Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

MODELING AFFILIATIONS IN NETWORKS

Brian Cloteaux

Mathematical and Computational Sciences Division
National Institute of Standards and Technology
100 Bureau Drive, Stop 8910
Gaithersburg, MD 20899-8910, U.S.A

ABSTRACT

One way to help understand the structure of certain networks is to examine what common group memberships the actors in the network share. Linking actors to their common affiliations gives an alternative type of network commonly called an affiliation network. Recently, there have been several studies examining the problem of modeling the dynamics of a network through the changes in the affiliations of its actors. We examine the closely related problem of modeling the affiliations for a given network. We especially focus on the case of trying to mine these affiliations when the original network is potentially missing links.

1 INTRODUCTION

In a general sense, a network simply records some type of relationship between entities. For example, in a collaboration network, the entities can be authors and the links can designate whether the authors have ever collaborated on a publication. This type of modeling allows us to treat a network as a simple undirected graph.

Alternatively, many networks can be naturally represented as relationships between entities and the groups (or *affiliations*) to which the entities belong. Going back to the example of the collaboration network, we can represent the same information as a set of authors and a set of papers produced by those authors. If an author has written a paper, then there is a link between author and paper. We notice that we can still extract the information about coauthorships by examining the coauthors on all the papers that an author has produced.

This idea of analyzing networks in terms of the shared affiliations between individuals has a long history among sociologists. Lattanzi and Sivakumar (2009) trace the idea of analyzing the explicit relationship between persons and groups in terms of networks of interpersonal and intergroup ties to the work of Breiger (1974). Even earlier, Wasserman and Faust (1994) (page 292) note that George Simmel, in work done in the 1950's, defined an individual's social identity in terms of the group affiliations to which they belong. Rather than concentrating on relations between individuals, the focus was on understanding common affiliations.

A recent resurgence in affiliations research has been partly because of the observation that there are a large number of real-world networks that have a natural bipartite structure inherent in them. Examples of networks with some type of natural bipartite structure include the large number of different collaboration networks. Specific examples include participation in social events, shared memberships in social clubs for CEOs, scientific collaboration, actors in common movies, coownership of companies, and shared membership on boards of directors (Newman, Strogatz, and Watts 2001, Newman, Watts, and Strogatz 2002). But this appearance of bipartite networks goes beyond just collaboration networks. For example in linguistic analysis, networks can be formed by linking words that commonly occur together in sentences (Cancho and Solé 2001). In this case, the sentences can be viewed as membership groups for the words themselves. In another example, we can view also users of peer-to-peer systems as forming bipartite graphs. Here the membership criteria is the common sharing of some piece of information (Le Fessant, Handurukande, Kermarrec, and Massouli 2005).

The increase in research in affiliation networks has also been spurred by a pair of recent developments. First, it has been shown that we can analyze any network in terms of an associated affiliation network, whether or not it has a natural underlying bipartite structure. Even for networks without an obvious bipartite decomposition, it has been shown that these networks can still be associated to some minimal set of affiliations (Guillaume and Latapy 2004) and therefore methods which have been developed for analyzing the affiliations (Latapy, Magnien, and Vecchio 2008) can be usefully applied as general analysis tools for all networks.

A second development has been in showing how networks can evolve in terms of changes to their affiliations (Zheleva, Sharara, and Getoor 2009, Lattanzi and Sivakumar 2009, Guillaume and Latapy 2006). Within the past two years, it has been shown that several desirable properties for models, seen in real-world networks, can be explained and captured by examining the changes to the affiliations associated with the networks. These properties include power-law distribution, clustering, densification, and a shrinking diameter (Barabási and Albert 1999, Watts and Strogatz 1998, Leskovec, Kleinberg, and Faloutsos 2005). It appears that understanding the affiliations associated with certain networks is necessary to be able to accurately model how those networks will evolve.

With both of these developments in mind, we have started to investigate how to find or mine the affiliations associated with a network. While there is an extensive body of literature on mining various aspects and properties from networks (Chakrabarti and Faloutsos 2006), there has been very little published in trying to determine affiliations. The principle result is from Guillaume and Latapy (2004) who gave a simple algorithm for finding a minimal set of affiliations from a given network. Their work is based on the assumption that we have perfect knowledge of the network in which we are extracting the affiliations. In this paper, we seek to extend their results by examining networks where we are potentially missing information. Specifically, we are interested in mining affiliation in networks when there may be missing links between the entities. Our main result is our proposal of a method for finding many of the affiliations for in these networks with missing links. We also discuss some assumptions that we made in creating our algorithm and some of the implications that these assumptions have on its use.

