Identifying Missing Node Information in Social Networks*

Ron Eyal[1]**, Sarit Kraus**[1] **and Avi Rosenfeld**[2]
[1]Department of Computer Science, Bar-Ilan University Ramat-Gan, Israel 92500
[2]Department of Industrial Engineering, Jerusalem College of Technology, Jerusalem 91160, Israel
Email: eyalro@macs.biu.ac.il, sarit@cs.biu.ac.il, rosenfa@jct.ac.il

Abstract

In recent years, social networks have surged in popularity as one of the main applications of the Internet. This has generated great interest in researching these networks by various fields in the scientific community. One key aspect of social network research is identifying important missing information which is not explicitly represented in the network, or is not visible to all. To date, this line of research typically focused on what connections were missing between nodes, or what is termed the "Missing Link Problem". This paper introduces a new *Missing Nodes Identification* problem where missing members in the social network structure must be identified. Towards solving this problem, we present an approach based on clustering algorithms combined with measures from missing link research. We show that this approach has beneficial results in the missing nodes identification process and we measure its performance in several different scenarios.

Introduction

Social Networks, which enable people to share information and interact with each other, have become a key Internet application in recent years. These networks are typically formally represented as graphs where nodes represent people and edges represent some type of connection between these people (Liben-Nowell and Kleinberg 2007), such as friendship or common interests. Examples of these social networks include popular websites such as Facebook (www.facebook.com) and Myspace (www.myspace.com).

Because of their ubiquity and importance, various aspects of these networks have been studied. One important factor that is often studied is the structure of these networks (Clauset, Moore, and Newman 2008; Fortunato 2010; Leroy, Cambazoglu, and Bonchi 2010; Liben-Nowell and Kleinberg 2007; Porter, Onnela, and Mucha 2009). Previously, a "missing link problem" (Liben-Nowell and Kleinberg 2007; Clauset, Moore, and Newman 2008) was defined as attempting to locate which connections (edges) will soon exist between nodes. In this problem, the nodes of the network are known, and unknown links are derived from existing network information, including complete node information. In contrast, we consider a new *Missing Nodes Identification* problem which attempts to locate and identify missing nodes within the network. This problem is significantly more difficult than the previously studied missing link problem as both the nodes and their edges are not known with certainty.

To understand the importance of the missing nodes identification problem we introduce, please consider the following example. A hypothetical company, Social Games Inc., is running an online gaming service within Facebook. Many Facebook members are subscribers of this company's services, yet it would like to expand its customer base. As a service provider, Social Games maintains a network of users, which is a subset of the group of Facebook users, and the links between these users. The users of Facebook which are not members of the service are not visible to their systems. Social Games Inc. would like to discover these Facebook nodes, and try to lure them into joining their service. The company thus faces the missing nodes identification problem. By solving this problem, Social Games Inc. could improve its advertising techniques and aim at the specific users which haven't subscribed to their service yet.

The above example exemplifies just one possible application of the missing nodes identification problem. In addition to entertainment applications, commercial companies might want to identify new customers based on solutions to this problem. These nodes might represent missing persons that are sought after by family members or people wanted by the police as suspects in a crime. As a result, solving the missing nodes identification problem can be of considerable importance.

We focus on a specific variation of the missing nodes problem where the missing nodes requiring identification are "friends" of known nodes. An unrecognized friend is associated with a "placeholder" node to indicate the existence of this missing friend. Thus, a given missing node may be associated with several "placeholder" nodes, one for each friend of this missing node. We assume that tools such as image recognition software or automated text analysis can be used to aid in generating placeholder nodes. For example, a known user might have many pictures with the same unknown person, or another user might constantly blog about a family member who is currently not a member of the net-

*This research is based on work supported in part by MAFAT. Sarit Kraus is also affiliated with UMIACS.

work. Image recognition or text mining tools can be employed on this and all nodes in the social network to obtain indications of the existence of a set of missing nodes. Placeholders can then be used to indicate where these missing nodes exist. However, it is likely that many of these placeholders are in fact the same person. Thus, our focus is on solving the identification of the missing nodes, or given a set of these placeholders, which placeholders do in fact represent the same person.

