On Traffic Matrix Completion in the Internet

Gonca Gürsun
Department of Computer Science
Boston University
goncag@cs.bu.edu

Mark Crovella
Department of Computer Science
Boston University
crovella@cs.bu.edu

ABSTRACT

The ability of an ISP to infer traffic volumes that are not directly measurable can be useful for research, engineering, and business intelligence. Previous work has shown that *traffic matrix completion* is possible, but there is as yet no clear understanding of which ASes are likely to be able to perform TM completion, and which traffic flows can be inferred.

In this paper we investigate the relationship between the AS-level topology of the Internet and the ability of an individual AS to perform traffic matrix completion. We take a three-stage approach, starting from abstract analysis on idealized topologies, and then adding realistic routing and topologies, and finally incorporating realistic traffic on which we perform actual TM completion.

Our first set of results identifies which ASes are best-positioned to perform TM completion. We show, surprisingly, that for TM completion it does not help for an AS to have many peering links. Rather, the most important factor enabling an AS to perform TM completion is the number of direct customers it has. Our second set of results focuses on which flows can be inferred. We show that topologically close flows are easier to infer, and that flows passing through customers are particularly well suited for inference.

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: Network monitoring; C.2.5 [**Local and Wide-Area Networks**]: Internet — BGP

Keywords

Interdomain Routing, Matrix Completion

1. INTRODUCTION

Interdomain traffic – the traffic flowing between autonomous systems – is the fundamental workload of the Internet. It reflects global economic activity and information flow. Knowledge of interdomain traffic volumes is therefore of immense engineering, scientific and societal interest.

On a more local scale, knowledge of interdomain traffic volumes has great value for business intelligence. Consider an ISP that is pondering a bid for a competitor's customer. That ISP has a significant advantage if it knows how much business the competitor currently does with the customer (i.e., how much traffic they exchange), and how the customer's traffic would impact the ISP's network should the customer change providers.

Unfortunately, broad knowledge of interdomain traffic volumes on the Internet is hard to come by. The inherently distributed architecture of the AS-level Internet means that there is no single place where all Internet-wide traffic can be measured, and the competitive relationship of the commercial Internet means that sharing such information across organizational boundaries is unlikely. The authors in [10] review the situation and note that an inter-AS traffic matrix is an "elusive object."

Hence we are prompted to turn to statistical inference where direct measurement is impossible. The problem can be cast in terms of a *traffic matrix* – measurements of traffic volume from sources (rows) to destinations (columns). Any given AS can observe some of the elements of this matrix – namely, exactly the traffic that flows through the AS. Can an AS ever 'fill in' the missing entries (corresponding to traffic *not* flowing through the AS) thereby 'completing' the matrix? Doing so would give the AS a view of a much larger set of traffic volumes, or even of traffic volumes across the entire Internet.

Surprisingly, recent work has suggested that in some cases, a single AS *can* complete at least some of the missing portions of its traffic matrix [2, 27]. The general idea (described in detail in Section 2) is as follows. The first step is to note that traffic matrix elements show strong statistical regularities. There are predictable relationships between elements, such that missing elements can often be cast in terms of linear functions of observable elements. One way of describing this phenomenon is to note that traffic matrices often have *low effective rank* (which we define in Section 2). The second step is to apply methods of statistical inference that are recently emerging in the signal processing community, termed *matrix completion*. These methods are specifically designed to perform missing-element inference on matrices that have low effective rank. A wide variety of such methods have now been developed [3, 4, 5, 17, 25].

The key to matrix completion is the ability to observe a sufficiently useful subset of the matrix entries. If enough en-

IMC'12, November 14–16, 2012, Boston, Massachusetts, USA.

tries, in the right positions, can be observed, the rest of the entries can be 'filled in.' In the context of interdomain traffic matrices, this question relates to the network's AS-level connectivity and routing patterns. Some ASes, by virtue of their topological position and commercial roles, may observe enough traffic passing through their networks to allow them to infer traffic volumes *not* passing through their networks. Initial studies have shown existence proofs that such inference is possible [2, 27]. The question is – for whom? For which ASes is traffic matrix completion most likely to be successful? And, for those ASes that can infer some TM elements, which elements can they infer?

Those questions are the focus of this paper. We seek to understand *which* ASes are likely to be able to perform TM completion, *which* elements they can infer, and *why*. We seek to answer these questions from two standpoints: from an analytical standpoint, we look for graph-theoretic properties of the AS topology that lead to increased traffic inference ability for an AS. And from a practical standpoint, we look to answer to these questions in terms of metrics that relate to an ISP's business and engineering relationships – e.g., how many customers and peering links it has.

To do so, we provide a framework for analyzing the inference capability of a given AS based on its position in the AS graph and the set of paths that pass through it. This is the first contribution of our paper, and is independently useful, for example, when an individual AS seeks to evaluate its own inference capability. However, once having developed this framework, our second contribution is to apply the framework to a large AS graph to investigate actual ASes and their TMs.

The first stage of our work explores the relationship between TM completion ability and certain idealized graph models. We develop an algorithm that allows us to prove a lower bound on TM completion ability, and using it we gain insight into how TM completion ability relates to local graph topology.

The second stage of our work brings realistic routing into the picture. For this we rely on an extensive survey of the AS-level Internet, comprising over 100 million AS paths, captured at a single time. This rich dataset allows us to explore how TM completion ability varies over the set of all ASes in the Internet.

Finally, the third stage of our work applies actual matrix completion to realistic TMs across ASes in the Internet. For each AS we evaluate its accuracy when completing a TM comprising about 30 million elements, of which between 0.001% and 0.3% are actually visible to the AS, depending on the set of AS paths that flow through it.

The three stages of our effort mutually support our primary conclusions. We find that the key to TM inference ability lies in the *set of customers* of an AS. Our analysis and measurements show that an AS's customers provide the AS with crucial knowledge of interdomain traffic flows needed for TM completion. When asking which flows are most readily estimated, we find that the closer a flow passes to an AS in the BGP graph, the more readily it may be estimated; and when an AS seeks to specifically recover the entries of flows that pass through another AS, it is most successful when the other AS is a neighbor – especially, when the other AS is a customer.

2. BACKGROUND & RELATED WORK

2.1 Definitions

A *traffic matrix* (TM) is an $m \times n$ matrix T in which T_{ij} is a measure of the traffic flowing from a set of IP addresses $\mathscr{S}_i$ to a set of IP addresses $\mathscr{D}_j$ during a specific time interval. At any moment, the *view* of a network P consists of all the source-destination pairs (s,d) such that any traffic flowing from s to d will at some point pass through P. A network's view can be captured in the form of an $m \times n$ *visibility matrix* M, where $M_{ij} = 1$ if traffic from $\mathscr{S}_i$ to $\mathscr{D}_j$ passes through P, and zero otherwise.

A key property for traffic matrices in our work is *low effective rank*. If an $m \times n$ matrix T can be factored into an $m \times d$ matrix X and a $d \times n$ matrix Y, such that $XY = T$, then T has rank (no greater than) d. If $d \ll \min(m,n)$ then we say T has *low rank*. When working with measurement data, a matrix T may be strictly speaking full rank, but nonetheless well-approximated by a low-rank matrix. That is, if there exists a rank d matrix T' such that $T \approx T'$, we say that T has low effective rank. For example, we may use a least-squares criterion: $T \approx T'$ if $\sum_{i,j}(T_{ij} - T'_{ij})^2 / \sum_{i,j} T_{ij}^2$ is small enough. Low rank is important because it means that elements of T are related; only a small amount of information (X and Y) is needed to construct T, so some elements of T can be computed as linear functions of other elements. Likewise, if a matrix has low effective rank, then some elements can be approximated as linear functions of other elements.

In this work, TMs will be organized as either node-to-node TMs (in our idealized examples in Section 3) or AS-to-prefix TMs (when using real topologies in Sections 4 and 5).

2.2 Properties of Traffic Matrices

Our work deals with large-scale inference of traffic matrices that span ASes. While an interdomain TM remains an "elusive object" [10], a few previous studies have built models of interdomain traffic. The work described in [13] estimates Web-related interdomain traffic, using server logs from a large CDN provider. The work described in [8] brings more AS-specific information to the table, including business relationship, population size, and AS role, and fuses this information to form estimates of interdomain traffic volume. These models and methods inform our work, but the focus of our work is not explicitly on modeling TMs. Rather, we only assume that TMs show low effective rank.

Indeed, there is considerable evidence that traffic matrices often show low effective rank. In [19], the authors document low effective rank in measurements of temporal traffic matrices, in which each column is a time-series of the traffic volume between a source-destination pair. In [2], the authors present a similar result for measurements of spatial traffic matrices, in which the rows represent the sources and the columns represent the destinations (as do the matrices in this paper). More generally, traffic matrix modeling often assumes that TMs have low effective rank. The often-used *gravity* models are rank-1 models; such models have been used, for example, in [9, 20, 22, 23, 26]. Likewise, the authors in [12] show that a rank-2 model is a good fit to measured TMs. Finally, a number of papers have explicitly relied on the property of low effective rank in TMs as the basis for their results [2, 18, 27].