2 CREATING AFFILIATIONS NETWORKS

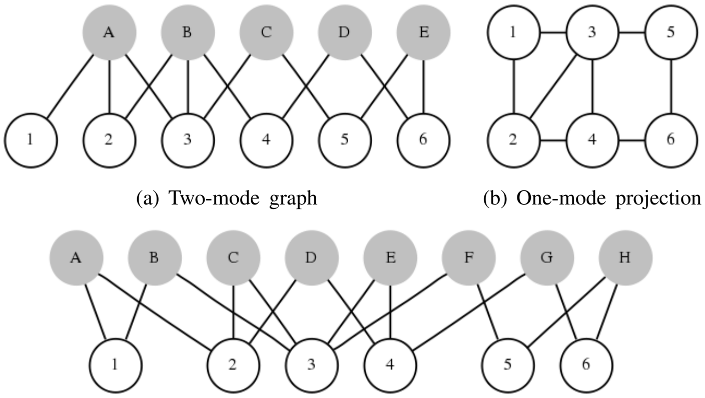
Throughout the remainder of this paper, a network is defined be to a simple undirected graph, where the nodes are the entities in the network and the edges between the nodes represent some sort of relationship between pairs of entities. This type of network is commonly called a *one-mode representation*. For the one-mode network G, it is designated as $G = (N,E)$ where N is the set of nodes or entities and E is the set of edges. To denote the set of the edges in a specific network G , we use the notation $E(G)$.

An alternative representation of the data in a network is its *affiliation network*, which is also called the *two-mode representation*. We will use both terms interchangeably in our discussion. When discussing affiliation networks in a formal manner, we will use notation borrowed from Guillaume and Latapy (2004). An affiliation network A is a bipartite graph $A = (\top,\bot,E)$ where $\top$ and $\bot$ are the disjoint sets of nodes composing the top and bottom sets respectively and E is the set of edges. The bottom set represents the entities in the graph, while the top set is the set of affiliations. All edges in E link a node in $\top$ to a node in $\bot$ (i.e. $E \subseteq \top \times \bot$).

If we are given the two-mode representation of a network, we can easily and uniquely obtain the corresponding one-mode representation. This network is called the *one-mode projection* of the affiliation network. In order to take the one-mode projection we start by defining its set of nodes as the bottom set in the two-mode graph. Then to define the edge set, we record an edge between two nodes if those nodes share an affiliation in the two-mode representation. In other words, an edge (n_i,n_j) exists in the one-mode network G only if there exists an affiliation node $a \in \top$ such that the two-mode graph has the edges (n_i,a) and (n_j,a). It should be pointed out that although multiple edges between two nodes can be defined from nodes sharing more than one affiliation, since we only deal with simple graphs, only one edge is recorded in the projection. An example of a two-mode network and its one-mode projection is given in Figure 1. To denote the one-mode projection of a two-mode network A, we will use the notation $\bot(A)$.

While there is a unique one-mode projection for any two-mode network, the converse is not true. It is easy to show multiple two-mode graphs with identical one-mode projections. As a trivial example, we can take each edge in a one-mode graph to be its own affiliation. For example, in the graph in Figure 1(b), we can create the two-mode graph from each of the edges as shown in Figure 1(c). Since there can be multiple possible affiliation networks for each one-mode graph, this raises a question over which of two-mode graphs should we use as a model of the affiliations.

The simplest answer, which was originally put forth by Guillaume and Latapy (2004), is to look for a minimal two-mode representation whose one-mode projection is the network we are modeling. To clarify what we mean by a minimal two-mode representation, we measure the size of these two-mode graphs by the size of their top sets. So a minimal two-mode graph is



(a) Two-mode graph

(b) One-mode projection

(c) Alternative two-mode graph

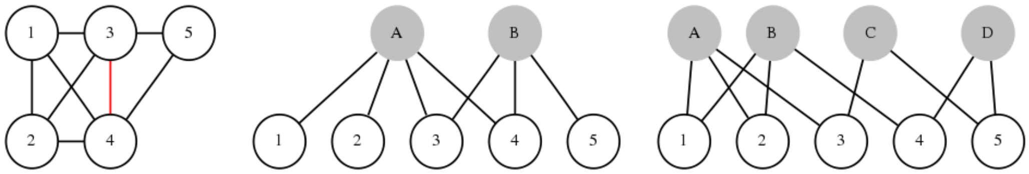
Figure 1: A two-mode network and its one-mode projection. Here in Figure 1(a), the set of entities or the bottom set is $\perp = \{1,2,3,4,5,6\}$ while the set of affiliations or the top set is $\top = \{A,B,C,D,E\}$. In Figure 1(b), we see the one-mode projection of the affiliation network. Here there is an edge between two nodes in the one-mode network if those nodes both are connected to the same affiliation. Notice, that the edge from node 2 to node 3 is specified in both affiliations A and B, but is represented as a single edge in the one-mode network. In Figure 1(c), we see an additional two-mode graph whose one-mode projection is also the graph in Figure 1(b). This shows that there is not necessarily a unique two-mode network corresponding to every one-mode graph.