In this paper, we present a general method of solving this problem by using a spectral clustering algorithm previously considered only for other problems (Ng, Jordan, and Weiss 2001; Almog, Goldberger, and Shavitt 2008). One key issue in applying the general clustering algorithm is what specific measure should be used for helping identify similar nodes to be clustered together. Towards solving this point, we present five measures for judging node similarity. One of these measures is the Gaussian Distance Measure, typically used over Euclidean spaces in spectral clustering (Ng, Jordan, and Weiss 2001), while the other four measures are non-Euclidean measures which have been adapted from a related missing link problem (Liben-Nowell and Kleinberg 2007). We found that these measures helped solve the missing nodes identification problem significantly better than the base Gaussian measure. We begin this study by further describing these measures and the spectral clustering algorithm.

Related Work

In solving the missing nodes identification problem, this paper uses variations of two existing research areas – spectral clustering algorithms and metrics built for the missing edge problem. The spectral clustering algorithm of Ng, Jordan and Weiss (Ng, Jordan, and Weiss 2001) is a well documented and accepted algorithm, with applications in many fields including statistics, computer science, biology, social sciences and psychology (von Luxburg 2007). Most similar to our paper is the previous work by Almog et. al, using spectral clustering to unite missing nodes in the Internet (Almog, Goldberger, and Shavitt 2008). While the basis of their solution was also Spectral Clustering, they focused on a different problem - how to identify unknown unresponsive router nodes in the Internet. The differences in the problem settings generate different graph structures and locations of the missing nodes in it. Perhaps a more important difference is the availability of an inherent distance function between internet nodes in the form of time delay, making the specifics of their solution not applicable to the missing nodes identification problem we consider for social networks. Thus, while spectral clustering algorithms have been applied to many areas, its use in the missing nodes identification problem has not been previously considered and cannot be directly applied from previous works. Consequently, our first goal was to formally describe this new problem to facilitate study as to how it may be solved.

The key challenge in applying the spectral clustering algorithm to the missing nodes identification problem, is how to compute the level of similarity between missing nodes, or what Ng, Jordan and Weiss refer to as an *affinity matrix*. Towards calculating this measure, we consider measures developed for a related problem, the *missing link problem*. In the missing link problem there are a set of known nodes, and the goal is to discover which connections, or edges, will be made between nodes. In contrast, in the missing nodes identification problem, even the nodes themselves are not known, making the problem significantly harder. Nonetheless, we propose a solution where unknown nodes are represented as placeholders, after which we use measures created to solve the missing link in order to form the affinity matrix to help solve this significantly harder problem as well.

Various methods have been proposed to solve the missing link problem. Approaches typically attempt to derive which edges are missing by using measures to predict link similarity based on the overall structure of the network. However, these approaches differ as to which computation is best suited for predicting link similarity. For example, Liben-Nowell and Kleinberg (Liben-Nowell and Kleinberg 2007) demonstrate that the measures such as the shortest path between nodes, the Euclidean distance between two nodes, and the number of common neighbors can all be useful. They also considered variations of these measures, such as using an adaptation of Adamic and Adar's measure of the similarity between webpages (Adamic and Adar 2003) and Katz's calculation for shortest path information that counts short paths more heavily (Katz 1953) than the simpler shortest path information. After formally describing the missing nodes identification problem, we detail the spectral clustering algorithm and how these missing link methods can be applied to more accurately solve the missing nodes identification problem.

Formally Defining the Missing Nodes Identification Problem

We formally define the new missing nodes identification problem as follows. Assume that there is a social network $G =< V, E >$ in which $e =< v, u >\in E$ represents interactions between people, or formally, an interaction between $v \in V$ and $u \in V$. Some of the nodes in the network are missing and are not known to the system. We denote the set of missing nodes $V_m \subset V$, and assume that the number of missing nodes $N = |V_m|$ is given. We denote the rest of the nodes as known, i.e., $V_k = V \setminus V_m$ and the set of known edges is $E_k = \{< v, u > \mid v, u \in V_k, < v, u >\in E\}$.

Towards identifying the missing nodes, we focus on the visible part of the network, $G_v =< V_v, E_v >$, that is known. In this network, each of the missing nodes is replaced by a set of placeholders. Formally, we define a set V_p of placeholders and a set E_p for the associated edges. For each missing node $v \in V_m$ and an edge $< v, u >\in E$, $u \in V_k$, we add a placeholder for v as v' to V_p and for the original edge $< v, u >$ we add a placeholder for $< v', u >$ to E_p. We denote the origin of $v' \in V_p$ with $o(v')$. Putting all of these components together, $V_v = V_k \cup V_p$ and $E_v = E_k \cup E_p$. The problem is that for a given missing node v there may be many placeholders in V_p. The challenge is to try to determine which of the placeholders are associated with v. This will allow us to reconstruct the original social network G.