In this paper we start from the assumption that TMs show

low effective rank. However, we do not assume that TMs have any *particular* effective rank; our analyses and experiments treat matrix rank k as a parameter.

2.3 Traffic Matrix Completion

Our paper applies ideas from matrix completion to traffic at the AS level. Matrix completion is a relatively new area in statistical inference with a number of recent results [5, 17]. The matrix completion problem consists of recovering a low-rank matrix from a subset of its entries. Let the $m \times n$ matrix T having rank $k \ll min(m,n)$ be unknown, except for a subset of its entries Ω which are known. If the set Ω contains enough information, and T meets a condition called *incoherence,* then there is a unique rank-k matrix that is consistent with the observed entries.

Recently, a variety of algorithms have been proposed that solve the matrix completion problem under various assumptions [3, 4, 5, 17, 25]. These algorithms are typically analyzed under the assumption that the locations of the known entries of T are distributed uniformly at random across the matrix. However, matrix completion can be possible when the location of entries are not uniformly spread across the matrix. In particular, the algorithm in [21] does not assume uniformly spread entries, and furthermore has a more general capability. Rather than focusing exclusively on matrix completion, it can also be used to identify *which* elements of a matrix can be recovered, even when full completion is not possible. It is this property of the algorithm that we make use of in our work. We review this algorithm and our use of it in the next subsection.

Given the tendency for traffic matrices to show low effective rank, a number of authors have applied matrix completion to different types of TMs. In particular, the authors in [27] develop algorithms for accurately recovering missing values (due to measurement failures) in intra domain TMs in which the sources and destinations are in the observer's network. And in the study mentioned previously, the authors in [2] develop methods for inferring traffic volumes for traffic that does not pass through the observer's network, and hence cannot be measured. In [2], the authors show that a network P can infer the traffic that does not flow through P but flows through its direct customer network T. However [2] only demonstrates this for one particular pair of networks and does not give insight into when TM completion is possible in general. In contrast, our paper asks the broader question - what relationship should P and T have in order for TM completion to be successful.

2.4 ICMC and AICMC

To analyze the ability of an AS to perform matrix completion, we adopt a particular algorithm from the matrix completion literature called *Information Cascading Matrix Completion (ICMC)* [21]. ICMC can be applied to matrices that are exactly low-rank, or approximately low-rank; for simplicity in the description below we describe it as applied to an exactly low-rank matrix. However extensions to deal with approximately low-rank matrices are not difficult, as described in [21].

We use ICMC as a tool for exploring the TM completion ability of ASes. The advantage of using ICMC as compared to other matrix completion algorithms is that it identifies which matrix elements can definitely be recovered in a given setting. That is how we use it in this Section and Sections 4 and 5. However, not all matrix completion algorithms work in this elementwise, all-or-nothing fashion; other algorithms try to form estimates of all missing elements. Hence we confirm our results by using a different matrix completion algorithm in Section 6.

ICMC assumes that the $m \times n$ matrix T having rank k is *non-degenerate*, meaning that T can be factored into the matrices $X \in \mathbb{R}^{m \times k}$ and $Y \in \mathbb{R}^{k \times n}$ such that any k rows of X are linearly independent, any k columns of Y are linearly independent, and $XY = T$. The basic idea of ICMC is to progressively compute rows of X and columns of Y so that $(XY)_{ij} = T_{ij}, \forall (i,j) \in \Omega$.

In fact, our goal in this paper is not performing matrix completion per se, but rather identifying *whether* and *when* matrix completion is possible. Hence we employ ICMC in a manner we refer to as *abstract* ICMC, or *AICMC*.

AICMC may be expressed in terms of operations on a bipartite graph, as shown in Figure 1. The graph consists of two sets of vertices, $U = \{u_i, i = 1, \ldots, m\}$ and $V = \{v_j, j = 1, \ldots n\}$. An edge exists between u_i and v_j if $(i,j) \in \Omega$; otherwise no edge exists. Thus there is a correspondence between vertex u_i and row i of X; and there is a correspondence between vertex v_j and column j of Y.

AICMC progresses by successively marking vertices as 'infected,' which means that the corresponding row of X or column of Y can be recovered. The set L consists of infected u vertices, and R consists of infected v vertices. Infection propagates through the graph: v_j can be infected if there are at least k edges from v_j to vertices in L. Analogously, infecting u_i requires at least k edges from u_i to vertices in R. When no more nodes can be infected, the set L identifies the rows of X that can be recovered, and R identifies the recoverable columns of Y. The authors in [21] prove the correctness of this process for recovering X and Y.

Figure 1 shows an example visibility matrix and corresponding bipartite graph. This process is shown in the figure for $k = 1$. Starting with infected vertex u_1, each step progressively infects nodes on alternating sides of the bipartite graph. While in this case the final set of infected nodes corresponds to the largest connected component, note that for $k > 1$ the final set of infected nodes is not necessarily the largest connected component.

To start the algorithm, one notes that the solution X, Y is not unique, and hence without loss of generality the algorithm can be initiated by setting any k rows of X to the $k \times k$ identity matrix, and marking the corresponding k vertices as infected (forming the initial population of the set L). Beginning from this initial set of infected nodes, the algorithm proceeds by alternately adding to the sets R and L. When these sets contain all vertices in the graph, the entire matrix is recovered at rank k.

That said, one can set aside the graph interpretation and express AICMC simply in terms of an observer's visibility matrix M. Note that $M_{ij} = 1$ iff $(i,j) \in \Omega$. AICMC proceeds as follows

1. Choose k rows of M and set L to those rows.
2. If L contains all rows of M and R contains all columns of M, stop - the matrix T can be fully recovered. Otherwise:
   (a) For every column of M such that there are at least k 1s in rows from set L, add the column to R. If there are no such columns, stop.

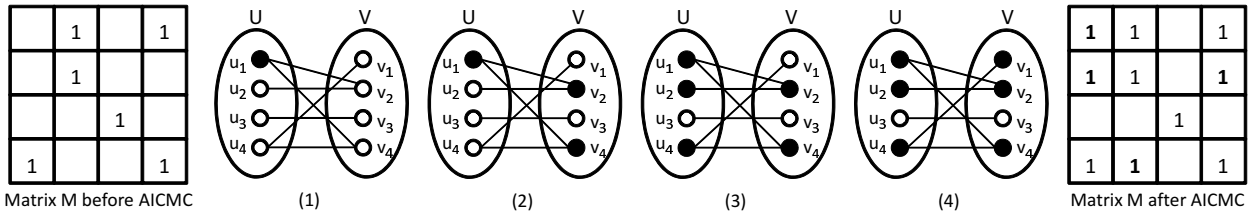

Figure 1: AICMC Example: T is a 4×4 data matrix (not shown) of rank $k = 1$. T's known elements correspond to the positions of the 1s in its visibility matrix M (left side of the figure). The steps of AICMC are: (1) $L = \{u_1\}$, (2) $R = \{v_2, v_4\}$, (3) $L = \{u_1, u_2, u_4\}$, (4) $R = \{v_1, v_2, v_4\}$. The algorithm stops at the end of (4). The completed elements are $(1,1)$, $(2,1)$, $(2,4)$, and $(4,2)$ - 1s shown in bold in M (right side of the figure).

(b) For every row of M such that there are at least k 1s in columns from set R, add the row to L. If there are no such rows, stop.

3. Go to 2.

At completion, an element (i, j) can be recovered if row i is in L and column j is in R. Thus AICMC allows us to examine an AS's visibility matrix, and identify, for each invisible element, whether it can be recovered at a given rank (or approximate rank) k. In Figure 1 the recoverable elements are shown on the right side of the figure. Note that if an AS can complete its TM at rank k, it can complete it at any rank $r \leq k$.

2.5 Interdomain Topology

A central aspect of our work is establishing a connection between the AS-level topology of the Internet, and the ability of individual ASes to do traffic matrix completion. Hence we rely on the considerable body of work that has characterized the AS level topology, of which we can only review a portion here.

At the highest level, the AS graph is usually characterized as having roughly three distinguishable parts [6, 15, 16, 24]. Forming the center of the graph is a mesh-like core that is a clique or 'almost' a clique. This core is fed by a collection of ASes in provider-customer relationships that are 'tree-like' but not strictly trees. Finally the vast majority of ASes are *stubs,* ASes at the edge of the network having no customers themselves. A number of methods have been proposed for organizing ASes into a small number of *tiers* [15, 24].