minimal in the number of affiliations. Unfortunately, finding such a minimal two-mode representation is a computationally difficult problem in general. To understand why this is, we see that all nodes attached to an affiliation in the two-mode network form a clique among themselves in its one-mode projection. In other words, the problem of finding a two-mode representation for a network is essentially the problem of finding a set of cliques which cover of all the edges in the graph. This is a long studied problem, and it has been shown that in general, finding a *minimal* clique cover of a graph is NP-hard (Kou, Stockmeyer, and Wong 1978, Garey and Johnson 1990). Therefore, we cannot expect for there to be a polynomial time algorithm for finding these clique coverings in all cases.

In practice though, it is usually not difficult to produce a minimal or almost minimal two-mode representation for most real-life networks. A simple heuristic for producing two-mode representations was given by Guillaume and Latapy (2004). Their method starts by taking each edge in the network and finding a maximal cliques which contains that edge. By collecting together all these maximal cliques and assigning each one an affiliation, we get a good approximation of a minimal two-mode representation. This works quickly for many real-world networks since networks with power-law distributions tend to have small cliques in them.

Unfortunately, this straightforward clique decomposition scheme does not work well if we are missing information about the network. In particular, if we are missing edges in our network, then a minimal clique covering can be significantly larger than the true value. In Figure 2, we see an example of why this can be true. The edge highlighted in the network in Figure 2(a) belongs to two different large cliques embedded in the network. Thus, as the figure shows, if we miss that edge, the size of the minimal two-mode representation doubles.

If we are going to be able to accurately model the affiliations associated with a network which is missing links, we cannot simply look at minimal clique coverings. Instead, we need to use some generalization of the clique concept that allows for missing edges. We will explore some different definitions that meet this criteria in the next section, but we first should note an important property that we are looking for in any generalization that we choose. While we do not want to create a two-mode representations whose projection is exactly the original network, we also do not want to lose any information in the form of edges in our one-mode projection. In other words, for any network G and its two-mode network A which is produced using some type of generalization of the clique concept, we want the assurance that $E(G) \subseteq E(\perp(A))$. Therefore, we will allow additional edges to be introduced in one-mode projection in order to have higher cohesion in the affiliations, but we never allow the existing edges to be changed.



(a) One-mode network (b) Minimal two-mode representation for the network with the edge (3,4) (c) Minimal two-mode representation for the network without the edge (3,4)

Figure 2: An example of how missing links in a network can significantly increase the size of the minimal two-mode representation. For the network in figure 2(a), we show that removing the edge between the nodes 3 and 4 doubles the size of the minimal two-mode representation.

3 LOOKING FOR AFFILIATION RELAXATIONS

An affiliation is one of a number of ways of defining "social cohesion" (Moody and White 2003). In the literature, these different versions usually end up as some generalization of the clique concept. One of the earliest suggestions was made by Luce (1950) when he introduced the *k-clique*. A k-clique is a maximal subset S of the nodes of a graph G where the minimal path length in the graph G between any two nodes in the set S is at most k. This is a generalization of cliques since every 1-clique is a clique, but one problem with this definition is that it allows for distant and even disconnected nodes to be grouped together. A way to strengthen the k-clique definition was given by Alba (1973) who defined the *k-clan*. A k-clan S is a k-clique such that the diameter of S is no more than k. Thus every k-clan is a k-clique, but not vice-versa. A further generalization was given by Mokken (1979) who defined the *k-club* as a maximal subgraph of a graph G that has diameter k.

The above definitions are all based on distances within a graph, but for reasons that we will discuss in more detail in the next section, we are more interested in generalizations based on node degree. The clique generalization that we will focus on was originally introduced by Seidman and Foster (1978). A *k-plex* is essentially a complete graph which allows for a certain number of edges to be missing in it. Formally, a graph G is called a k-plex if, for some positive integer k, the minimal node degree in G, denoted as $\mu(G)$, has the size that is at least $\mu(G) \geq |N| - k$. From this definition, in a 1-plex every node would connect to the $|N| - 1$ other nodes and so would form a clique. For a 2-plex, each node could be missing at most one connection to the other nodes. This definition of a "near clique" is the one we will use throughout the remainder of this paper while searching for affiliations.

Since we have already mentioned that finding a maximal 1-plex, or clique, is NP-hard, then we would not expect for there to be a polynomial time algorithm for finding maximal k-plexes. Recently, there have been introduced some new algorithms for computing maximal k-plexes (such as Balasundaram, Butenko, and Hicks (2010) and Moser, Niedermeier, and Sorge (2009)), but, in practice for networks with power-law distributions, using extensions of the combinatorial algorithms for finding maximal cliques (for example Carraghan and Pardalos (1990), Östergård (2002)) tend to work quickly and are simple to implement. A detailed discussion on how to extend these combinatorial methods for finding maximal cliques into methods for finding k-plexes was made by McClosky and Hicks (2009).