Formally, we define the missing nodes identification problem as: given a known network $< V_k, E_k >$, a visible network $G_v =< V_v, E_v >$ and the number of missing nodes N, divide the nodes of $V_v \setminus V_k$ to N disjoint sets $V_{v_1}, \ldots, V_{v_N}$ such that $V_{v_i} \subseteq V_p$ are all the placeholders of $v_i \in V_m$.

To better understand this formalization, consider the following example. Assume Alice is a Facebook member, and thus is one of the visible nodes in V_v. She has many social links within the network, but her cousin Bob is not a member of Facebook. Bob is represented by a missing node $w \in V_m$. From analysis of text in her profile we might find phrases like "my long lost cousin Bob", indicating the existence of the missing node representing Bob. Alternately, from examining pictures from Alice's profile pictures, an image recognition software package identifies 10 pictures of Alice with an unknown male person, resulting in a different indication of the existence of a missing node. Each indication is represented by a placeholder node $v' \in V_p$ and a link $(u, v') \in E_p$, where u is the known node (e.g., Alice) which contains the indication. By solving the missing nodes identification problem we aim to identify which of these indications point to the same missing node, representing Bob, and which represent other missing nodes.

The Spectral Clustering Based Algorithm

Our proposed solution is to use a spectral clustering based algorithm as follows: We first cluster the placeholder nodes, thus creating disjoint groups of placeholders, which hopefully have a high chance of representing the same missing node. We then classify the placeholders according to their clusters and unite all the placeholders in each cluster to a predicted missing node that represents that cluster, in hopes of creating a graph that is as similar as possible to the original graph G. In this section we briefly introduce spectral clustering and focus on how we apply it to the missing nodes identification problem.

Spectral clustering is a general algorithm used to cluster data samples using a certain similarity between them. This algorithm accepts as its input a set of sample coordinates in a multi-dimensional space, described by a matrix S. In its original form, the algorithm constructs an **affinity matrix** which describes the affinity between each pair of samples based on the Euclidean distance on the multi-dimensional space. While the reader is encouraged to review the algorithm in its entirety (Ng, Jordan, and Weiss 2001), a brief description of this algorithm is as follows:

1. Define s_i to be the coordinates of every sample i in the multi-dimensional space and calculate an affinity matrix $\mathbf{A} \in \Re^{M \times M}$, where M is the number of samples. A defines the affinity between all pairs of samples (i,j) using the Gaussian distance function:

   $$A_{ij} = exp(-||s_i - s_j||^2/2\sigma^2)$$

   The parameter σ is used to tune the function to only consider points that are closer than a certain threshold.

2. Define D to be the diagonal matrix whose (i,i) element is the sum of A's i-th row, and construct the matrix L:

   $$L = D^{-1/2}AD^{-1/2}$$

3. Find x_1,x_2,...,x_N, the N largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix X = $[x_1 x_2 ... x_N]$ $\in \Re^{M \times N}$ by stacking the eigenvectors in columns.

4. Form the matrix Y from X by renormalizing each of X's rows to have unit length.

5. Treating each row of Y as a point in $\Re^N$, cluster them into N clusters via K-means or any other clustering algorithm.

6. Finally, assign the original sample i to cluster j if and only if row i of the matrix Y was assigned to cluster j.

The key to the success of this algorithm is constructing the *affinity matrix* in the first step of the algorithm as accurately as possible. Note that this algorithm assumes data nodes residing in some Euclidian space and thus defines affinity between samples accordingly. However, in the case of social network graphs, it is very difficult to embed the nodes of the graph in a Euclidean space as defined by the original algorithm. It is unclear if such a space can be defined in a way which represents the actual distance between nodes in the graph. To understand this difficulty, consider the triangle inequality which is one of the basic characteristics of a Euclidean space. This inequality may not hold for many reasonable distance metrics between nodes. For instance, consider a distance metric that incorporates the number of common neighbors between two nodes. While nodes a and b may not have any common neighbors, causing their distance to be infinite, they both may have common neighbors with node c. This common neighbors measure for this example does not obey the triangle inequality because under this measure d(a,b) > d(a,c) + d(c,b). In general, defining a space which obeys the triangle inequality is difficult because it requires an examination of more than two nodes at once when defining the distances. One distance metric which obeys this inequality is the shortest path length between two nodes, which we define later. Nonetheless, as we have found, even this measure does not necessarily yield the best results.