In our work we seek a finer-grained and less arbitrary measure of centrality in the AS graph than tiers, and so we turn to a tool for graph analysis called *k-core decomposition* [1] [1]. K-core decomposition separates the vertices of a graph into successive sets called "shells". These are operationally defined: the 1-shell consists of all nodes of degree 1, plus all nodes that become degree-1 when degree-1 nodes are removed. Removing all such nodes leaves only nodes of degree-2 and higher, and the process repeats. As described in [6], this is a parameter-free way of characterizing the AS graph, and it naturally identifies a 'nucleus' (innermost shell) of the graph which is observed to consist of major provider ISPs, major IXPs, CDNs and content providers. In our data the nucleus is shell 58, containing 120 ASes. Each node in the nucleus is connected to about 70% of the other nodes.

[1]Note that parameter k used in k-core decomposition represents degree order It is unrelated to rank parameter k we use throughout the paper.

Our knowledge of the AS level graph is derived from measurements, and is generally understood to be imperfect. A good review of the issues is presented in [10], but a persistent concern is that maps of the AS graph miss links, in particular peering links [7]. Missing links may result in some inaccuracy in certain graph metrics we use: k-core decomposition, degree, number of peers and number of customers. For that reason we do not base results on precise values of these metrics, but rather focus on the trends seen as these metrics vary. However missing links do *not* cause inaccuracies for our key metrics: completion ability and expected rank (defined below). This is because (as explained in Section 4.1) we select a subset of all AS paths in such a way that these metrics are known with high confidence.

Finally, a portion of our results relies on the classification of AS-AS links as customer-provider or peer-peer (we do not consider sibling-sibling links). For this we rely on the body of knowledge that has been built up on how to do this classification since [14], and in particular rely on the comprehensive approach used in [11].

3. ANALYSIS

Our first step is to develop high-level insight about the relationship between graph topologies and the opportunity for traffic inference. We do that by establishing provable lower bounds on traffic matrix completion in various idealized networks. These models necessarily ignore important aspects of the AS level Internet (e.g., they assume shortest-path, symmetric routing) but our goal here is to build intuition. Later, in Sections 4 and 5, we will examine real AS level graphs.

Each of our idealized models starts with a particular graph $G = (V, E)$, with $|V| = n$. Each node $v_i \in V$ sends one traffic flow to every node $v_j \in V$ (including v_i itself). All flows travel over shortest paths, assuming edges have unit weight. In each graph we designate an *observer node*, denoted v_o; we will analyze the observer's ability to do traffic inference.

The information available to the observer node is summarized in a *visibility matrix* M of size $n \times n$. We set $M_{ij} = 1$ if the flow from v_i to v_j passes through v_o and so is measurable by v_o; otherwise we set $M_{ij} = 0$. By convention we assign v_o to matrix index 1. Thus the first row and the first column of M are always fully populated with 1s, since all traffic that originates or terminates at v_o is visible to v_o. Furthermore, because of our assumptions about flow routing, M is symmetric.

To find a lower bound on the traffic inference capabilities of v_o, we apply Abstract ICMC (AICMC) to M as described in

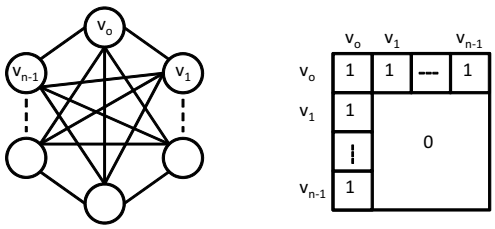

Figure 2: Full mesh network.

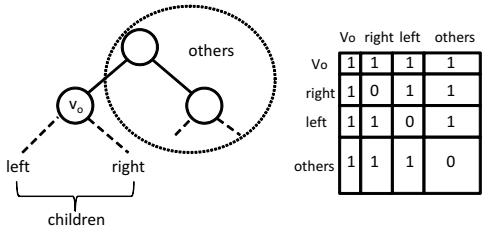

Figure 3: Example tree.

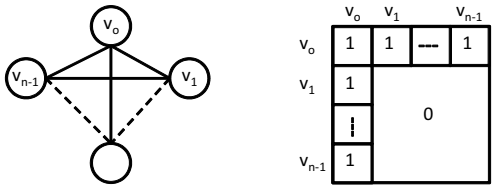

Figure 4: Mesh-of-Trees

Section 2.4. Using AICMC we can identify invisible elements of the network-wide traffic matrix that can be recovered by v_o, assuming the traffic matrix is rank k. For simplicity, we ask the following question in each case: For what values of k can v_o recover the entire TM? Larger values of k imply a greater ability to do TM completion.

We study a progression of idealized networks, starting with highly decentralized networks, then moving to trees and tree-like networks, and finally considering some more specialized topologies that are inspired by the connection pattern of ASes in the Internet.

3.1 Idealized Networks

We study three idealized networks: a clique, and two trees that differ in terms of size and degree.

Clique: In a clique (a full mesh), there is a direct link between every pair of nodes (Figure 2). As a result, the observer node v_o can only measure flows having itself as either source or destination, resulting in the visibility matrix shown in the Figure.

PROPOSITION 3.1. *Given a full mesh with n nodes, (a) the observer can complete its TM for $k = 1$; and (b) the observer cannot complete its TM at any rank $k > 1$.*

PROOF. For (a): in the initial step we choose the first row of M and set L to that row, i.e. $L = \{1\}$. Next, all columns are added to R, i.e. $R = \{1,\ldots,n\}$, since they all have 1s in the first row of M. Finally, all the rest of the rows are added to L, i.e. $L = \{1,\ldots,n\}$, since they all have 1s in the first column of M.

For (b): when $k > 1$, the initial step chooses k rows of M and sets L to those rows. However, no choice of k rows yields more than one column with k 1s, so completion is impossible at rank $k > 1$. □

Trees: Figure 3 shows an example tree and the visibility matrix of an arbitrary observer node, v_o. Node v_o has two children, which form the roots of its *left* and *right* subtrees (extension to the case where v_o has more than two children is straightforward). Nodes besides v_o and its children are referred to as *others*. The visibility matrix reflects the fact that the observer can measure traffic between nodes in its subtrees and others, and traffic between nodes in its right and left subtrees. The observer cannot measure traffic flowing only within the right subtree, or within the left subtree, or among others.

PROPOSITION 3.2. *Given a tree containing an observer node v_o with (at least) two children, as in Figure 3, let n_r be the number of nodes in the right subtree, n_l be the number of nodes in the left subtree, and n_o be the number of other nodes. If $n_r \geq k-1$, $n_l \geq k-1$, and $n_o \geq k-1$, the observer can complete its TM at rank k.*

PROOF. Let $\mathcal{N}_l$ be the indices of the left children, $\mathcal{N}_r$ be the indices of the right children, and $\mathcal{N}_o$ be the indices of the other nodes. The initial step chooses the first k rows of M and sets L to those rows, i.e. $L = \{1,\ldots,k\}$. Next, the columns that correspond to the indices of the observer, left children, and the others are added to R, i.e. $R = \{1, \mathcal{N}_l, \mathcal{N}_o\}$, since they have at least k 1s in the rows of L due to the assumption $n_r \geq k-1$. Next, all the remaining rows are added to L, i.e. $L = \{1,\ldots,n\}$, due to the assumptions $n_l \geq k-1$ and $n_o \geq k-1$. Finally, the columns that correspond to right children are are added to R, i.e. $R = \{1,\ldots,n\}$ since they have at least $3k-2$ 1s in the rows of L. □

Note that the proposition does *not* hold if the observer has only one child; observation of traffic *between* children is important for overall traffic matrix completion.

The previous proposition showed that the number of customers in each subtree matters. Next, we show that local connectivity (node degree) matters as well.

PROPOSITION 3.3. *Given a tree (a star) consisting of an observer node v_o connected to d other individual nodes, the observer can complete its TM at rank k, where $2k \leq d+1$.*

PROOF. In this topology, the observer v_o sees the traffic between any pair of nodes. This results in a visibility matrix M in which all elements are 1s except the last $n-1$ elements in the diagonal. In the initial step we choose the first k rows of M and set L to those rows, i.e. $L = \{1,\ldots,k\}$. Next, the columns that correspond to the indices greater than k are added to R, i.e. $R = \{1, k+1, \ldots, n\}$ since they have k 1s in each of the rows of L. Next, all the remaining rows are added to L, i.e. $L = \{1,\ldots,n\}$, since they have at least k 1s due to the assumption $2k \leq d+1$. Finally, all the remaining columns are added to R, i.e. $R = \{1,\ldots,n\}$ since they have at least $2k-1$ 1s in the rows of L. □

Thus there are two node characteristics that influence the ability to complete the TM in a tree: the observer can complete

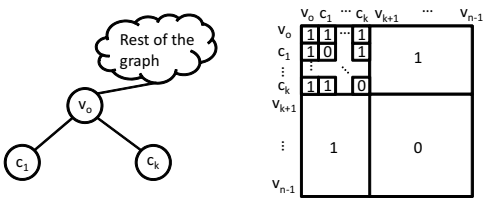

Figure 5: A node with k single-parent customers

its TM if the number of nodes in each of its subtrees is high enough, or if its degree is high enough.