In looking for a minimal or near-minimal k-plex covering of a network, our starting point is the algorithm of Guillaume and Latapy (2004). Like their algorithm, our routine finds the maximal k-plexes for each edge using a modification of an algorithm originally given by Carraghan and Pardalos (1990) for finding maximal cliques in a graph and extended to finding maximal k-plexes by McClosky and Hicks (2009). This algorithm returns us a set of potential affiliations with which we can use to construct our two-mode representation. In Algorithm 1, we give this simple algorithm for finding the maximal connected k-plexes which contain a given edge.

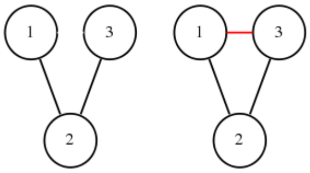
Our algorithm does have a couple of important differences from the algorithm of Guillaume and Latapy (2004). A first difference is that we only consider k-plexes that are connected. The reason why we check for connectivity is discussed in the next section, but obviously this type of explicit check is not needed for simple clique coverings since all cliques are connected. In fact, even for most k-plexes this type of explicit connectivity check is not always needed. Seidman and Foster (1978) showed that for all k-plexes, if

$$k < \frac{|N|+2}{|N|}$$

```
**ConnectedKPlexSearch**($K$,$U$,$\mathscr{M}$)
**Input**: $K$–a set of nodes forming a $k$-plex, $U$–the set of candidate nodes, $\mathscr{M}$–the set of maximal $k$-plexes found
**while** $U \neq \emptyset$ **do**
    $max \leftarrow |M|$ for any $M \in \mathscr{M}$
    **if** $|K| + |U| \leq max$ **then**
        **return**
    **end**
    arbitrarily choose some $v \in U$
    $K \leftarrow K \cup \{v\}$
    $U \leftarrow U - \{v\}$
    $U' \leftarrow \{u \in U : K \cup \{u\}$ is a $k$-plex and is connected$\}$
    **ConnectedKPlexSearch**($K, U', \mathscr{M}$)
**end**
**if** $|K| > max$ **then**
    $\mathscr{M} \leftarrow \emptyset$
**end**
**if** $|K| \geq max$ **then**
    $\mathscr{M} \leftarrow \mathscr{M} \cup \{K\}$
**end**
**return**
```

Algorithm 1: This algorithm **ConnectedKPlexSearch** is for computing all the maximal connected k-plexes for a graph. To find a maximal connected k-plex for a graph $G = (N,E)$, this algorithm is invoked with $K = \emptyset$, $U = N$, and $\mathscr{M} = \emptyset$. If we instead initialize K with the nodes from an edge, then the algorithm will compute the maximal connected k-plexes containing that given edge.

then not only must the k-plex be connected, but we can also establish that its diameter is no more than two. Thus we actually only need to perform a full connectivity checks if k fails to meet this condition.



(a) Example network

(b) The k-plex that is matched for any $k \geq 2$

Figure 3: This figure shows a common problem of matching small affiliations to k-plexes where $k \geq 2$. In Figure 3(a), we see a a two edge network that can be covered by two affiliations. As shown in Figure 3(b), for any $k \geq 2$, the entire network is a single connected k-plex.

A second difference is that we return *all* of the maximal k-plexes found containing a certain edge, not just one member of the maximal set. The reason for this is that as k becomes larger, the number of different maximal k-plexes also greatly increases. Thus if we randomly choose one maximal k-plex for each edge when we are trying to create an associated affiliation network, then the size of our affiliation set can approach the number of edges in the network. Thus, we end up collecting all the possible k-plexes and then passing them off to a second algorithm (shown in Algorithm 2) which sorts through them to find minimal covering set out of the collection of maximal k-plexes.

