\tau}M_{ij}^{(T)}|}{\sum_{(i,j)\in\tau}M_{ij}^{(T)}}$$

7. RESULTS

7.1 Estimating Individual Elements

The accuracy obtained in estimating individual elements is shown in Figure 7. For each of the 14 days, we compute NMAE of the estimates; each point on these figures represents the average over the 14 days. Where error bars are smaller than the within-class variation, they are omitted for clarity.

The figures illustrate a number of results. First, the estimation errors for the prefix-prefix indexing are much larger than for the others; most prefix-prefix results are off the scale in Figure 7(a). Prefix-prefix results for ridge regression and matrix completion were equally poor and are not shown.

Second, the regression methods are more effective in general than matrix completion. More study is needed to understand the reasons for this, but as noted earlier, our data may not meet the assumptions necessary for matrix completion to be successful.

Third, the plots show that regression methods show improvement with increased regularization (decreased rank or df of $\tilde{\mathbf{X}}$). This reflects the strong correlation in measured TMs. However, matrix completion does not show a significant dependence on the assumed rank r.
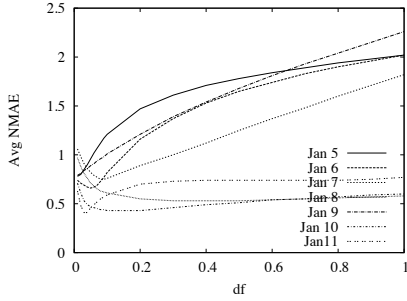
Figure 8: Cross-Validation Error, Ridge Regression

Figure 9: Example Distribution of Errors

Finally, it appears that the particular kind of regularization applied by ridge regression is the most effective kind for our data. For ridge regression, the magnitudes of NMAE found in the best cases are between 0.8 and 1 (with $0 < \text{df} < 0.2$). This means that average error in the estimates is less than the average magnitude of a TM element.

As mentioned earlier, proper selection of the tunable parameter is important to achieve highest accuracy. Figure 8 shows that cross-validation can be used for this task. We performed very simple cross validation as follows: for each target, after forming $\hat{\beta}$, we selected a single known, nonzero element from $\mathbf{M}^{(P)}$ and used $\hat{\beta}$ to estimate it. The NMAE of those estimates for the first seven days of our data under ridge regression is shown in Figure 8. The figure shows that cross-validation NMAE shows a minimum in the same range of df as we find the actual average minimum error (in Figure 7(b)). Notably, different days show different minimal values for df; these differences are averaged out in Figure 7(b). We note that this method of cross-validation is far from optimal (see, e.g., [9]), and improving it is a focus for future work; the example simply serves to show how the tuning parameter may be selected in practice.

7.2 Estimating Total Invisible Traffic

Given that TM elements range in size over many orders of magnitude, achieving an NMAE less than 1 is good. However, we would like to form more accurate estimates, so we examine the nature of the errors. For illustration purposes we focus on January 8 (day 4), and examine the errors in estimating values for packets at the Atom-Atom aggregation level under ridge regression (with df = 0.2).

A histogram showing the distribution of $M_{ij}^{(T)} - \hat{M}_{ij}$ around zero is shown in Figure 9. The histogram shows that the errors are more or less symmetric about zero. This suggests that the estimation process is relatively unbiased, so that the main contributor to NMAE is the variance of the estimator. This leads to the conclusion that summing estimates is likely to lead to improved accuracy, as errors cancel out.

Summing all the estimates for a single day leads to an estimate for the total amount of invisible traffic flowing through T. The relative error of this estimate is shown in Figure 10. The figure shows that the resulting accuracy can be quite
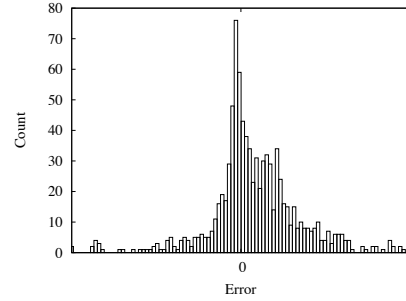
good; it is best for ridge regression. A more direct sense of the achievable accuracy can be obtained by looking at the comparison between daily estimates and actual values of the total traffic flowing through T. This is shown (for ridge regression) in Figure 11. The figure shows actual and estimated total invisible traffic, and for comparison purposes, the volume of traffic that flows through both P and T. This latter traffic constitutes the information used by P to estimate the total traffic through T. The figure shows that for both packets and bytes, whether using the atom-atom or AS-atom aggregations, measurements taken in P are sufficient to accurately estimate day-over-day traffic volumes in T.

