An Analysis Tool for Social Networks and Other Relational Data

Senior Design 2009-2010

Robert McMullen
wmcmull@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

Lyle Ungar
ungar@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

Shawndra Hill
shawndra@wharton.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

ABSTRACT

In recent years, social networking has become an integral part of life for students, professionals, and the general population alike. With different sites catering to different types of people, there are more and more ways for people to get involved. Social networks have gone beyond a casual hobby and have the potential to shape the ways people go about everyday life.

As these sites grow in popularity, many people have amassed large networks with various types of connections to other users. As a result, while only a small amount of data may be available about a given individual, there is likely a vast amount of information about people they have chosen to be connected to through these websites. And as more people join every day, people’s networks are expanding as more opportunities to establish links come about. This phenomenon is relatively new, so research in the area of social network data is young, but growing quickly.

In order to aid future research on social networks, a software package was developed to allow analysis on network data. It was developed with the intention of being an all-purpose research tool, with the ability to store large amounts of data and do simple analysis on it. Users are able to create networks and output their findings, based on a few simple output options. A custom query language was also developed with the system in order to allow users to examine whole networks at a time and ask network-based questions, rather than focus on the individuals.

1. INTRODUCTION

Social Networks have grown in size and popularity since their inception and continue to see exponential growth. A social network can be defined as any online medium that allows a user to create a profile with public information and a list of users who they are connected to [2]. Users are then allowed to update their personal information and look at their friends within that system. The first social network was developed in 1997, and though a large number of people are already involved, the number of users does not appear to be tapering off. In the last year, users of the popular site Facebook.com have doubled within the past year alone [3] (from 200 million users in April, 2009 to 400 million users in February, 2010). Among them, over half of the users access their account each day.

As social network sites grow in size and popularity, each individual has the opportunity to expand the amount of connections that they make. Once his or her friend, family member, or business associate has a profile, they have the option to establish a link to them. A link, or connection, is an association between two people on a social networking site. Examples of this are a friend on Facebook.com or an instant message between two accounts in Yahoo Messenger.

An individual person’s online profile may contain useful information for a company, but by examining their networks one can potentially garner a more broad understanding of a potential customer or consumer group. An example of this is a company who may want to advertise to close friends of a person who has bought their product online. This draws on the concept of network neighbors, which are customers who are linked through a social network connection to a prior customer [6].

Another way network data can be used is to make predictions about people within a network who do not have publicly available information. Based on the information supplied by others within their network, one could create an algorithm to make assumptions or attempt to draw conclusions about the anonymous user. This relies on homophily, the principle that people who are have more contact with each other have similar personalities, tastes, and desires [10]. However, people may also have connections to people whom they barely know, who have dissimilar tastes.

People use social networks in many different ways, and make connections for many different reasons. Another way to look at connections is to try to attach a tie strength attribute to each link, to attempt to quantify how well two connected people know each other. With this, one could create a network of close friends rather than just looking at every loose connection made. This would be more valuable in making predictions about a given user based on their networks.

To allow for future research in the field of social networks, a software tool for network analysis was built. The software is comprised of three main components: Data input, network creation and analysis, and data output. Data input is allowed from three basic formats, which are converted into user data and link data for the analysis engine.

Once imported, the system allows users to create static networks or generate networks based on connections. These networks can then be stored and used for further analysis. Network creation relies on the concepts of a baseuser and a hop level. A baseuser, or group of baseusers, are root nodes from which the network is built. The hop level determines the diameter of the furthest reaching nodes to be added to the network from the root.
From there, a customized query language allows the user to run statistical tests or generate output files on their networks. Data output can be in the form of built-in tools from the system or output files for further analysis. This document will outline both the functionality and implementation of the system, showing its potential uses and effectiveness.

2. RELATED WORK

Due to the rise in popularity of social networking sites, a wealth of research has been done regarding the connections that people make. Most of this research has been done on the trends in social networks and the different ways that people use them. There have been attempts to create software tools before to help analyze network data, which will be discussed. Finally, a brief look will be taken at the principle of homophily, which will be the first test case when the tool is complete.