There is an additional point that needs to be made about how we collect the maximal k-plexes for each edge. When we match k-plexes against nodes of low degree, we can generate false affiliations. This can result with us finding a much smaller set of affiliations than there actually are. To illustrate this problem, we have an example in Figure 3. Here, if we start with the network in Figure 3(a), we notice that matching this to any k-plex, where $k \geq 2$, will cause the false connection between nodes 1 and 3 to be included. Since networks with a power-law distribution have many degree one nodes, then this example problem with k-plexes will be often repeated and lead to a large number of false affiliations. To prevent this, we implemented a cutoff size in our maximal k-plex collections. If the maximal k-plexes for an edge are smaller than the cutoff size, we then replaced those k-plexes with the edge itself. In our tests, we typically took four nodes to be the cutoff size.

```
MinimizeNumKPlexes($\mathscr{C}$,$E$,$\mathscr{A}$,$M$);
Input: $\mathscr{C}$–collection of minimal affiliations for each edge, $E$–set of edges not in any affiliation, $\mathscr{A}$–current minimal
       affiliation set, $M$–smallest affiliation set found
if $|\mathscr{A}| > |M|$ then
    return
end
if $E = \emptyset$ then
    if $|M| > |A|$ then
        $M \leftarrow A$
    end
    return
else
    arbitrarily take any $e \in E$
    foreach $D \in \mathscr{C}_e$ do
        $\mathscr{B} \leftarrow \mathscr{A} \cup \{D\}$
        $N \leftarrow E$
        foreach $p \in D$ do
            $N \leftarrow N - \{p\}$
        end
        MinimizeNumKPlexes($\mathscr{C}$,$N$,$\mathscr{B}$,$M$)
    end
end
return
```

Algorithm 2: The algorithm **MinimizeNumKPlexes** is used to find a minimal set of k-plexes which covers all the edges in a network. The algorithm is passed a collection of k-plexes for each edge and a list of edges. It then returns a minimal set selected out of the collection which covers all edges.

4 ALGORITHM ASSUMPTIONS

In our approach to finding affiliations, we make two simplifying assumptions about how links are missed from the one-mode network representation. The first assumption is that the probability of finding an edge that exists in a network is some identical value p for all edges and the second assumption is that each of these probabilities for all the edges are independent. Although it is debatable whether these assumptions are valid in many domains, they do provide a well-studied starting point for our investigation.

They also provide some justification for some of the choices in our algorithm. We can create a random graph by starting with a complete graph and for each edge deciding with probability p whether to leave the edge in or remove it. This is called an Erdős-Rényi random graph. We are essentially looking for subgraphs of this type when we are looking for affiliations in networks which have missing links under the above assumptions. Since this is a well-studied area, we can use the known properties about Erdős-Rényi graphs in order guide our search.

One result that we use is the fact that resulting degree distribution of a Erdős-Rényi graph has a Poisson distribution (see chapter 3 of Bollobas (2001)). Since in a Poisson distribution the degrees will be fairly tightly grouped around the mean, then using a clique generalization which limits how small the minimal degree can be, i.e. the k-plex, approximately captures the set of nodes in this distribution. In other words, if we chose a large enough k, then we should be able to match the maximal k-plexes against a majority of the affiliations.

In addition, it was shown by Erdős and Rényi (see chapter 7 of Bollobas (2001)) that for probability p of an edge being included in a graph, then if

$$p > \frac{\ln|N|}{|N|}$$

the graph is almost surely connected and if it is less than this bound then the graph is almost surely disconnected . This threshold result provides a justification why we only examine connected k-plexes in Algorithm 1. Unless the probability of finding an edge is very small, all the affiliations should be connected.

It also follows that as long as $k \leq |N| - \ln|N| \cdot \frac{|N|-1}{|N|}$ then any k-plexes which match possible affiliations will be connected. We can use this value as a bound on the resolution of our method under these assumptions. If p is less than the threshold bound given above, then we cannot tell the difference between a true affiliation and a simple group of disconnected vertices. Thus, when p is below this threshold, for any value of k we will not be able to dependably mine any affiliation information. It follows that if we allow the value of k to be greater then the above bound, it would be to find affiliations in graphs where the value of p is less than the threshold value. Thus, this method could not be profitably used beyond this bound on k.

5 TESTING THE METHOD

In order to test our approach, we generated a number of random affiliation networks, and then randomly removed edges from their one-mode projections. We then tested how well we could reconstruct the original affiliation networks from the modified one-mode projections.

In order to construct a random affiliation network, there are two degree distributions that have to be considered: the degree distribution of the bottom set and the degree distribution of top set. By generating two distributions, we can build a random network. Since many of the networks we are interested in modeling have power-law like degree distributions (Barabási and Albert 1999), we would like to create an affiliation graph whose one-mode projection has a power-law degree distribution. A helpful result for this type of model generation was given by Guillaume and Latapy (2006) where they proved that if the bottom distribution of an affiliation network has a power-law degree distribution, then the degree distribution of its one-mode projection will also have a power-law distribution with the same exponent.

For the top distribution, Guillaume and Latapy (2006) point out that there are some real networks with Poisson distributions for their top set while other real-world networks have shown a power-law degree distribution for their top set. For our tests, we only use a power-law distribution with an $\alpha = 2.5$ for the top set. In addition, to assure that the degrees of all the nodes is at least two in the top set, we added one to each node degree that we randomly picked from the power-law random number generator. This choice does not significantly change the results of our experiments since we are principally interested in determining how accurate our method is in determining affiliations.