3.2 Internet-Like Graphs

Now we turn to graph models that are intended to capture aspects of the Internet topology at the AS level. We apply the idealized graph models studied above to various Internet-inspired topologies. Again, these models ignore important aspects of the AS level Internet, but we build some intuition about the AS level Internet by studying them.

Our first model is a full mesh of nodes, each of which is the root of a subtree, as illustrated in Figure 4. This model is intended to capture some aspects of the relationship between top-tier ASes, as described in Section 2.5.

PROPOSITION 3.4. *Given a mesh of trees in which each mesh node v_i is the root of a tree, the observer mesh node v_o has at least two child trees each of size at least $n_c \geq k-1$, and the sum of the sizes of all other trees (including the roots) is $\geq k-1$, the observer node v_o can fully complete its visibility matrix at rank k.*

PROOF. Straightforward adaptation of Proposition 3.2. □

This example shows that even though a node participates in a decentralized mesh (as for example happens at the top of the AS hierarchy), if it has enough nodes in its subtrees it can complete its traffic matrix.

Single-Parent Stub Customers: Next we turn to analyze models of AS topologies that are more typical further down in the AS hierarchy. We define *single-parent customers* as nodes that use only one provider to connect to the rest of the network during the time interval in which measurements are taken. Note that single-parent customers are not necessarily single-homed customers — they may have multiple providers, but they only route traffic through one provider at any given time.

A node with single-parent customers can see the traffic between these customers and the rest of the network. Figure 5 (left) shows an observer v_o that has k single-parent customers, $c_1, \ldots, c_k$, which are stub networks. Figure 5 (right) shows the visibility matrix of v_o.

PROPOSITION 3.5. *Given a network of size n, an observer v_o that has k single-parent stub customers can complete its visibility matrix M at rank k, where $n \geq 2k+1$.*

PROOF. In the initial step we choose the first k rows of M and set L to those rows, i.e. $L = \{1, \ldots, k\}$. Next, the columns that correspond to the indices $v_o, c_k, v_{k+1}, \ldots, v_n$, are added to R, i.e. $R = \{1, k+1, \ldots, n\}$. Next, the row that corresponds to c_k is added to L, i.e. $L = \{1, \ldots, k+1\}$. Next, the columns that correspond to $c_1, \ldots, c_{k-1}$ are added to R, i.e. $R = \{1, \ldots, n\}$. Finally, the rows that correspond to $v_{k+1}, \ldots, v_n$ are are added to L, so that $L = \{1, \ldots, n\}$. Note that this is a simple extension of Proposition 3.3. □

An important loss of visibility occurs when some customers have peering relationships. If two customers c_i and c_j have a peering relationship, v_o can not see the traffic between them. This yields a visibility matrix like Figure 5, but with two more 0 entries on the upper left submatrix. In general, this type of peering relationship can happen between more than one pair of customers. In the worst case, all customers have peering relationships and this makes the upper left part of M all 0s except for its first row and column.

PROPOSITION 3.6. *Given a network of size n, for an observer v_o that has k single-parent stub customers, if at least $k-p-1$ of its customers have no more than p peering links with other customers, where $p \geq 0$ and $n \geq 2k+1$, then v_o can complete its TM at rank $r = k-p$.*

PROOF. Assume that the customers are indexed (starting from 2) in order of increasing number of peering links. In the initial step we choose the first r rows of M and set L to those rows, i.e. $L = \{1, \ldots, r\}$. Next, the columns that correspond to v_o, v_{k+1}, …, v_n have r 1s in the rows of L. The columns that correspond to customers are not guaranteed to have r 1s; it depends on the number of peering links they have. Therefore, $R = \{1, k+1, \ldots, n\}$. Next, the rest of the rows that correspond to the customers are added to L, i.e. $L = \{1, \ldots, k+1\}$ due to the assumption that $n > 2k+1$. After this point, the columns that are not added to R yet are $\{2, \ldots, k+1\}$. These correspond to the customers c_1, …, c_k. Likewise, the rows that are not added to L are v_{k+1}, …, v_k. For these rows to be added, at least $r-1$ columns that correspond to the customers should have at least r 1s. Rewriting this statement for $r = k-p$, to complete the matrix at rank $k-p$, at least $k-p-1$ columns that correspond to the customers should have at least $k-p$ 1s. If an AS has p peering links, then it has $k-p$ 1s in its corresponding column. This shows that in order to complete the matrix at rank $k-p$, at least $k-p-1$ customers should have no more than p peering links with other customers. □

This shows that the presence of a limited amount of peering links diminishes, but does not necessarily destroy, the observer's ability to complete its TM.

Single-Parent Customer Trees: Next, we consider non-stub single-parent customers. We refer to the set of all single-parent descendants of the observer as its *Single-Parent Customer (SPC) Tree*. Figure 6 shows a SPC tree example. In this example, v_o cannot observe traffic between c_1 - c_2, c_1 - c_3, c_2 - c_3, or c_4 - c_5. Note that this creates the same visibility matrix as the case where all nodes $c_1, \ldots, c_5$ are stubs, but with peering relationships between the pairs (c_1, c_2), (c_2, c_3), (c_1, c_3), and (c_4, c_5).

PROPOSITION 3.7. *Given a node v_o and its SPC tree, any subtree of v_o which consists of d nodes creates the same visibility matrix as the case where the nodes are stubs, and there are peering links between each pair of nodes of the subtree.*

PROOF. Clear by construction. □

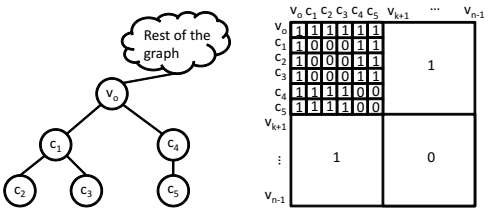

Figure 6: Single-Parent customer tree example.

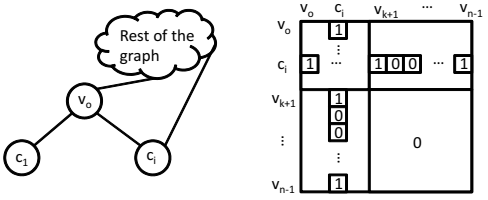

Figure 7: Multi-Parent customers.

Thus a subtree of size d has the same completion effect as d stubs, each having $d-1$ peering links. Thus, given an observer v_o and its SPC tree, proposition 3.7 and 3.6 can be used together to determine matrix completion ability at any rank. For example, the network in Figure 6 is equivalent to one having five stub customers, each having no more than 2 peering links, and so v_o can complete its matrix at rank 3.

PROPOSITION 3.8. *Consider a network of size n, and a node v_o which has a SPC tree that consists of m subtrees of sizes $d_1, \ldots, d_m$. Let k be the total number of customers in this SPC tree s.t. $d_1 + \cdots + d_m = k$ and $2k+1 \leq n$. Let the size of some subtrees be smaller than $p+1$, i.e., $d_1, \ldots, d_i \leq p+1$, where $p \geq 0$. For v_o to complete its visibility matrix M of rank $r = k-p$, it must be true that $d_1 + \cdots + d_i \geq k-p-1$.*

PROOF. Follows from Propositions 3.6 and 3.7. □

Multi-Parent Customers: Finally, we consider the influence of multi-parenting on the ability of a node to do TM completion. We define *multi-parent customers* as nodes that use multiple providers to connect to the rest of the network during the time interval in which measurements are taken. Note that all multi-parent customers are multi-homed customers.

Assume that the observer v_o has a multi-parent customer c_i. Customer c_i exchanges traffic with some nodes in the rest of the graph, as well as the other customers of v_o, through v_o. However, c_i also exchanges traffic with some other nodes in the rest of the graph through other providers. The example graph in Figure 7 yields the visibility matrix shown in the Figure.

PROPOSITION 3.9. *Given a network of size n, an AS v_o, which has $c+1$ multi-parent customers out of k customers, is guaranteed to fully complete its visibility matrix M at rank $k-c$, where $c \geq 0$ and $n > 2k+1$.*

PROOF. Assume that the customers are indexed (starting from 2) in the order of decreasing number of 1s in their rows. In the initial step we choose the first $k-c$ rows of M and set L to those rows. Due to the assumption that $n > 2k+1$, the densest rows correspond to $v_o, c_1, \ldots, c_{r-1}$, i.e. $L = \{1, \ldots, r\}$. Next, if none of the customers were multi-homed, the columns that correspond to $v_o, v_{k+1}, \ldots, v_n$ since they would have r 1s. However, if some of the customers are multi-homed, then the columns that correspond to the ASes which they send/receive traffic to/from through other providers may have less than r 1s. For instance, consider an AS v_j which all $c+1$ multi-homed customers send/receive traffic to/from through other providers, then its corresponding column has $k-c$ 1s. This implies that completion at ranks higher than $k-c$ is not guaranteed for v_o. □

Thus, Proposition 3.9 can not provide a guarantee that a multi-parent customer can improve an node's TM completion ability. However, in practice there are a number of ways in which the flows sent by the multi-parent node through the observer may contribute to TM completion ability. First, they may nonetheless provide sufficient visibility to improve TM completion, since the Proposition only establishes a lower bound on ICMC's performance; and second, the additional visibility may be useful when using inference methods other than ICMC.