In order to bypass the Euclidean space difficulty, we have decided to alter the first step of the algorithm and to define direct measures for building the affinity matrix A, rather than using the Euclidean space. In order to apply this algorithm to missing nodes identification, we need to address how this measure can be calculated to represent affinity between nodes in a social network. Due to the complexity of social networks, and the need to compute this measure in a multi-dimensional space, calculating this measure is far from trivial. We considered several different methods, most of which have been proven to be useful when solving the missing links problem in social networks. We have empirically compared these methods with a Euclidean method described in the original algorithm. These methods are discussed in detail in the next section.

Incorporating Missing Link Measures

We propose that affinity measures be constructed based on general graph measures or previously developed measures for the related missing edge problem (Liben-Nowell and Kleinberg 2007; Adamic and Adar 2003; Katz 1953). Specifically, we discuss how five such measures can be potentially applied to the spectral clustering algorithm (Ng, Jordan, and Weiss 2001):

1. **Gaussian Distance:** Define D_{ij} to be the length of the shortest path between nodes i,j. Define D_i to be the vector of the length of shortest paths from node i to all other nodes. Calculate A_{ij} as in step 1 of the original spectral clustering algorithm:
   $A_{ij} = exp(-||D_i - D_j||^2/2\sigma^2)$
2. **Graph Distance:**
   $A_{ij} = 1/(\text{length of the shortest path between nodes } i, j)^2$
3. **Relative Common Friends:**
   $A_{ij} = |\Gamma(i) \bigcap \Gamma(j)|/min(|\Gamma(i)|, |\Gamma(j)|)$
   where $\Gamma(i)$ is defined as the group of neighbors of node i in the network graph.
4. **Adamic / Adar:**
   $A_{ij} = \sum_{k \in \Gamma(i) \bigcap \Gamma(j)} 1/log(|\Gamma(k)|)$
5. **Katz Beta:**
   $A_{ij} = \sum_{k=1}^{\infty} \beta^k \cdot$ (number of paths between i and j of length exactly k)

All five of these measures present possible similarity values for creating the affinity matrix, A_{ij}, in the spectral clustering algorithm. The **Gaussian Distance** is based on the standard distance measure often used in spectral clustering (von Luxburg 2007). In our experiments the value for σ was set to 1. The **Graph Distance** measure is the inverse square between two points i and j, here representing two nodes in the social network. This measure presents an alternative to the original **Gaussian Distance** used to measure Euclidean distances, and is not inspired by the missing link problem. In contrast, the next three measures were directly inspired by this literature. The **Relative Common Friends** measure checks the number of neighbors nodes i and j have in common ($|\Gamma(i) \bigcap \Gamma(j)|$). We divide by $min(|\Gamma(i)|, |\Gamma(j)|)$ in order to avoid biases towards nodes with a very large number of neighbors. Similarly, the **Adamic / Adar** measure also incorporates the common neighbor measure ($\Gamma(i) \bigcap \Gamma(j)$), checking the overall connectivity of each common neighbor to other nodes in the graph and giving more weight to common neighbors who are less connected. Since the nodes that act as placeholders for missing nodes only have one neighbor each, the common neighbors and the Adamic/Adar measures do not represent these nodes well. Therefore, for these measures only, we also consider them to be connected to their neighbor's neighbors. Last, the Katz measure (Katz 1953) directly sums the number of paths between i and j, using a parameter β which is exponentially damped by the length of each path. Similar to common neighbor measure, this measure also stresses shorter paths more heavily. Finally, we considered a baseline **Random Assignment** algorithm that assigned each placeholder to a random cluster and represents the most naive of assignment algorithms.