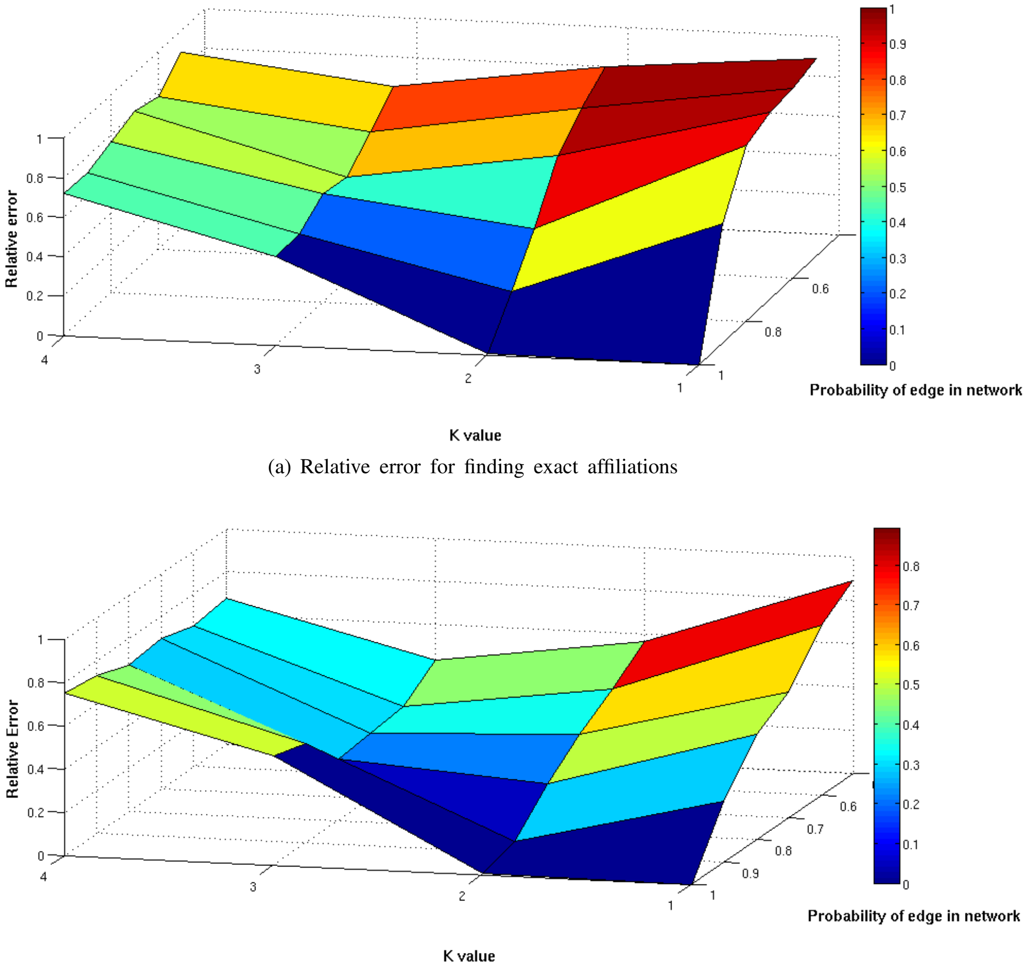
In order to generate a random affiliation network, we use an approach that was described by Guillaume and Latapy (2006). Their generation algorithm randomly assigns the degree values, drawn from a given distribution, for the vertices from the top and bottom set. If the sum of the top and bottom degree sequences do not match, then a random value from both sets is dropped and redrawn until the sums match. This method for sampling allows us to randomly obtain the degree distributions without introducing bias and is discussed in Newman, Strogatz, and Watts (2001) and Newman, Watts, and Strogatz (2002). Finally, we randomly connect the vertices from the bottom to the top set. If the model is large enough, we can safely ignore any multiple edges between nodes in the top and bottom sets without affecting the degree distribution of the resulting network.

With the creation of a random affiliation network, we now take its one-mode projection. Given a probability p of an edge in the network being detected, we create a new network by randomly deciding, with probability p, whether to include each edge from the one-mode projection. This gives us a network to test using our algorithm and the set of affiliations that generated it.

We first examined how well that Algorithm 1 is able to find the large affiliation in the test networks. For each k from 1 to 4, and for a series of p values from 1 to 0.5, we created 30 random networks of 200 nodes each. Then, for the affiliations of at least 4 nodes, we examined average relative error that Algorithm 1 had in finding these affiliations. The graph of those results is shown in Figure 4(a). These results show that, except for $p > 0.9$ and $k \leq 2$, there can be a large relative error in finding the exact set of nodes for large affiliations. An explanation for this discrepancy lies in the fact that the degree distribution in a k-plex does not exactly match a Poisson distribution in many cases. In a Poisson degree distribution, most of the values are fairly tight around the mean value, but if the affiliation is large enough we can expect for there to be nodes in the affiliation whose degree value is less than the lowest value allowed in the k-plex. We would thus miss those nodes by matching a k-plex against them.