In summary, the examples in this section have provided a number of insights into the relationship between graph topology and TM completion ability. First, we find that the decentralized nature of meshes is a strong impediment to TM completion. On the other hand, tree structures can be suitable for TM completion, and two aspects of a tree are important: increasing the degree of the observer node and increasing the number of nodes in each subtree both tend to improve TM completion.

Applying these models to Internet-like topologies, Proposition 3.4 suggests that despite its mesh-like nature, the topological relationship of top-tier ASes is amenable to TM completion. For ASes further down in the AS hierarchy, Proposition 3.5 shows the value of having single-parent customers, while Proposition 3.6 shows that peering relationships between one's customers are detrimental, but only in a limited way, to TM completion. Propositions 3.7 and 3.8 show that when one's customers themselves are providers, nodes deeper in the tree contribute more limited information for TM completion.

Taken together, Propositions 3.5 and 3.8 show that it is good to have a large single-parent customer tree, and it is better for those nodes to be arranged in a wide tree rather than a deep tree. For example, we can compare two organizations of a SPC tree, as shown in Figure 8. Consider the case when an AS v_o has k ASes in its SPC tree (and assume the network as a whole is large enough). When all of v_o's descendants are its direct customers (a), it can complete its TM at rank k. In comparison, when only two of v_o's descendants are direct customers (b), that is, its customers are grouped in two subtrees each of size $k/2$, it can only complete at rank at most $k/2$.

4. WHICH ASES CAN DO TM COMPLETION?

The analyses in the previous section provide some insight regarding the best conditions for TM completion, but they have a number of drawbacks. First, although the previous analyses give some indication of what conditions are best to allow an AS to perform TM completion, it is not clear *where* in the Internet those conditions are most prevalent. Second, the analyses assume highly idealized network models, which differ sig-

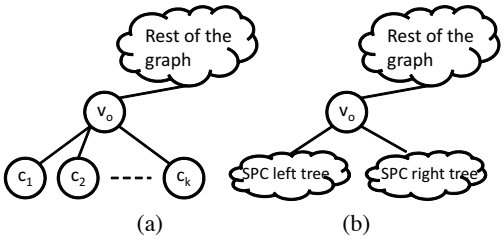

Figure 8: Comparison of two SPC trees.

nificantly from the actual AS topology. For example, the analyses assume there is a single source and destination in each node, and that routing is shortest-path and symmetric. These assumptions are all invalid in the AS-level Internet.

In this section we evaluate the ability of real ASes to do TM completion. Our goal in doing so is twofold: first, we seek to verify that the positive results from Section 3 hold in practice – namely, that TM completion is possible, at least for certain rank matrices, in the real Internet. Second, we seek to answer a set of natural follow-on questions. In particular, we would like to know: (1) Given that the analyses in the previous section suggested that TM completion may be possible at different 'locations' in the Internet (ie, among top-tier ISPs as well as ISPs lower in the AS hierarchy), where in fact is the opportunity for TM completion greatest? And: (2) Given that the analyses in the previous section pointed to various factors that can influence TM completion ability, what factors are actually most significant in the Internet?

4.1 Data

To answer these questions we analyze a large survey of AS paths in use in the Internet. Our data consists of a snapshot of all active BGP paths in use by 376 ASes *(monitors)*, taken at midnight UTC on August 6th, 2010. The dataset consists of over 100 million AS paths, and contains 524,761 unique prefixes. (Note that not all BGP tables show paths to all prefixes.) Because these paths are the *active* paths at the time of collection, each path represents the sequence of ASes that traffic will flow through when going from the particular monitor to the path's destination prefix.[2]

Next we select a subset of monitors and a subset of prefixes such that, for every monitor and every prefix, our dataset has the AS path from the monitor to the prefix. This results in 133 monitors and 225,041 prefixes.

Because we have the path from every monitor to every destination, we can construct visibility matrices for every AS appearing in the dataset – 28,763 ASes. These visibility matrices have size $133 \times 225{,}041$; for each entry in each visibility matrix, we can determine whether its value is 0 or 1. This is because we have the AS path corresponding to each element of the matrix, and to determine the 0-1 status of that element for a particular observer AS, we simply need to check whether the observer AS appears on that AS path. So our input to the analyses below consists of over 28,000 visibility matrices, each of which consists of about 30 million elements, known with high confidence.

Of course, these visibility matrices are only a portion of the complete visibility matrix of each AS, so our analyses in this section concern each AS's attempt to apply matrix completion to a portion of its TM.

In some our results, we make use of AS relationships (customer/provider and peer/peer); for that purpose we use the AS relationship labeling performed and published by CAIDA [11], which is based on the most comprehensive methodology available at present.

4.2 Metrics

We characterize an observer AS in two ways: via standard metrics used in the study of complex networks, and using metrics that capture networking-specific properties. First, to measure "centrality" of an observer AS, we use its *k-core decomposition shell* (or just "k-shell") [1]. As described in Section 2.5, the k-core decomposition identifies shells (vertex sets) of a graph that are nested, and successively more densely interconnected. Since we have seen in Section 3 that node degree is significant, we also measure each observer's *degree* (the number of ASs that are adjacent along a BGP path with the observer). Finally, we also consider networking-specific metrics: the number of customers of the observer, and the number of peers of the observer.

Each observer's TM completion ability depends on the rank k at which TM completion is attempted, with higher rank indicative of more accurate completion ability. In most cases in our data, observers cannot complete their entire TMs. However, AICMC identifies the subset of elements that can be recovered for any given rank. Thus rather than asking "at what rank k can the entire TM be recovered?" as we did in Section 3, here we use a different metric, which we call *expected rank*. Expected rank is defined as the expected value of the maximum rank at which a randomly chosen entry can be recovered. To compute the metric, we take the average over all non-visible entries of the maximum rank at which the element can be recovered (using zero when the element cannot be recovered at any rank).

Note that the matrices we use in this section are not structured the same way as those in Section 3. For example, matrices in this section are not symmetric, and are indexed differently. Because of this, rank values cannot be compared directly. Hence our focus is on how effective rank varies, rather than its specific value.

4.3 Results

We first consider whether centrality in the Internet as measured by k-shell is a good predictor of TM completion ability. For this, we look at the top 500 ASes in terms of k-shell number. Figure 9(a) shows a scatterplot of k-shell versus expected rank, and Figure 9(b) shows expected rank for ASes in order of decreasing k-shell number. In the this figure, values have been smoothed to reduce the effects of noise.

The figures show that centrality as measured by k-shell has some relationship to completion ability, but the relationship is not strong. Among ASes in the innermost shell (the nucleus), many have low completion ability. In fact, on average ASes in the core have lower completion ability than those 'just outside' the nucleus.

[2]We are ignoring possible configuration errors, false BGP advertisements, or path changes that have not yet reached the monitors – each of which we expect to have negligible effects on our results.

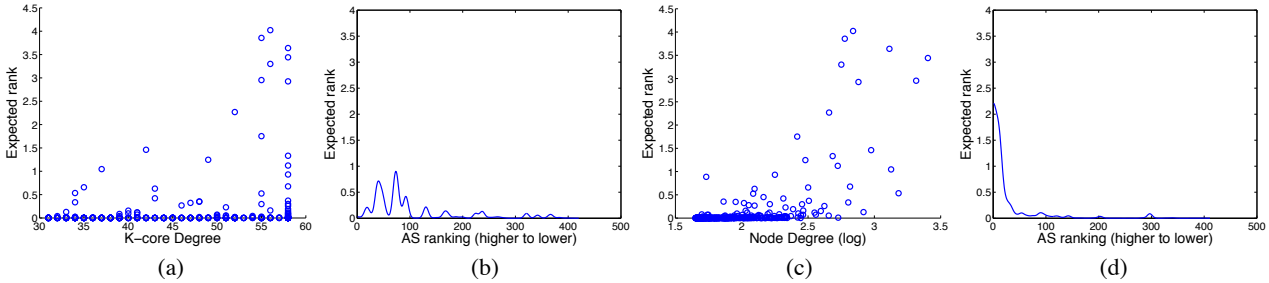

Figure 9: Expected rank as a function of (a),(b) k-shell and (c),(d) degree.