Finding Data Clusters

As opposed to general clustering problems where all the data must be clustered in an unsupervised manner, in our case most of the nodes are known and the challenge is finding a correct clustering for the placeholders. Despite this, the affinity between the known nodes and the placeholders contains important information which should be utilized. For this reason, we have decided to embed all the nodes in the affinity matrix. In other words, the affinity matrix represents the affinity between **each pair of nodes in** G_v. In step 5 of the algorithm, we do not cluster all the nodes, because the known nodes need not be identified. Instead, we cluster only the rows of Y which match the placeholder nodes. Notice that the information obtained from the known nodes in the embedding process is still present in the matrix Y in the form of the coordinates matching the place-holders, but the known nodes themselves are simply removed from the final step of the clustering process. In this manner, each cluster output can represent a missing node from the original graph.

As input, the spectral clustering algorithm typically requires that the number of clusters, equal to the number of missing nodes (N), be known in advance. While we have focused on a problem setting where the number of clusters is assumed to be known, in many real-world cases this assumption may be false. For example, assume once again that the fictitious company Social Games Inc. is looking for more subscribers by using the clustering approach we present in order to identify missing nodes. The base assumption of the algorithm requires us to know **exactly** how many potential new subscribers exist in the social network, something which is hardly realistic for Social Games Inc. to know. To address this shortcoming, we propose the following method for estimating the number of clusters: Let d be the average degree of a node in V_v. The expected number of clusters is then $|V_p|/d$. This estimation is based on the assumption that the predicted nodes should have the same average degree as the nodes in the visible graph V_v. As we show in the results section, this estimation leads to similar results when compared to running the algorithm with a known number of clusters.

Data Preparation

The goal of this research is to study which of the above measures will help best solve the missing nodes identification problem. To empirically study this point, we have used a previously developed social networking dataset - the Facebook MHRW dataset (Gjoka et al. 2010). This dataset contains structural information sampled from Facebook, containing over 900,000 nodes and the links between them. For each node certain social characteristics are stored, such as network membership within Facebook. These networks incluse academic networks (e.g. Harvard University), corporate networks (e.g. workers in AIG), geographical networks (e.g. members from Idaho), or people who share similar interests (e.g. love of chocolate). All nodes and networks are stored as numbers without any indication of their true identity, ensuring the privacy of people's data.

The main challenge we had to address in dealing with a dataset of this size was implementing and testing the spectral clustering algorithm within a tractable period. In order to create more tractable datasets[1], we considered two methods of creating subsets of the Facebook data (Gjoka et al. 2010). In the first method, we created a subset based on naturally occurring similarities between nodes according to users' network membership characteristics within the social network. Each subset was created by sampling all the nodes in a specific user network and the links between them. Nodes with only one link or no links at all were removed. The advantage to this method of creating the subsets is that there is a higher chance of affiliation between the nodes in the user network in comparison to random nodes selected from the entire social network. However, the disadvantage is that the nodes composing the user network may not be completely connected. In fact, the subgraph of nodes that are part of a specific user network may be very sparse. In contrast, for the second method of creating the subset, we began with the entire dataset, and extracted a subset based on a BFS walk starting from a random node in the dataset. Here no previous information about the social network is necessary, but the BFS generated subset may not accurately represent the actual topology of the entire network.

In order to synthesize the missing nodes within these two subsets, we randomly marked N nodes as the missing nodes, V_m. We then removed these nodes from the network, and replaced each link (v,u) between v $\in V_m$ and u $\in V_k$ with a placeholder node v' $\in V_p$ and a link (v',u) $\in E_p$. The resulting network, G_v is the visible network used as input to the missing nodes algorithm.

Results

Overall, we found that the four affinity measures inspired by work from the missing link problem allowed us to solve the missing nodes identification problem significantly better than the baseline Random Assignment algorithm or even the Gaussian Measure in the basic Spectral Clustering algorithm. The results presented in this section suggest that better affinity measures, specifically the Graph Distance and Common Neighbor measures, were most successful in facilitating better solutions.

To measure the accuracy of different affinity measures, we ran the clustering algorithm with each of the two datasets and used the *purity measure* often used in evaluating clustering algorithms (Ng, Jordan, and Weiss 2001). The purity measure is calculated in the following manner:

1. Classify each cluster according to the true classification of the majority of samples in that cluster. In our case, we classify each cluster according to the most frequent true original node v $\in V_m$ of the placeholder nodes in that cluster.
2. Count the number of correctly classified samples in all clusters and divide by the number of samples. In our case the number of samples (nodes) that are classified is $|V_p|$.