Since we realize that we can potentially be losing some of nodes of low degree by matching them against k-plexes, we next examined if we are able to detect significant subsets of those affiliations. For this test, we determined if any of the large k-plexes we found were large subsets of an affiliation. In Figure 4(b), the average relative error is recorded for these tests and we notice the relative error dramatically decreases when searching for these large subsets instead. This insight gives us some confidence that our approach will at least detect the majority of the nodes in the large affiliations that are present. This seems especially true when $p > 0.8$ and $k \geq 3$.

In our next tests, we examined how well we were able to predict the size of the top set. Because of the problem of matching k-plexes against small affiliations, we set a cutoff value for matching of four nodes. We compared a series of 30 random networks of 200 nodes each against predicted size of minimizing a set of maximal affiliation using Algorithm 2. The set we minimized was the set of 2-plexes produced by Algorithm 1 whose size was at least four nodes. For any edge



(a) Relative error for finding exact affiliations

(b) Relative error for finding large subsets of affiliations

Figure 4: This figure shows the relative error in finding large affiliations in networks. In Figure 4(a), we see the relative error of finding the exact affiliations whose size is at least 5. For each point in the graph, a series of 30 random affiliation graphs each of size 200 nodes were used.

whose maximal 2-plex was under four nodes, we instead used the nodes in the edge itself as the 2-plex. As shown in Figure 5, for $p > 0.7$, we had very low relative error in predicting the size of the top set. For $0.75 \leq p \leq 0.8$, we had almost no error at all. Thus, this method does seem to be able to give an accurate idea of the number of affiliations when the p value is not too small.

6 CONCLUSIONS

This paper is meant to be a starting point in investigating how to extract meaningful affiliations from network data. We have focused on reconstructing affiliations when there are edges, or relationships between the entities, that are potentially missing from our network data. While we have demonstrated some success in our approach, there are several areas where we can extend and improve on our techniques.

Our investigation made two important assumptions: that the probability of missing a link is identical and independent for all edges. Based on this assumption, we investigated using k-plexes to cover the original network. There are a number of directions for future research from this starting point. One starting point is to relax the assumptions we made on the model. The idea of allowing different failure probabilities certainly has application over certain domains. In addition, depending on

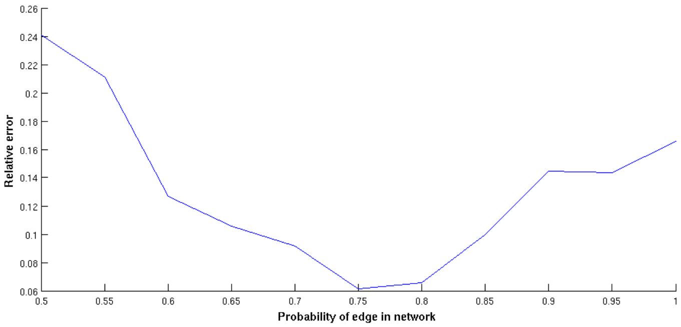


Figure 5: This shows the relative error in the determining the size of the top set for a number of 200 node random networks while using 2-plexes with a cutoff value of 4. We see that even the maximum relative error has a small value.

how the network data is collected, dependencies between the probability that edges appear in a network can be introduced (Willinger, Alderson, and Doyle 2009). Addressing either issue will cause us to have to rethink the algorithms.

Another direction is to refine our search for affiliations. An obvious source of inaccuracy is that the structures we are looking for have a Poisson degree distribution, but we match these against k-plexes. Since a k-plex limits the lowest degree allowed in it, the nodes in the affiliation with low degree are potentially not included by matching against a k-plex. This implies that for larger affiliations, we often can only find a subset of it and not the entire affiliation. A direction that we are actively researching is the use of quasi-cliques with Poisson degree distributions for the modeling of the affiliations instead of k-plexes. A quasi-clique puts a lower bound on the number of edges in it, instead of the node degree (Abello, Resende, and Sudarsky 2002). By looking for only quasi-cliques with a Poisson degree distribution, we hope to improve the accuracy of our method.

ACKNOWLEDGMENTS

This work is a contribution of NIST, an agency of the US government, and is not subject to US copyright.

REFERENCES

Abello, J., M. Resende, and S. Sudarsky. 2002. Massive quasi-clique detection. In *LATIN 2002: Theoretical Informatics*, 598–612. LNCS 2286.