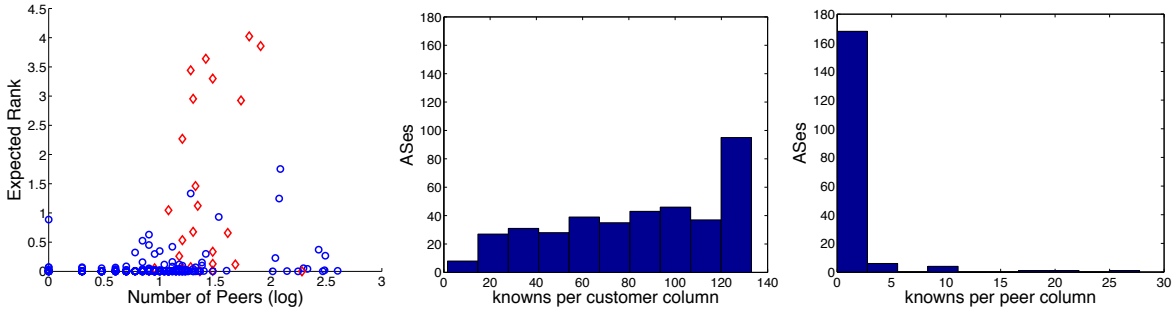

Figure 11: Effect of peers vs. customers completion ability.

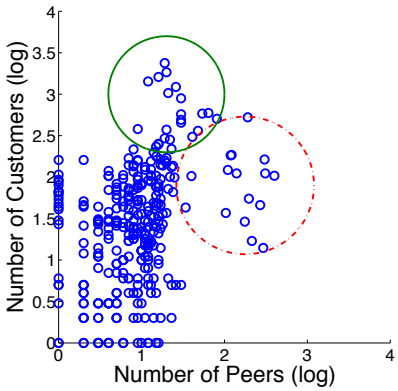

Figure 10: Peer degree vs. customer degree.

Since centrality *per se* is not a strong indicator of completion ability, we turn to the analyses in Section 3 to guide our intuition. Proposition 3.3 showed that increasing the degree of a node can increase its TM completion ability. The relationship between degree and TM completion ability is shown in Figures 9(c) and 9(d). The figures show that degree is a much better predictor of completion ability than centrality.

However, close examination of Figures 9(c) shows that some of the very highest-degree ASes have poor completion ability. Further consideration of the implications of Proposition 3.3 suggest an explanation that sharpens our understanding. The proposition was based on the assumption of shortest-path routing, and so does not directly apply to the AS graph. In particular, in the AS graph, a link may be between customer and provider, or it may be between two peers. The topology considered in Proposition 3.3 resulted in traffic between nodes flowing through the observer, and so links in that case were analogous to customer-provider links. In contrast, in the AS graph, traffic between two peers of the observer does *not* flow through the observer, because peers do not transit traffic for other peers.

This suggests that we should separate a node's degree into two components: the number of customer links, and the number of peer links.[3] This separation is shown in Figure 10, which plots customer degree against peer degree across the highest-degree ASes. The figure shows that high-degree ASes tend to fall into two different groups (shown in circles): some have more customer links than peer links, while others have more peer links than customer links.

Thus, it makes sense to analyze these two groups separately. If our analysis based on Proposition 3.3 is correct, ASes with high customer degree should show increased TM completion ability, while those with high peer degree should not necessarily show high completion ability.

This is in fact confirmed by our results, which are shown in Figure 11. Figure 11(a) shows TM completion ability versus the number of peers of the observer AS. There is no strong relationship between number of peers and completion ability; in fact the ASes with the greatest number of peers (more than 100) all have quite poor completion ability. In the figure, the red diamonds correspond to those ASes with the highest number of customers; it can be seen that these are the ASes with the greatest completion ability, but which typically have intermediate peer degree.

We can understand this difference by examing the influence of customers and peers on the visibility matrix of the

[3]The number of provider links per AS in our data is usually quite small and we ignore them in this analysis.

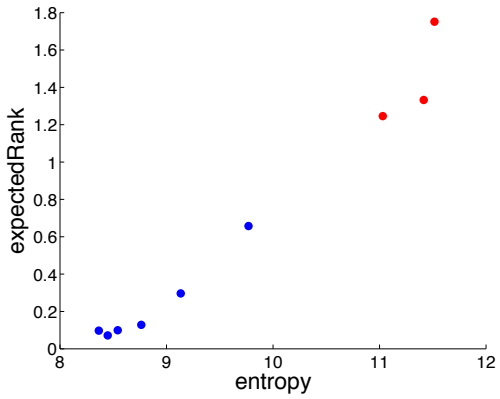

Figure 12: Expected rank vs. entropy for ASes having comparable densities.

observer AS. We do this by examining how many knowns (visible flows) are contributed to an observer AS on average by a customer and by a peer, for the set of ASes in Figure 11(a). Figure 11(b) shows a histogram of this quantity for customers, and Figure 11(c) shows the result for peers. The histograms show that often, a customer provides a highly dense column of the visibility matrix, while a peer typically provides very few entries in the visibility matrix. In particular, a single-parent stub customer provides a complete column.

In this regard, it is also important to note that improving completion ability is not simply a matter of maximizing the number of visible elements in the AS's traffic matrix. It is important *where* in the matrix the visible elements appear. In general, it is better for visible elements to be broadly distributed across columns and rows of the matrix. To demonstrate this fact, we select a set of 9 ASes with comparable density of visible elements — all ASes for which the number of visible elements lies in the range $(4 \times 10^5, 6 \times 10^5)$. To characterize the dispersion of visible elements we measure their entropy across columns. That is, for a matrix M of size $m \times n$, we compute $E = -\sum_{j=1}^{n} \frac{C_j}{N} log(\frac{C_j}{N})$ where C_j is the total number of knowns in column j and N is the total number of knowns in the entire matrix.

The relationship between entropy and expected rank for the 9 ASes is shown in Figure 12. When this entropy measure is large, visible elements are dispersed throughout the columns, while when it is small, visible elements are concentrated in few columns. The figure shows that ASes with very similar numbers of visible flows can vary considerably in their completion ability, and that completion ability is much better when visible elements are spread widely across the columns of the matrix.

In summary, our results in this section confirm key elements of our analysis from the previous section. In particular, our results point to the importance of having customers as a resource for TM completion. Further, we find that ASes best at TM completion are not generally those with a large number of peers, nor do they tend to be in the innermost, densest-connected k-shell.

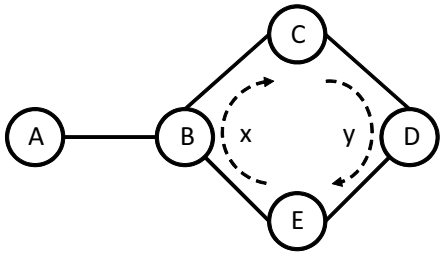

Figure 13: Computing distance to a flow. Flows x and y take AS paths of $E-B-C$ and $C-D-E$, respectively. The distance between A and flow x is 1 while the distance between A and flow y is 2.

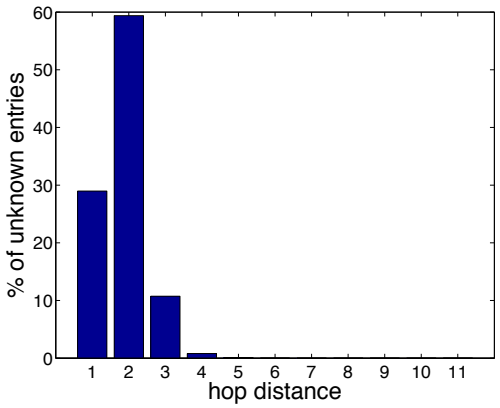

Figure 14: Distance to all flows.

5. WHICH ELEMENTS CAN BE RECOVERED?

While the results in the last section focused on comparing ASes globally across the Internet, we now turn to questions that are specific to individual ASes. Since a given AS may only be able to recover some of its invisible elements, it is important to develop an understanding of *which* elements are most readily estimated.

To capture the relationship between an AS and a flow that is invisible to that AS, we define a metric for distance between an AS and a flow. Figure 13 illustrates how flow distances are computed. For any given AS and flow, we find the shortest-path distance in the AS graph between the observer AS and each AS that the flow passes through. The distance between the AS and flow is the minimum of these shortest path distances. Of course, the distance to a known flow is zero.

To get a sense of typical distance values, we measure the distribution of distances across all (AS,flow) pairs. The result is shown in Figure 14. The figure shows that around 60% of unknown flows are distance 2 away from the observer ASes. Distances 1 and 3 follow by 30% and 10%, respectively; the percentage of unknowns that are further away is negligible. Thus, most unknown flows are at least two hops away from the observer AS.

Our first set of results characterizes the distance to flows that can be recovered, aggregating across all ASes. Figure 15 shows the fraction of unknown flows that can be recovered at

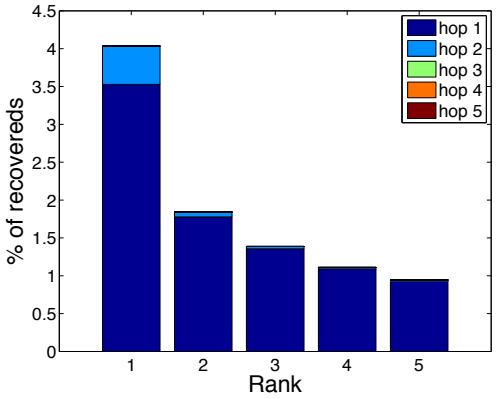

Figure 15: Distance to recovered flows.