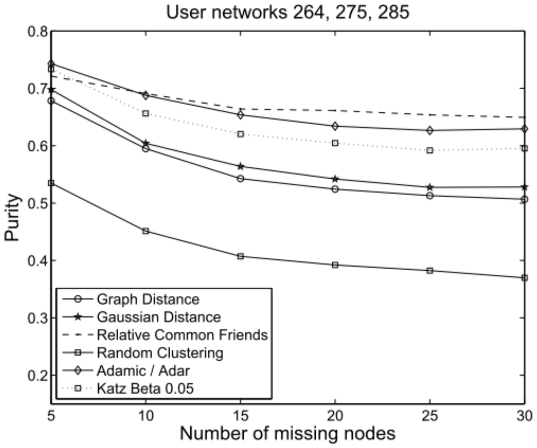

Figure 1: Comparing the Clustering Accuracy (Purity) of Five Affinity Measures in first Facebook data subset

Formally, purity is defined as:

$$purity(C) = \frac{1}{|V_p|}\sum_k max_{v_j \in V_m} |c_k \cap \{v' \in V_p \mid o(v') = v_j\}|$$

Where c_k is defined as the set of placeholders which were assigned to cluster k. Note that as the number of missing nodes increases, correct clustering becomes more difficult, as there are more possible original nodes for each placeholder. As our results show, the purity indeed decreases as the number of missing nodes increases.

First, we considered the results from the first dataset with full subsets of the Facebook data. To create these results, we considered three user networks within Facebook, marked by the numbers 264, 275 and 284 in our dataset. As described above, for each of these user networks we began by creating a graph consisting of all of the users in the dataset who are network members and the links between them. From each graph we removed the nodes with only one link or no links at all. After this process, the network 264 graph consisted of 2301 nodes and 3200 edges. The graph of network 275 contained 2087 nodes and 2905 edges, and the graph of network 285 contained 2754 nodes and 4118 edges. For each graph we then randomly removed sets of 5, 10, 15, 20, 25 and 30 nodes from the three networks. Each trial involved selecting a different group of random points. For each network and each number of missing nodes we ran the clustering algorithm with the five previously described affinity measures plus the random clustering baseline.

Figure 1 displays the results from this first set of data. To ensure the validity of the results, each point represents a total of 30 iterations (10 iterations over each user network graph). Note that the Random Clustering algorithm, as expected, clearly performs the worst, with a purity measure often being 30% worse than that of the algorithms based on spectral clustering. This indicates the overall effectiveness of this approach in solving the missing nodes identification problem. Within the five affinity measures we considered, the basic Gaussian measure clearly did not perform well in comparison to others, again demonstrating the need for considering

[1] We have not yet been successful in creating algorithms that can process all 900,000 nodes in the dataset within a tractable time. Nonetheless, we do report on methods that can deal with relatively large datasets with up to 40,000 nodes.

measures specifically geared for this problem. Specifically, the Relative Common Friends measure achieved the best purity result.

We also considered a second dataset created with a BFS extraction from the entire Facebook corpus, creating 15 networks of 2000 nodes each. We again ran the clustering algorithms with the same five measures plus the random baseline, and randomly removed 5 to 30 nodes in the same fashion. The results of these experiments are presented in Figure 2.

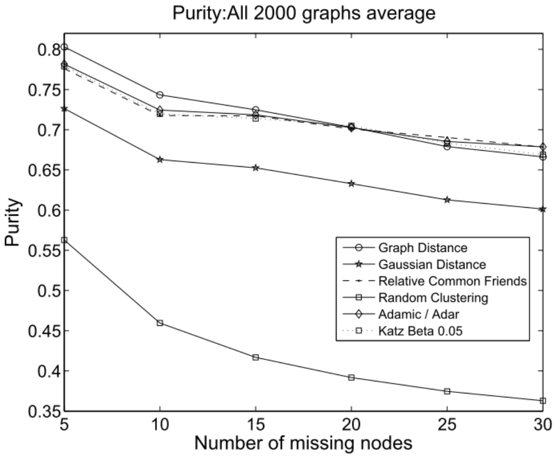

Figure 2: Comparing the Clustering Accuracy (Purity) of Five Affinity Measures in second Facebook data subset

Note that in these results, the Random Clustering algorithm again performed significantly worse, and the base Gaussian measure performed the worst of the 5 variations based on spectral clustering. We did not find a significant difference between the remaining 3 measures. One possible reason why the Common Neighbor measure did worse than the Shortest Graph distance in this dataset is as follows. The second dataset was engineered with a BFS search, and was thus guaranteed to have connectivity between all nodes before removing the missing nodes. After removing the missing nodes, there was still a high probability of remaining with a connected graph. In contrast, the natural topology of the first dataset included data which was sparse (e.g. had very few edges), such that removing the missing nodes usually resulted in the network losing its connectivity. As a result, the graph distance measure in the first dataset was often not a good measure, as the distance between unconnected elements was infinitely large. Thus, we posit that measures focused on smaller areas, such as the Common Neighbors measure, were more effective.