Alba, R. D. 1973. A graph-theoretic definition of a sociometric clique. *Journal of Mathematical Sociology* 3:113–126.

Balasundaram, B., S. Butenko, and I. V. Hicks. 2010. Clique relaxations in social network analysis: The maximum k-plex problem. *Operations Research*. To appear.

Barabási, A.-L., and R. Albert. 1999. Emergence of scaling in random networks. *Science* 286:509–512.

Bollobas, B. 2001. *Random graphs*. 2nd ed. Cambridge University Press.

Breiger, R. L. 1974. The duality of persons and groups. *Social Forces* 53 (2): 181–190.

Cancho, R. F., and R. V. Solé. 2001. The small world of human language. *Proceedings of the Royal Society of London. Series B: Biological Sciences* 268 (1482): 2261–2265.

Carraghan, R., and P. M. Pardalos. 1990. An exact algorithm for the maximum clique problem. *Operations Research Letters* 9 (6): 375–382.

Chakrabarti, D., and C. Faloutsos. 2006. Graph mining: Laws, generators, and algorithms. *ACM Computing Surveys* 38 (1): 2.

Garey, M. R., and D. S. Johnson. 1990. *Computers and intractability; a guide to the theory of NP-completeness*. New York, NY, USA: W. H. Freeman & Co.

Guillaume, J., and M. Latapy. 2004. Bipartite structure of all complex networks. *Information Processing Letters* 90 (5): 215–221.

Guillaume, J., and M. Latapy. 2006. Bipartite graphs as models of complex networks. *Physica A: Statistical and Theoretical Physics* 371 (2): 795–813.

Kou, L. T., L. J. Stockmeyer, and C. K. Wong. 1978. Covering edges by cliques with regard to keyword conflicts and intersection graphs. *Communications of the ACM* 21 (2): 135–139.

Latapy, M., C. Magnien, and N. D. Vecchio. 2008. Basic notions for the analysis of large two-mode networks. *Social Networks* 30 (1): 31–48.

Lattanzi, S., and D. Sivakumar. 2009. Affiliation networks. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, 427–434. Bethesda, MD, USA: ACM.

Le Fessant, F., S. Handurukande, A. Kermarrec, and L. Massouli. 2005. Clustering in Peer-to-Peer file sharing workloads. In *Peer-to-Peer Systems III*, 217–226. LNCS 3279.

Leskovec, J., J. Kleinberg, and C. Faloutsos. 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 177–187. Chicago, Illinois, USA: ACM.

Luce, R. D. 1950. Connectivity and generalized cliques in sociometric group structure. *Psychometrika* 15 (2): 169–190.

McClosky, B., and I. V. Hicks. 2009. Combinatorial algorithms for the maximum *k*-plex problem. `http://www.caam.rice.edu/~bjm4/CombiOptPaper.pdf`. Accessed Oct. 29, 2009.

Mokken, R. J. 1979. Cliques, clubs and clans. *Quality and Quantity* 13:161–173.

Moody, J., and D. R. White. 2003. Structural cohesion and embeddedness: A hierarchical concept of social groups. *American Sociological Review* 68 (1): 103–127.

Moser, H., R. Niedermeier, and M. Sorge. 2009. Algorithms and experiments for clique relaxations – Finding maximum s-Plexes. In *Proceedings of the 8th International Symposium on Experimental Algorithms*, 233–244. LNCS 5526.

Newman, M. E. J., S. H. Strogatz, and D. J. Watts. 2001. Random graphs with arbitrary degree distributions and their applications. *Physical Review E* 64 (2): 026118.

Newman, M. E. J., D. J. Watts, and S. H. Strogatz. 2002. Random graph models of social networks. *Proceedings of the National Academy of Sciences of the United States of America* 99 (Suppl 1): 2566–2572.

Östergård, P. R. J. 2002. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics* 120 (1-3): 197–207.

Seidman, S. B., and B. L. Foster. 1978. A graph-theoretic generalization of the clique concept. *Journal of Mathematical Sociology* 6:139–154.

Wasserman, S., and K. Faust. 1994. *Social network analysis: Methods and applications*. Cambridge University Press.

Watts, D. J., and S. H. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature* 393 (6684): 440–442.

Willinger, W., D. Alderson, and J. C. Doyle. 2009. Mathematics and the internet: A source of enormous confusion and great potential. *Notices of the American Mathematical Society* 56 (5): 586–599.

Zheleva, E., H. Sharara, and L. Getoor. 2009. Co-evolution of social and affiliation networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1007–1016. Paris, France: ACM.

AUTHOR BIOGRAPHY

Brian Cloteaux is a computer scientist in the Mathematical and Computational Sciences division at the National Institute of Standards and Technology. He holds a PhD degree in Computer Science from New Mexico State University. His email address is `<brian.cloteaux@nist.gov>`.