2.1 Social Networking Sites

Much of the research on social networks has centered on the different ways that people use a given networking website. For any social network, there is a broad spectrum of reasons to establish a connection between two people. According to a study by Mike Thelwall [12], there are three main classes of social network connections: close friends, acquaintances, and strangers. He also contends that there are certain reasons to invite somebody to be an online “friend.” One of the reasons to initiate a link is out of courtesy, to avoid offending the other person. Another reason is to declare an actual friendship that exists outside of the social networking website. Some users, however, view the significance of an online friendship as minimal or nothing at all. An online friendship may also be created to coordinate offline activities such as basketball or poker. It is easier to write one message about an event to all members of a network than to contact each person individually. Finally, there is still the possibility that people are genuinely online friends [12].

As a result of these different types of friendships and friend requests, it is difficult to discern the true relationship between two people when their relationship is given the blanket term, “friend.” Therefore, it is important to understand the difference between a strong tie and a weak tie [4]. A strong tie would connect two close friends, family members, or teammates. A weak tie may be somebody that was met at a bar, or a fan of a famous sports team. If one can classify the strength of the tie, one can start to infer if somebody is a stranger or a friend, depending on outside factors.

Apart from differences in friends and ties, there are also various ways that people use social networking in the context of their social life [9]. For some, social networking is an active part of how they make plans and interact with their friends and acquaintances. For others, social networking is just a way to stay in touch, or is not a priority at all. It was also found that men and women sometimes have different uses for social networking. On MySpace.com, females were more interested in friendship while males were more interested in dating [12]. On Facebook, some people add friends to maintain offline friendships, while others attempt to make new friends on the website [9].

Much of the research done on social networking focuses on people’s habits, tendencies, and friend types. Some of the research also tried to make predictions about the types of connections between users. For example, Gilbert et al. [4] were able to predict whether a tie between two people was strong or weak with 85% accuracy. Gilbert et al. also discovered that intimacy in their conversations was the highest indicator of a strong connection.

Another interesting facet of social networks is that they become denser as they grow larger [5]. Intuitively, it was assumed that as a network got bigger, the number of degrees (links of a given member) would grow linearly, assuming a constant average degree. However, it was been shown that the average degree increases exponentially as the network expands. This is called the densification power law [5]. Also, as the network grows larger, the average shortest distance between nodes decreases. Researchers now assume that some of the new nodes in the expanding network act as bridges between others, bringing the average distance down. This is also counter intuitive as well, as you would assume with more people, some people would have very loose paths to one another. However, these two phenomena have been experimentally shown in various studies.

2.2 Social Network Analysis Tools

The two main types of social network tools that exist are strategic analysis tools and visualization tools. Strategic analysis tools look at the members and links within a network to draw conclusions or create models. These are analytic tools, which is what our tool will attempt to build upon. Visualization tools simply provide an interface for drawing and examining the network itself. Some of the tools incorporate both, which our tool will do as well.

One previous tool that is widely used is Proximity [8]. Proximity was created by researchers as the University of Massachusetts Amherst for analyzing relational data. It has its own query language called QGraph [8] which locates patterns within imported relational data. It also uses the Java API and Python scripts to generate statistical models. Proximity also accepts XML inputs and allows visualization of the network being analyzed. This tool is similar to what this document describes, and many lessons can be learned from it. By narrowing the focus to social networks, we can create a more targeted tool, while drawing on some of the concepts used in creating Proximity.

Another tool is called InFlow [7], and is another hybrid of data analysis and visual representation. InFlow allows users to examine networks to get information about clusters, centrality, shortest paths, and network density, among other things. It also allows for what-if analysis, which allows users to predict what will happen if they change certain dynamics of the network in question. InFlow is an all-encompassing program, which incorporates similar goals and also provides training. However, the program and training are expensive, whereas our tool will be free for any potential users. It may help by providing an example of how data analysis can be done on social networks.