We also considered how these algorithms scale in the missing nodes identification problem. We constructed a dataset of 5000 nodes, again taken as a subset of the entire Facebook corpus through BFS search. We then ran the same 5 spectral analysis algorithms plus the random baseline on 10 randomly selected nodes. The results from this experiment are found in Figure 3. Note that, as was the case in Figure 1, the Graph Distance measure yielded the highest clustering accuracy (purity) with all measures being significantly better than the baseline random classifier.

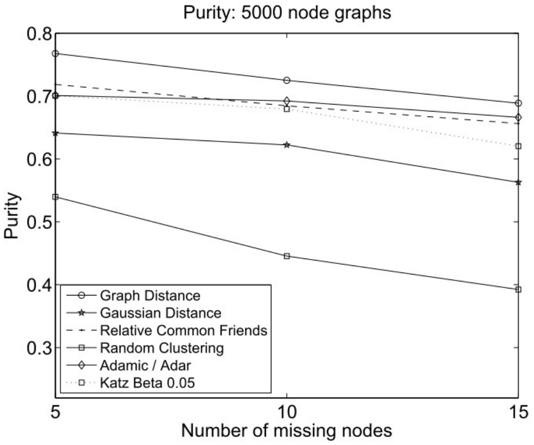

Figure 3: Considering how Performance (Purity) is Effected by Scaling to a 5000 Node Dataset

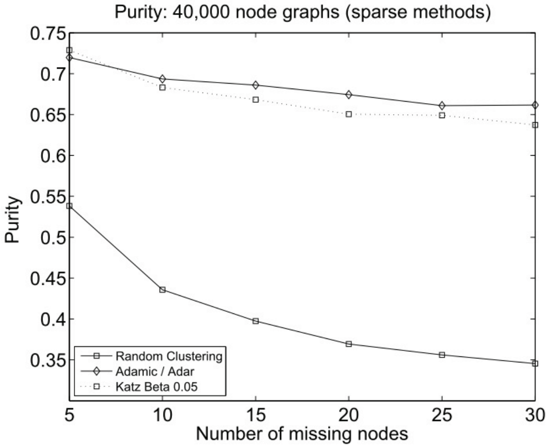

Figure 4: Considering how Larger-Scale Datasets (40,000 nodes) effects the Clustering Accuracy (Purity) in the second Facebook data subset

In order to scale our algorithm to larger graphs, we have employed sparse matrix representations of the graph and the resulting affinity matrices. In these sparse representations, only non-zero values are saved. While the initial graph structure is very sparse, some of the affinity measures, such as Graph Distance and Katz Beta, induce dense affinity matrices. These methods must be approximated in order to maintain a sparse matrix. For instance, we have altered the Katz Beta method to only include paths up to length 4, thus creating a sparse affinity matrix. The Adamic / Adar method required no change since the affinity matrix generated by this method is sparse enough. Figure 4 displays the success of scaling these two methods in comparison to a random clustering. These methods were tested on datasets with 40,000 nodes, again generated through BFS of the MHRW dataset.

Figure 5 illustrates the results of estimating the number of missing nodes, as described above, in comparison to working with a known number of N clusters in advance. Specifically, we use our implementation of spectral clustering using