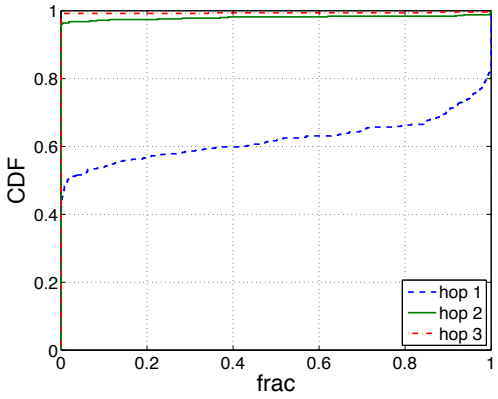

Figure 16: Success rate of targeted completion: fraction of target-visible unknowns that can be recovered in the predictor.

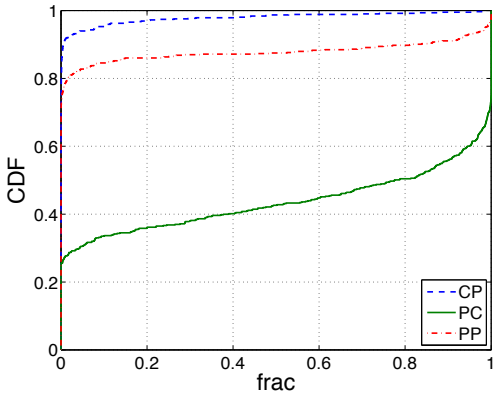

Figure 17: Success rate of targeted completion by predictor-target relationship.

each hop distance for varying rank values. At all rank values, the flows that ASes recover are primarily at distance 1. Only at rank 1 is there a non-negligible amount of flows recovered at distance 2 (despite the fact that distance 2 flows are much more numerous, as shown in Figure 14). The percentage of recovered unknowns at hop distance 3 and greater is negligible at any rank. These results show that there is a strong relationship between the distances to a particular flow and the potential to recover the flow. In particular, the unknown flows that an observer AS is most likely to recover are those that pass through its direct neighbors.

An important set of questions from a business intelligence standpoint concerns the ability of one AS (a *predictor*) to infer the set of flows that pass through some other particular AS (a *target*). We call this *targeted TM completion*. For example, consider the case described in the Introduction: an ISP may wish to know how much business a competitor is doing with a prospective customer. In this case the first ISP is the predictor and its competitor is the target.

To understand the ability of an AS to do targeted TM completion, we consider pairs of *(predictor, target)* ASes. Each pair has an associated hop distance in the AS graph. After constructing all such pairs and measuring their distance, we randomly sample 500 pairs at each distance. We then measure the fraction of the flows visible in the target that were filled-in during TM completion in the predictor. That is, let V be the set of elements visible in the target, U the set of unknown (invisible) flows in the predictor, and R the set of recovered flows in the predictor. Then for every pair we compute the fraction $frac = |V \cap R|/|V \cap U|$.

The results are shown in Figure 16 as a CDF across all 500 pairs at each hop distance. The figure shows that for pairs at hop distance 2 or 3, very little targeted completion is possible – in more than 95% of such cases, no targeted completion can be performed. However the situation is quite different for hop distance 1, which corresponds to ASes that are adjacent in the AS graph. In that case, only 45% of predictors cannot do any targeted completion. Most predictors can do some targeted completion, and for 19% of the predictors, *all* of their target's flows can be recovered. Thus, if an AS wishes to do targeted completion, its best targets are its neighbors.

While an AS's neighbors make the best targets, it is important to note that an AS can have a variety of different kinds of neighbors. We focus on three business/routing relationships that may exist between predictor and target: they may be *customer-provider* (CP), *provider-customer* (PC), or *peer-peer* (PP). Starting with our previous set of 1-hop AS pairs, we divide pairs into these three groups and examine the same metric as before (fraction of target unknowns completed). In the CP group, the predictor seeks to estimate flows passing through its provider; in the PC group, the predictor seeks to estimate flows passing through its customer; and in the PP group, the predictor is estimating flows passing through a peer.

The results are shown in Figure 17. The differences between the three cases is sharp. The least opportunity for targeted completion occurs when estimating a provider's flows; only about 10% can estimate any provider traffic. The situation is slightly better for peers: about 20% can estimate some peer flows, and a small percentage can estimate all of a peer's flows. However, the situation is very different for customer flows. Most providers can estimate a significant fraction of their customer's flows; and 30% can recover all of the flows passing through their customers.

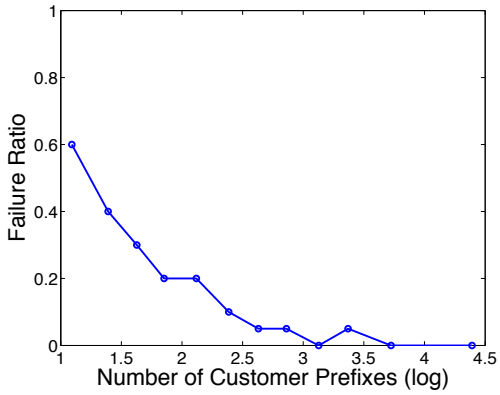

Figure 18: Estimation failure rate vs. number of customer prefixes.

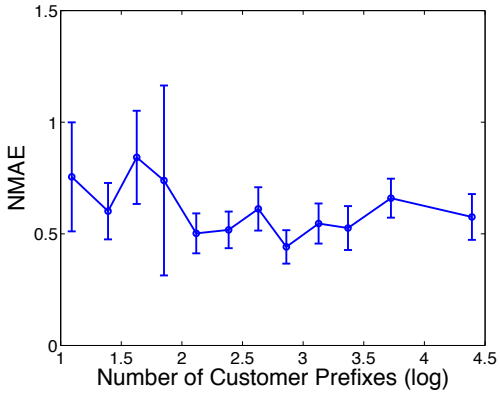

Figure 19: NMAE vs. number of customer prefixes.

6. ESTIMATION ACCURACY

Our results so far are in fact somewhat conservative: AICMC identifies when an element is surely estimable, but this does not mean that estimation of other elements is necessarily inaccurate in practice, particularly when using estimation algorithms other than ICMC. In general, it is important to confirm that the trends observed when using AICMC to analyze visibility matrices in fact agree with the results obtained when actually performing TM completion.

6.1 Approach

In order to confirm that our results are valid in practice, we perform actual matrix completion as it would be done by each AS. We provide each AS with only the knowledge of the TM entries as determined by its visibility matrix. We then perform matrix completion to estimate the entries that are invisible to that AS. Note that the fraction of the 30 million matrix elements visible to any AS varies from 0.3% (for the small number of ASes with highest density) down to 0.001% and lower for the vast majority of ASes.

Our focus is on evaluating how AS's visibility affects its TM completion ability, so it is important that we use similar traffic for studying each AS. We do not want our results to be affected by the differing nature of traffic in each AS (and obtaining actual traffic measures for each of the ASes in our dataset is out of the question in any case). Hence we take a single traffic matrix R (of real traffic, measured in the Géant network) and use it to populate each AS's TM. The traffic matrix R is a 54 $\times$ 54 submatrix of the entire Geant TM, and we have chosen rows and columns for R such that all 2,916 elements are visible to Geant (and therefore represent valid measurements). The elements of R consist of traffic flowing from ASes to prefixes, which matches the organization of our visibility matrices. We then populate an experimental TM D of size 133 $\times$ 225,041 by tiling D with copies of R. Although R is a rank of 54 matrix, our analysis of R (not shown) estimates its effective rank as 2 (95% of the variation in R can be captured in a rank-2 matrix), and so the rank of D is 54 and effective rank of D is 2 as well.

Since one of our goals in this section is to validate previous results that relied on ICMC, it is important that we use a different matrix completion algorithm for these experiments. For that reason we turn to an algorithm that works very differently from ICMC, namely *LMaFit* [25]. While ICMC works by incrementally constructing the matrix factors X and Y at full accuracy, LMaFit works by computing progressively more accurate versions of X and Y (in their entirety) via successive over-relaxation. LMaFit terminates when it (1) converges to a solution, meaning that visible elements are accurately represented in the solution or (2) detects an inability to converge, in which case LMaFit reports failure.

For each AS, we proceed as follows: First, we identify the visible elements of D (using the same visibility matrices as in Section 4). We next set the invisible elements in D to be zero. We then apply LMaFit to estimate the missing elements of D (using D's effective rank of 2 as the input value of k for LMaFit) yielding either failure, or a completed matrix $\hat{D}$.