Can finer-grained estimates be accurate as well? Here we recall our motivating example (Figure 1(a)). Can we estimate how much traffic T is exchanging with other ASes? Referring to Figure 2, note that T has two providers (other than P). We refer to these two providers as AS1 and AS2. In Figure 12 we show actual and estimated traffic flowing between T and these two providers (in packets). The figure shows that P can indeed form accurate estimates of the invisible traffic flowing between T and its other providers.

It is important to ask how these results are affected by the topological relationship between networks P and T. Of particular interest is the amount of information that P has about T's TM. In our data, for the atom-atom indexing, about 20.4% of the nonzero elements in $\mathbf{M}^{(T)}$ are nonzero (visible) in $\mathbf{M}^{(P)}$. That is, 20.4% of T's atom-atom flows also flow through P. One way to assess the importance of this overlap is by artificially reducing the information available to P. To do this, we chose a fraction p of elements at random from the set nonzero in both P and T, and made these unavailable to P for use in estimation. We varied p from 5% to 95%. The results are shown in Figure 13.

The Figure shows the average relative error in estimating the total traffic in T across all days (using ridge regression with df = 0.2 on byte counts with atom-atom indexing; error bars show one standard error). In the Figure, the x-axis shows the fraction of T's nonzero elements that P uses in estimation. This varies from about 20% down to about 1%. The figure shows that P can form a reasonably accurate estimate of T's total traffic even when P only has access to as little as 5% of the nonzero elements in T.
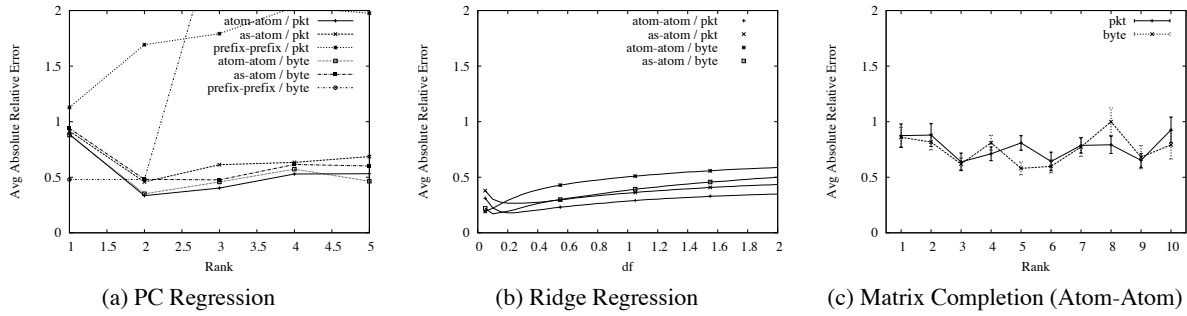
(a) PC Regression   (b) Ridge Regression   (c) Matrix Completion (Atom-Atom)

Figure 10: Relative Error in Estimating Total Invisible Traffic

(a) AS-Atom, Bytes, df = 0.05   (b) AS-Atom, Packets, df = 0.2

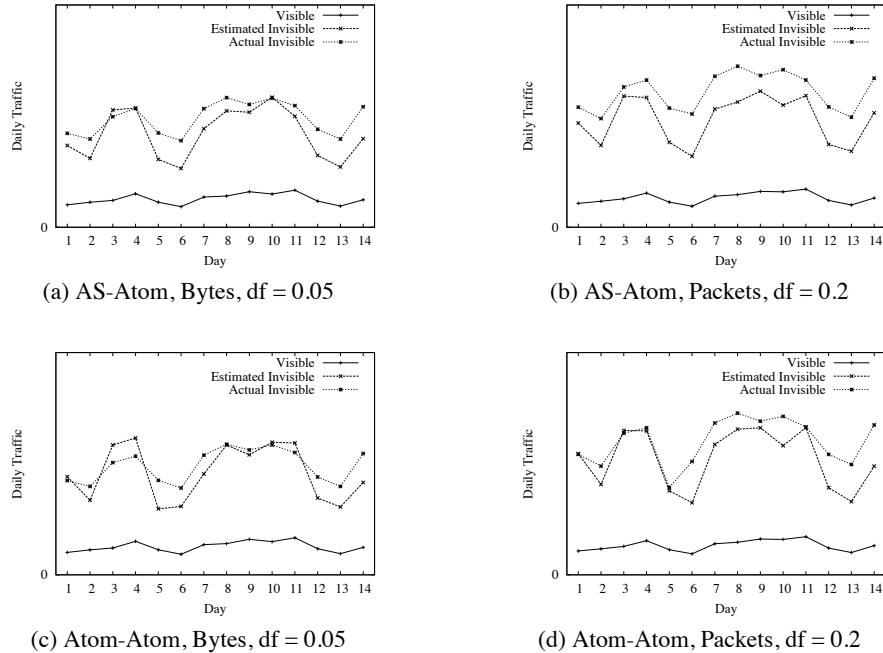(c) Atom-Atom, Bytes, df = 0.05   (d) Atom-Atom, Packets, df = 0.2