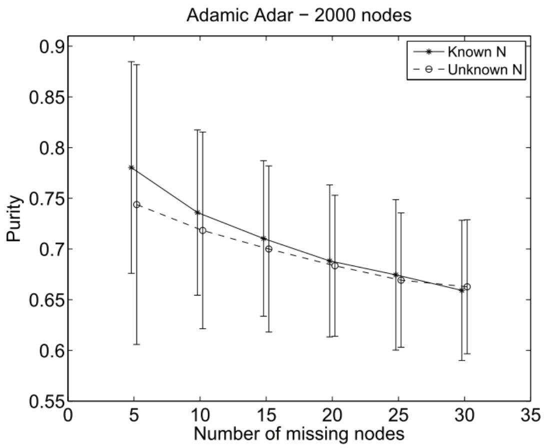

Figure 5: Comparing the Purity when the Number of N Nodes Requiring Identification is Known versus when an Estimate is Used

the Adamic/Adar measure and input both a known value for N and an estimation for N derived from the average degree for nodes in the visible network, V_v. Note that, in general, using the estimation yielded only slightly worse accuracy (purity). In fact, in some cases using the estimator for the number of missing nodes produced even better results. We found that this typically occurred when the estimate was in fact an overestimate of the actual value for N. This is based on the definition of purity that determines the classification of each cluster according to the majority of samples within that cluster. Thus, having an overestimate can actually increase the purity measure of each cluster and, as an implication, increase the overall purity across the entire network. While we present a comparison between using an estimate for N and the actual value for N within the Adamic/Adar measure, we similarly observed that relaxing the requirement for using the actual value for N does not significantly degrade the algorithms' performance for the other measures as well.

Conclusions and Future Work

In this paper we introduce the missing nodes identification problem, where nodes representing people must be identified within social networks. This problem has not been previously considered and is likely to be of importance within a wide variety of applications. For example, this work is important in entertainment domains where people wish to find new social contacts outside of their known network; commercial applications whereby new customers can be identified; and security applications where police enforcement members can find wanted individuals.

This paper makes several contributions to addressing this new problem. First, we formally present this problem. Second, based on this formalization we describe how the existing spectral clustering algorithm can be applied to solve this problem (Ng, Jordan, and Weiss 2001). However, one key issue in implementing this algorithm for the missing nodes identification problem is the similarity between nodes in the network, a measure needed to form the affinity matrix within this algorithm. Third, we present one possible solution inspired by previous work (Almog, Goldberger, and Shavitt 2008) which focuses on a Gaussian measure to measure possible node similarity. Last, we also present four additional affinity measures based on Missing Edge literature (Liben-Nowell and Kleinberg 2007). We empirically compared these five possibilities within two types of problem subsets from a Facebook repository (Gjoka et al. 2010). We have shown that all five methods provide for good solutions in comparison to a random clustering baseline solution. In addition, we have found that the measures based on the missing link problem typically yielded significantly improved performance and aid in better solving this problem.

For future work several directions are possible. This work proposes a simple method to estimate the number of missing nodes. In the short term, we have begun researching how assumptions on the number of missing nodes can be further relaxed, specifically by using lazy clustering algorithms which do not need advance knowledge of this value. We also hope to study what new metrics can be introduced to further improve the accuracy of the algorithms presented here. Finally, we plan to study additional datasets and domains.

References

Adamic, L. A., and Adar, E. 2003. Friends and neighbors on the web. *Social Networks* 25(3):211–230.

Almog, A.; Goldberger, J.; and Shavitt, Y. 2008. Unifying unknown nodes in the internet graph using semisupervised spectral clustering. *Data Mining Workshops, International Conference on* 174–183.

Clauset, A.; Moore, C.; and Newman, M. E. J. 2008. Hierarchical structure and the prediction of missing links in networks. *Nature* 453(7191):98–101.

Fortunato, S. 2010. Community detection in graphs. *Physics Reports* 486(3-5):75 – 174.

Gjoka, M.; Kurant, M.; Butts, C. T.; and Markopoulou, A. 2010. Walking in Facebook: A Case Study of Unbiased Sampling of OSNs. In *Proceedings of IEEE INFOCOM '10*.

Katz, L. 1953. A new status index derived from sociometric analysis. *Psychometrika* 18(1):39–43.

Leroy, V.; Cambazoglu, B. B.; and Bonchi, F. 2010. Cold start link prediction. *SIGKDD 2010*.

Liben-Nowell, D., and Kleinberg, J. 2007. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology* 58(7):1019–1031.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, 849–856. MIT Press.

Porter, M. A.; Onnela, J.-P.; and Mucha, P. J. 2009. Communities in networks. *Notices of the American Mathematical Society* 56(9):1082–1097.

von Luxburg, U. 2007. A tutorial on spectral clustering. *Statistics and Computing* 17(4):395–416.