There is a full department devoted to social network research at the University of Carnegie Mellon, the Center for Computational Analysis of Social and Organizational Systems (CASOS) [1]. CASOS accesses various departments within Carnegie Mellon, including computer science, to conduct research on network dynamics and social networks. In dong their research, they have created tools which help them in their goals. They also have links to various other related tools on their website.
One tool created by CASOS is DyNetML (Dynamic Network Markup Language) [1]. This language allows users to create nodes, links, and attributes to represent a real social network. A similar outside tool is GraphML (Graph Markup Language) [11]. GraphML also allows for graph creation and also provides graphic tools for visualization of the created graphs. GraphML could be another output type for our software tool, and could provide an added dimension to the user interface if interoperability is possible.

The other CASOS tools examined are specific to other projects within the CASOS program. For example, ORA is a tool for finding a “critical member” of a network [1], or a person who is a threat to the network. This allows for examining knowledge flows within a group and task management, but does not relate to making predictions about networks, as our tool will allow. Our tool will draw on the concepts of Proximity and InFlow, while attempting to remain more general than the tools in CASOS and GraphML. We will attempt to create a unique tool which will complement these other software packages to create a new platform for innovative research.

2.3 Homophily

As previously mentioned, homophily is the principle that similar people have more interaction with each other than dissimilar people [10]. As a result, the groups that people surround themselves with consist of others who have similar characteristics. A study by McPherson et al. argues that people who share certain factors and behavioral characteristics are more likely to create these tight knit groups than others. The most prominent factors that link people, and therefore predicts they will interact with each other more, are race and ethnicity [10]. The other significant factors that indicate similarity between people are age, religion, education, occupation and gender.

Another study showed that people who were connected to former customers of a telecommunications company were 3-5 times more likely to adopt the service [6]. This study also showed that statistical models were more accurate when network information was included. The original models included geographic data, prior purchase data, and demographics, but were substantially improved with the addition of social networking data [6]. This study verified that people who are connected via social networks have similar actions in certain situations.

3. SYSTEM MODEL

Our system consists of three major components: (1) importing data, (2) analyzing data, and (3) outputting data. The first portion of this requires putting data into the proper format for the engine to examine it. Analyzing information is the most important part of the software, as it must be easy to extend for any use and be able to examine different combinations of data from various network types. This includes a custom query language in order to enable users to examine networks on the whole rather than individuals. Finally, the data can be exported to CSV files or basic statistical models can be made. All of these components will be contained within a graphical user interface which allows users to change the inputs, outputs, and analysis to be run. Figure 1 is a diagram of the workflow of our system, and Figure 2 is a screenshot of the user interface.

3.1 Importing Data

In order to be useful for other researchers, the software must be effective regardless of the format of the imported data. Every entity with social networking information likely has their own standards for how their information is stored. Two potential ways to store network data are edge lists and matrix representation. Another possible way to represent network data is as a network list. Here, each node is accompanied by a list of the other nodes it is connected to.

All imported data ends up stored as a list of edges and a list of demographic information stored in a database. Con-
version to matrix representation or network lists is available internally for certain data analysis. For example, a network list would be ideal for network creation, but is too large to store for each individual user. With all of these three components able to be created, fast and varied testing can be done, using the strengths and weaknesses of each representation as appropriate.

The tool allows the user to specify the data files containing the network information as well as the demographic information about the individual users. They can then select the format that their network data is in to ensure proper conversion. Figure 3 shows this portion of the user interface. The user then specifies the database where they would like the converted data to be stored. This can be done with a custom connection string or by specifying the host and database name to be created. Figure 4 shows how this is done on the user interface.

The interface allows the user to generate the queries. The different aspects that can be specified with the data conversion.

The tool also allows the user to aggregate all of the search groups into one, to analyze or make predictions on the network as a whole. In this case, all of the individually generated search groups are joined into one aggregate search group. For this case, attributes of the overall network can be compared to attributes of the search group containing the combines friends of the baseusers. This is useful in analyzing subgroups within a network.

There are two ways to specify a network in the software package: explicitly, by providing a static list of users, and implicitly, by specifying a baseuser and a hop level. For the implicit version, the software will search through the link data to generate a list of users for the network, extended out as far as the hop level specifies. This group of users can then be named and stored in the database for easy access later.

The return group, as noted earlier, is the set of users you are examining. Once this group is specified, the search group can either be a static group or a group dynamically created for each user in the return group. In the latter case, the search group will be different for each iteration of the query, using each member of the return group as the baseuser. In predictive queries, this will generate a separate predicted value for each user.