Figure 11: Estimated Daily Traffic Volume, Ridge Regression

7.3   Error Analysis

It is instructive to examine the cases in which traffic estimation fails. Figure 14 shows views of the complete distribution of errors for January 8 (same case as Figure 9). Whereas Figure 9 narrowed in on the errors around zero, Figure 14 shows all errors.

In Figure 14(a) there are four elements (not visible in the plot) that are so strongly negative as to offset the entire scale. Figure 14(b) shows the log-log complementary distribution plot of errors; it shows how clearly the four unusual elements stand out from the rest of the errors.

To understand the four unusual elements better, we examined the set of flows that were mapped to each element. Based on their traffic composition, we find that the four elements fall into two categories. The first category consists of three elements holding highly-local traffic. These consist of (1) the matrix element that represents T-T traffic (sourced

and destined within T); (2) traffic from T to O; and (3) traffic from O to T. Recall that T and O have a peering relationship, as shown in Figure 2.

The second category consists of the one remaining element. All of its traffic consists of UDP packets from a single source address (in network T) to a single destination address (in a geographically distant network) with destination port 53 – i.e., a DOS attack on a DNS server.

Looking at other days, we find that the same first three elements are typically very large, and typically underestimated. That is, we find that these three elements in the first category are consistently the source of the most significant errors. On some of the other days, there is also a single additional element (like our DOS attack) that introduces noticeable error.

It is not clear why such an unusually large amount of traffic flows between $T \leftrightarrow T$ or between $T \leftrightarrow O$. However, we note that ISPs often seek to keep traffic within their network boundaries, so the large $T \leftrightarrow T$ traffic may be a result of this.
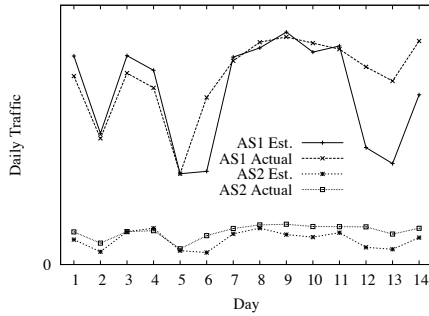
Figure 12: Estimated Daily Traffic Volume to Provider ASes 1 and 2.
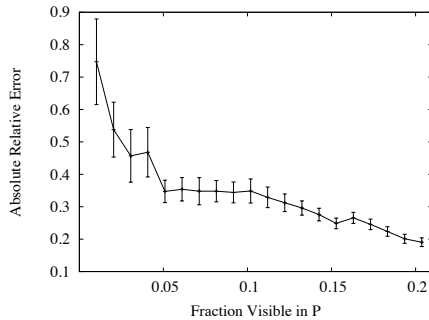

Figure 13: Effect of Reducing Overlap between $\mathbf{M}^{(T)}$ and $\mathbf{M}^{(P)}$.

Further, we note that T and O serve the same geographical region, and that the large volume of traffic passing between them seems to be dominated by distribution of large software packages via HTTP. Hence, it appears that T and O have a somewhat atypical relationship.

8. CONCLUSIONS AND FUTURE WORK

In summary, this paper has formalized the problem of inferring invisible traffic. We have defined and identified the need for atomic indexings, and pointed out a number of atomic indexings that are useful. We then presented evidence that the purely-spatial traffic matrices we measure show low effective rank. Building on this observation, we identified and evaluated a number of statistical methods for estimating invisible traffic. We found that regularized regression, applied on an element-by-element basis, shows good results. We also found that the fine granularity required by the need for atomic indexing is in tension with the aggregation needed to form accurate estimates.

This work has a number of limitations, which suggest the need for future study. A central limitation is that our measurements are taken from only two networks. This means, for example, that our observations of low effective rank of spatial TMs need replication in other settings. It also means that we cannot directly extrapolate to other network settings (for example, different topological relationships between P

and T) without further investigation.

Our work also suggests additional promising directions. There is opportunity to improve estimates (e.g., for cases like $T \leftrightarrow O$) through the introduction of additional side information, of the kind used in [5]. More refined statistical methods and better cross-validation could improve overall accuracy. The question of why matrix completion does not perform better is worth investigation as well. Finally, we note that the presence of errors caused by traffic anomalies suggest that purely-spatial methods may be a basis for anomaly detection.