We evaluate the results using two metrics: first, we want to know *whether* matrix completion can succeed: for this we note whether LMaFit succeeds in each case. Second, we want to know the *accuracy* of estimation that is possible in each case, which we measure using Normalized Mean Absolute Error (NMAE):

$$NMAE = \frac{\sum_{(i,j)\notin\Omega} |D_{ij} - \hat{D}_{ij}|}{\sum_{(i,j)\notin\Omega} D_{ij}}.$$

Note that the accuracy metric only applies to those cases where TM completion is successful.

6.2 Results

Our results compare number of prefixes announced by an AS's customers with its estimation failure rate and estimation accuracy. We sample 20 ASes in logarithmically spaced bins across the entire range of number of prefixes. In Figures 18 and 19, each point is the bin average, and 95% confidence intervals are shown in Figure 19.

Figure 18 shows that there is a strong relationship between number of customer prefixes and success rate of LMaFit. This is entirely consistent with the results in Section 4 and confirms that the ASes with large customer set can successfully perform TM completion. Figure 19 shows that the accuracy of TM completion can be quite good — generally between 0.5 and 1. Thus, as long as TM completion is possible, it can be done with high accuracy.

In fact, we find this last point to be true across all the experiments, i.e. regardless of the metric used to characterize ASes,

average NMAE is consistently in the range of 0.5 to 1, and there is no significant change in NMAE across metric values. This applies not just to number of customer prefixes, but to k-shell number and degree. In all cases, as long as TM completion is possible, it can be done with relatively high accuracy.

Thus, the metric that gives the most insight into TM completion ability is failure rate. We find that k-shell number is not a good indicator of low failure rate, whereas node-degree is (results not shown). This confirms our results from Section 4, and underscores that ASes with good ability to complete their TMs are generally those whose customers advertise large numbers of prefixes.

7. DISCUSSION

While the results in this study are suggestive, they do not precisely identify the TM completion ability of ASes. One reason is that in Sections 4 and 5 we are only working with a portion of each AS's visibility matrix. Although the visibility matrices we use have over 30 million elements, this is only about 0.5% of the full visibility matrix of an AS. That said, we have no reason to believe that the matrix portions we study are unusual.

Additionally, our results start from the assumption that TMs have low effective rank. While this fact has been empirically observed in numerous studies (as described in Section 2.2), all such observations to date have been at limited scale (hundreds or thousands of rows or columns). When considering TMs of the size in this paper (hundreds of thousands of columns) it is an open question whether and to what degree the property of low effective rank holds. However, this is a concern only if the AS seeks to complete its *entire* TM. For the results in Section 5 (including the business case described in Section 1) an AS is only concerned with completing a relatively small portion of its TM.

Broadly, the analytic and empirical sides of our study combine to yield a number of insights. In particular, our results suggest that:

- *ASes in the innermost k-shell of Internet are not necessarily effective at TM completion*. Proposition 3.4 showed that densely-meshed nodes can do TM completion, but only to a rank limited by the number of their customers. Empirically we find that densely-meshed ASes are not uniformly strong at TM completion (Figure 9(a)). In particular, the ASes that have the most peers are not especially well suited to complete their TMs (Figure 11(a)).

- *ASes with many single-homed customers are best suited to perform TM completion*. Propositions 3.5 and 3.8 show that it is good to have a large single-parent customer tree, and it is better for those nodes to be arranged in a wide tree rather than a deep tree. Empirically we find ASes with many customers are most effective at TM completion (Figure 11(a),(b)) and that an AS's customers contribute a large number of visible elements useful for TM completion (Figure 11(b)).

- *ASes are most effective at completing matrix entries that correspond to 'nearby' flows*. Flows that pass through neighboring ASes are more easily estimated than flows that do not pass through neighboring ASes (Figure 15). It seems that typical routing structures imply that flows that pass through neighbor ASes are more likely to have sources or destinations in common with visible flows, thus making recovery more likely.

- *When targeting specific ASes for completion, customer traffic is most readily estimated*. Among (predictor, target) AS pairs, the greatest completion ability exists when the predictor and target are neighbors (Figure 16) and in particular when the target is the customer of the predictor (Figure 17). Thus, not only do customers provide important information for completing TMs, but they are particularly good targets for TM completion.

The picture that emerges is that ASes with many direct single-homed customers have a particularly advantageous platform for performing TM completion. This suggests that ASes with many customers have a perhaps-underappreciated resource: not only do customers provide revenue, but the patterns of traffic that they send contain considerable information about traffic in other, more distant parts of the Internet.

8. CONCLUSION

In this paper we have investigated the application of the emerging concept of matrix completion to the specific case of Internet traffic matrices. The ability to perform matrix completion on TMs would provide considerable benefit spanning scientific, engineering, and commercial domains. Our goal is to understand how the structure of Internet routing and topology affects the ability of a given AS to estimate traffic flows that it cannot measure. We start by building intuition through analysis and we then deepen and extend our understanding using measurements of actual Internet routing.

We find that many ASes have the ability to perform at least partial TM completion. However which ASes are best at completion, and which elements they can recover, depends strongly on the local topology of the network. In particular, our study focuses attention on an AS's customers as its most important resource for TM completion. Customers provide rich information about traffic patterns; for example, a large array of single-homed stub customers provides an AS with the ability to infer invisible traffic even when the missing traffic is relatively complex (high rank). This suggests that many ASes scattered throughout the Internet have visibility into local traffic patterns that is well suited to inferring the nature of more distant, unmeasurable traffic.

9. ACKNOWLEDGEMENTS

This work was supported by NSF grants CNS-0905565, CNS-1018266, CNS- 1012910, and CNS-1117039. The authors thank the IMC referees and shepherd for their help in improving the paper. The authors also thank Renesys, Inc. for providing the data used in this paper.

10. REFERENCES

[1] J. I. Alvarez-Hamelin, L. Dall'Asta, A. Barrat, and A. Vespignani. k-core decomposition: a tool for the visualization of large scale networks. Technical report, Arxiv, 2005.

[2] V. Bharti, P. Kankar, L. Setia, G. Gürsun, A. Lakhina, and M. Crovella. Inferring invisible traffic. In *Proceedings of CoNEXT*, Philadelphia, PA, 2010.

[3] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20, March 2010.

[4] E. J. Candès and Y. Plan. Matrix completion with noise. *CoRR*, abs/0903.3131, 2009.

[5] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, 2009.

[6] S. Carmi, S. Havlin, S. Kirkpatrick, Y. Shavitt, and E. Shir. A model of internet topology using k-shell decomposition. In *PNAS*, volume 104, pages 11150–11154, July 2007.

[7] H. Chang, R. Govindan, S. Jamin, S. J. Shenker, and W. Willinger. Towards capturing representative AS level Internet topologies. *Computer Networks*, 44(6):737–755, 2004.

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

[9] H. Chang, S. Jamin, and W. Willinger. To peer or not to peer: Modeling the evolution of the Internet's AS-level topology. In *Proceedings of Infocom*, 2006.

[10] H. Chang, M. Roughan, S. Uhlig, D. Alderson, and W. Willinger. The many facets of Internet topology and traffic. *Networks and Heterogeneous Media*, 1(4):569–600, 2006.

[11] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, kc claffy, and G. Riley. AS relationships: Inference and validation. *ACM SIGCOMM CCR*, 37(1):29–40, 2007.

[12] V. Erramilli, M. Crovella, and N. Taft. An independent-connection model for traffic matrices. In *Proceedings of IMC*, pages 251–256, 2006.

[13] 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.

[14] L. Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Trans. Netw.*, 9:733–745, December 2001.

[15] Z. Ge, D. R. Figueiredo, S. Jaiswal, and L. Gao. On the hierarchical structure of the logical Internet graph. In *In Proceedings of SPIE ITCOM*, pages 208–222, 2001.

[16] G.Siganos, S.L.Tauro, and M.Faloutsos. Jellyfish: A conceptual model for the AS Internet topology. *Journal of Communications and Networks*, 2006.

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

[18] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. In *Proceedings of ACM SIGCOMM 2004*, pages 219–230, August 2004.

[19] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. In *Proceedings of SIGMETRICS '04/Performance '04*, New York, NY, USA, 2004. ACM.

[20] 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.

[21] R. Meka, P. Jain, and I. S. Dhillon. Matrix completion from power-law distributed samples. In *Proceedings of NIPS*, December 2009.

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

[23] M. Roughan. Simplifying the synthesis of internet traffic matrices. *SIGCOMM Comput. Commun. Rev.*, 35:93–96, October 2005.

[24] L. Subramanian, S. Agarwal, J. Rexford, and R. Katz. Characterizing the Internet hierarchy from multiple vantage points. In *In Proc. IEEE INFOCOM*, 2002.

[25] 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.

[26] 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):947–960, 2005.

[27] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu. Spatio-temporal compressive sensing and internet traffic matrices. *SIGCOMM Comput. Commun. Rev.*, 39(4):267–278, 2009.