The tool also allows the user to aggregate all of the search groups into one, to analyze or make predictions on the network as a whole. In this case, all of the individually generated search groups are joined into one aggregate search group. For this case, attributes of the overall network can be compared to attributes of the search group containing the combines friends of the baseusers. This is useful in analyzing subgroups within a network.

The user can also impose a list of restrictions on the search group, whether it is a user list or an automatically generated network. The restrictions can be on the link attributes or on the user attributes of the created network. For example, you may want to create a network based on the immediate friends of user 0001. In this case, the baseuser would be user 0001 and the hop level would be 1. But if you were only interested in friends of this user who were between 18 and 25, you could set a restriction based on user attributes to do that. Similarly, you may only want business associates, and a link attribute restriction could eliminate all others.

The interface allows the user to specify the groups they would like to create, and then allows them to be stored in the database. During this process, the user can visualize...
the group to get a general idea of how big the network is and what it looks like. A button on the interface launches a new window which contains a graphic image of the network, which can be saved as an image file. Three examples of this visualization follow, all for the same baseuser. Figure 6 shows a network with hop level 1, Figure 7 shows a network with hop level 2, and Figure 8 shows a network with hop level 3.

Figure 6: Network Visualization with hop level 1

Figure 7: Network Visualization with hop level 2

Figure 8: Network Visualization with hop level 3

Another option available to the user is to aggregate all of the data into one search group. This can be useful in drawing conclusions about a network as a whole, rather than individuals within that network. In doing this, the return group becomes a single entity, with a prediction or analysis made about the combination of all the search groups.

Finally, the user can choose to run the query. The system then takes the user input and performs the specified statistical tests, using the generated search group or search groups to generate the data. As explained in the following section, the output requested is returned to the user or outputted to an external file. The user is then able to look at their generated data, adjust the query, or continue to examine new networks.

3.3 Data Output

Once the user has imported their data and set up the query, there are a few different output options for the returned data. All of the real data and predicted data is stored internally and be outputted to a CSV file for further testing. Obviously, if the main goal of the researcher is simply to obtain the real attributes of the network, they can opt to output to CSV the real attributes of each member of the network, along with their user identifications.

For predictive tests on users, there are three main output types. First, a histogram or boxplot of the predicted values can be generated. An example of this is shown in Figure 9. When testing on a large network, this may be good for summarizing the findings and examining the network as a whole. Second, for individual results, a plot of real attributes vs. predicted attributes can be generated. There is also an option for adding a least-squares regression line to check for linear correlation. In this case, if the real attribute is unavailable, the user is left out of the plot, as there is no way to check for correlation. An example plot of real vs. predicted attributes follows in Figure 10.

Figure 9: Boxplot and Histogram of Predictions

Finally, examining real and predicted values can also be accomplished by outputting all of the real and predicted values to a CSV file. If linear association does not occur, this data can be examined by a variety of program to see what type of association there is. Rather than complicating the tool with all of these different options, the output file is provided for customized analysis. An example of how this would look is in Figure 11.

These output options allow the user to get a general view of their networks attributes as well as look for specific links between predictions and their corresponding real values. The user interface has options for each output option, and opens separate windows with the plots which can be saved as image files. If one chooses to output the results to a CSV file, the filename can be specified in the tool as well. Figure 12 shows how to choose output options on the user interface.

4. SYSTEM IMPLEMENTATION

The main coding of the software is done in Microsoft C-Sharp. This was chosen because of its powerful libraries, enormous amount of support, graphical abilities, and in-
The software utilizes MySQL for database software because it is widely used, supported, and free. The statistical tests done within the engine utilize the R COM, a .NET compatible interface for integrating the R statistics project. This aids with inner calculations as well as generating some of the user’s output options.

The graphical interface is done using C-Sharp Windows forms, with backend C-Sharp coding. The visualization of graphs is done with Glee, a free package from Microsoft. Other output types were considered, including GraphML, but an outside viewer are required to use the files once they are created. Glee was chosen as it could be fully embedded within the system. Details of how specific functionality were implemented follow.