Nonetheless, with respect to our motivating problem, our work shows that because of the low effective rank of our measured traffic matrices, accurate estimation of invisible traffic is possible. A network like P can form quite accurate estimates of the total traffic flowing through another network such as T, and even of the amount of invisible traffic that T exchanges with its providers. Our hope is that this result can lead to improved understanding of the spatial distribution of traffic in the Internet.

9. ACKNOWLEDGMENTS

10. REFERENCES

[1] Akamai Technologies, Inc. The state of the Internet. 4th Quarter, 2009.

[2] A. Broido and kc claffy. Analysis of RouteViews BGP data: policy atoms. In *Proceedings of Network Related Data Management Workshop (NRDM)*, Santa Barbara, CA, May 2001.

[3] CAIDA. Atom computation scripts. http://www.caida.org/funding/atomized_routing/documents/report1.xml, February 2003.

[4] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):712–772, 2009.

[5] H. Chang, S. Jamin, Z. Mao, and W. Willinger. An empirical approach to modeling inter-AS traffic matrices. In *Proceedings of IMC*, 2005.

[6] V. Erramilli, M. Crovella, and N. Taft. An independent-connection model for traffic matrices. In *Proceedings of the Internet Measurement Conference*, October 2006.
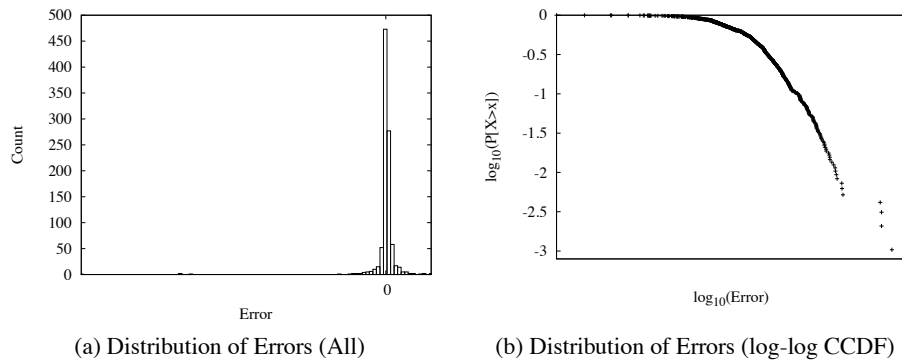
| (a) Distribution of Errors (All) | (b) Distribution of Errors (log-log CCDF) |

Figure 14: Error Analysis: Jan 8, Packets, Atom-Atom, Ridge Regression, df = 0.2.

[7] W. Fang and L. Peterson. Inter-AS traffic patterns and their implications. In *Proceedings of GLOBECOM*, 1999.

[8] A. Feldmann, N. Kammenhuber, O. Maennel, B. Maggs, R. De Prisco, and R. Sundaram. A methodology for estimating interdomain web traffic demand. In *Proceedings of IMC*, 2004.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2009.

[10] R. H. Keshavan, S. Oh, and A. Montanari. Matrix completion from a few entries. http://arxiv.org/abs/0901.3150, 2009.

[11] L. Kleinrock and W. Naylor. On the measured behavior of the ARPANetwork. In *AFIPS Conference Proceedings, National Computer Conference*, pages 767–780, May 1974.

[12] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. In *Proceedings of ACM SIGMETRICS*, pages 61–72, June 2004.

[13] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: Existing techniques and new directions. In *Proceedings of ACM SIGCOMM*, 2002.

[14] W. Mühlbauer, S. Uhlig, B. Fu, M. Meulle, and O. Maennel. In search for an appropriate granularity to model routing policies. In *Proceedings of ACM SIGCOMM*, 2007.

[15] University of Oregon. Route views project. http://www.routeviews.org/, July 2009.

[16] M. Roughan. First order characterization of Internet traffic matrices. Invited paper at the 55th Session of the International Statistics Institute, April 2005.

[17] J. Wallerich, H. Dreger, A. Feldmann, B. Krishnamurthy, and W. Willinger. A methodology for studying persistency aspects of Internet flows. *Computer Communication Review*, April 2005.

[18] Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. Technical report, Rice University, 2010.

[19] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. In *Proceedings of ACM SIGMETRICS*, 2003.

[20] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. Estimating point-to-point and point-to-multipoint traffic matrices: An information-theoretic approach. *IEEE/ACM Transactions on Networking*, 13(5):947960, 2005.

[21] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu. Spatio-temporal compressive sensing and Internet traffic matrices. In *Proceedings of ACM SIGCOMM*, 2009.