4.1 Data Conversion

Once the filenames and format of the input files is specified, the software pulls the information from the file and puts it into the database. As mentioned earlier, the network data can be in the form of network lists, edge lists, or a matrix representation. Regardless of the original format, the data is transformed into three separate components for storage: Link data, user data, and network data. Each of these is stored in a separate database table with the necessary information.

The database where the data will be stored is specified by the user in the user interface. The user can provide the host name, database name, username and password or simply input their own connection string. Once this is present, any data import will go to the specified database. Also, the same connection information will be used in any network visualization or data analysis. To change the database they want to perform analysis on, the user must simply update the database name field or connection string.

The link data is stored as an edge list with attribute information. For each connection, and edge is added from user 1 to user 2, and if that connection is discovered again from user 2 back to user 1, the link is not added again. This requires some initial overhead for data import, but takes half as much storage space and allows for faster speeds in data analysis, where commands will be run more often. Each edge also has a comma separated list of attributes which is stored within the table.

While the link data table has a static number of columns (user 1, user 2, and attributes), the user data table has the same number of columns as there are available user attributes. Once the table is created, the demographic information is put into the table. Each user is searched for by their unique user identification, which must the same as the identification used to specify links, or else the data analysis would be impossible. Therefore the first row in the user data table is always the identifier. The network table simply contains two sections, the name of the network and the name of the user. Figures 13, 14, 15 show examples of the link data table, user data table, and network data table respectively.

The attributes column of the link data table is not shown, as the dataset did not provide any link data.

4.2 Network Creation and Visualization

The driving force behind the software tool is the ability to define and examine social networks, so naturally the tool must contain the ability to create and store those networks. As previously discussed, the system allows two different ways to specify a network, as a list of users or as a set of baseusers.
and a hop level. If it is a user list, each user is read straight into the network data table with the corresponding network name.

When a baseuser network is created, the system must look at the link data table in order to generate the correct set of people. Within a Network class created in the code, there is a separation between the list of baseusers and the list of members of the network. This is done to help in creating the network, as depending on the hop level most data analysis is done on the root nodes. Once the network is created, all users are treated equally. When the network is stored into the database, it is simply a list of users, and therefore should be named appropriately to keep the distinction possible.

The visualization of graphs is done with Glee, a free package from Microsoft. Other output types were considered, including GraphML and DGML, but an outside viewer are required to use the files once they are created. Glee was chosen as it could be fully embedded within the system, and a picture could be returned directly from the user interface. Before this output, the network is translated into a set of vertices and a set of edges. From this an object called an Adjacency Graph is created, which Glee translates into an image. While GraphML and DGML are not used, support for creating files of these types is available for users who prefer this format. Figures 13, 14, and 15 from earlier in the document show examples of network images generated with GLEE.

### 4.3 Query Language

The query language is the main part of the software system data analysis. When the user sets up a query, the necessary data is pulled from the database tables and generates the chosen output. The following is a Backus-Naur Form description of the grammar for the queries.

```
<query> ::= <query-type> <stat-type> <return-attribute> “of” <return-group> “using” <search-group> “where” <restriction-list>
  <query-type> “predict” | “analyze”
  <stat-type> ::= “mean” | “median” | “min” | “max” | “sum” | “var” | “sd”
  <return-attribute> ::= <attribute>
  <return-group> ::= “userlist” <user-list> | “baseusers”
  <user-list> ::= “hoplevel” <positive-integer> | <network-name>
  <search-group> ::= “userlist” <user-list> | “baseusers hoplevel” <positive-integer> | <network-name>
  <user-list> ::= <user> | “,” <user-list> | <user> | “<”
  <user> ::= != usernames will not be known ahead of time
  <network-name> ::= != network names will not be known ahead of time
  <restriction-list> ::= “<” restriction | “<restric”
  <restriction> ::= “linkatt”<attribute> <comparator> <value> | “useratt” <attribute> <comparator> <value> | “<comparator” <attribute> <comparator> <value> | “==” <attribute> <comparator> <value>
  <value> ::= “”<attribute> “” | “<”<attribute> “<”
  <return-attribute> ::= != available attributes will not be known ahead of time
```

The analysis of data is then done through a combination of a few simple methods that access the actual databases.

- **get_info()**. This takes a user ID as a parameter and returns all of the attributes of that user. It accesses the user data table for this information.
- **get_attribute()**. This is similar to get_info, but also takes an attribute parameter. This returns only a single desired attribute, rather than a list of all of them.
- **get_friends()**. This method creates a Network based on three parameters: a list of user IDs to use as a baseuser, a hop level, and a list of restrictions. The method uses the link data table to find all friends, and validates them against the restrictions before adding them to the network or discarding them. If the hop level is greater than one, it calls itself recursively until it reaches the desired distance.
- **get_link_attributes()**. This uses the link data table to get the link attributes of the connection between two users. It takes two parameters, user 1 and user 2, and returns a null value if the two users are not connected.
- **is_Linked()**. This also uses the link data table to test whether two users are connected. It takes two parameters, user 1 and user 2. It can also take a third parameter which is a link attribute, and will only return true if the link has the specified attribute.
- **validate()**. This method tests a user against a list of restrictions to see if it should be added to a network. It takes three parameters: user 1, user 2, and a list of restrictions. For each restriction, if it is a user attribute restriction it uses the user data table, and if it is a link attribute restriction it uses the link data table to see if the user qualifies.
- **get_columns()**. This gets all of the column names from the user data table, which represent the available attributes for the users in the data set.
is useful especially for populating the dropdown menu of available attributes on the user interface.

When the query is run, the networks are parsed by the tool. If it is simply a user list or a previously created network, the network is simply created with the specified users. If it is a base user, the get_friends method is used to generate the user list, along with the provided restrictions. For each member of the search group, get_info or get_attribute is then called, depending on whether the user asked for all information or is just looking for a specific attribute.

All of the data about the search group network is then fed to the R COM interface. For this, it is translated into a vector. The chosen statistical test is then run on the vector, and the results are stored internally. Finally, based on the user’s output options, the information is returned to the user.

4.4 Data Output

For data output, the primary option for users is to output their finding to a file. The desired filename can be specified on the user interface. Each line of the output file will be dedicated to a user from the return group, and will contain the desired information. For analysis queries, the information will be the users information gathered from the user data table. For prediction queries, the information will be the users real attributes followed by the predicted value of the attribute based on the search group.

There are also three built-in display options for the generated data. The user can choose to view their predicted results in a histogram, boxplot, or scatterplot of real vs. predicted values. These plots are generated from the vector strings in the R COM interface, and appear as separate windows. For the scatterplot, the user can also add a least squares model to the plot with the line equation and correlation output below. These options can be chosen on the user interface, as seen earlier in Figure 12.

5. SYSTEM PERFORMANCE

5.1 Functionality

All of the discussed features for data import, analysis, and output are functional within the software package, and the user interface allows all aspects to be explored. To test the functionality of each piece, a test dataset was created with one hundred users. The users were given demographic information, and network data was created for the three data formats: network list, edge list, and matrix representation. The data was read into a MySQL database on a local machine.

For the sample dataset, the system is fast and reliable. The visualization works, the database lookups are all accurate, and running queries is fast. Obviously, as you increase the number of users in the return group and the hop level of the search group, more data lookups are necessary and the system does not move as fast. The following summarizes the effects on the sample dataset. In the first half of the table, a query is run on a single user. For the second part, all 100 users are in the return group, each with between four and seven friends. When the hop level is three, each user has approximately 70 to 80 people in their search group. From this, we can see that once the system reaches hop level of four, it becomes very slow. However, for anything up to hop level of three, the system runs fast for a given user. Even on 100 users with approximately 70 friends, it takes well under a minute to gather all of the data and return it. The average user on Facebook.com has 130 friends [5], so we can be confident our system will run in under one second for the average user at hop level one.

The reason for the slowdown at hop level four is not the number of friends, but rather the iterations of calling the get_friends method within the system. Therefore, we are confident the system will continue to be fast on larger datasets when the hop level is three or less. A distance of three is very significant, and it is unlikely that researchers will be interested in a hop level greater than three. However, as we will discuss now, the issue of importing a large dataset is complex and should be improved upon.

5.2 Scalability

To test how the network analysis tool works on a real, large group of network files, data was provided by Shawn-dra Hill, a professor at the Wharton School of Business at the University of Pennsylvania. The data is from a social networking information company, and consists of user demographic information and large network lists. The data consists of approximately 20,000,000 users and their network information. The network lists are broken into files with 2,000,000 users represented in each file. Each of these files is over 4 GB in size (for a total of over 45 GB of data).

For this data set, reading all of the information into the MySQL tables was unreasonable. Therefore, different amounts of users and network information were imported to test the system’s ability to handle large amounts of data. Figure 17 and Figure 18 show the latency in importing large data files.

<table>
<thead>
<tr>
<th># of Users</th>
<th>Time to Import</th>
</tr>
</thead>
<tbody>
<tr>
<td>50,000</td>
<td>1 minute, 25 seconds</td>
</tr>
<tr>
<td>100,000</td>
<td>2 minutes, 23 seconds</td>
</tr>
<tr>
<td>150,000</td>
<td>3 minutes, 37 seconds</td>
</tr>
</tbody>
</table>

Figure 17: Time to Import User Data

While this efficiency is not great, it only happens once when you import large quantities of data. After that, there will be longer access times for lookups, which will slow down the queries by a factor of about 5, based on examining select times within MySQL in the larger databases. This was not
able to be tested empirically since the users from the network lists and the users from the demographic information file did not match up, and often there was no data found for a given user. This is because proper sampling was not done on the data, and the first 50,000 or 100,000 users in each list were used. As we will discuss in Future Work, this needs to be rectified for effective usage of the system in general.

Another issue found when using the larger dataset occurred in the graph visualization. For the sample dataset, even the largest network comprising all 100 users was clear. For the larger dataset, some of the users have over 1,000 friends. When this graph was given to the GLEE software package, the resulting image was very wide and misshapen, making viewing the entire graph impossible. Even with the zoom functions, only small parts of the graph could be seen at once. The image is also too large to save. Figure 19 is a screenshot of what the graph looks like.

<table>
<thead>
<tr>
<th># of Network Lists</th>
<th># of Links</th>
<th>Time to Import</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>41,793</td>
<td>1 minute, 5 seconds</td>
</tr>
<tr>
<td>1,000</td>
<td>80,956</td>
<td>1 minute, 43 seconds</td>
</tr>
<tr>
<td>1,500</td>
<td>121,571</td>
<td>2 minute, 39 seconds</td>
</tr>
</tbody>
</table>

Figure 18: Time to Import Network Data

For future use of our network analysis tool, we will focus on two areas. First, we will look at improvements to the system that will allow for effective usage. Next, we will look at potential use cases and research that can be done using this software.

6. FUTURE WORK

For future use of our network analysis tool, we will focus on two areas. First, we will look at improvements to the system that will allow for effective usage. Next, we will look at potential use cases and research that can be done using this software.

6.1 System Improvements

As mentioned briefly before, an additional feature of the network analysis tool should be the ability to condense the amount of data being pushed into the system by sampling. For a given query, there are only a certain amount of users who need to be in the user data table. Similarly, only links that will be followed within that analysis should be in the link table. Clearly, any additional data would be a waste of space and will slow down the SQL lookups.

In order to do this, the system should be able to look through the link data files and pick out information that is relevant to the desired networks. This can be done using cygwin, a windows friendly linux environment, and the grep command. The same logic used for creating networks from the databases can be used to create the networks in this case.

The graph visualization problem can also be fixed when improving the system. More research can be done on what graph visualizers will not have problems when dealing with larger datasets. As noted before, a GraphML viewer can still be used, and MSAGL could potentially fix the problem.

Another feature that should be added is a separate query function for categorical data. In R, this cannot be read into a vector file, the only current option is to generate an output file with the necessary information about the network. While this is a simple step, a main goal of the software is to provide easy to use functionality.

Finally, the system could be updated to allow users to extend the query language to suit their own needs. While it was designed to allow various use cases, some may not like its structure, and may have to take multiple steps to accomplish their goals. By making the basic functions described in Section 4.3 available to the user, they could design their own ways of putting the data together.

6.2 Future Research

Since social networks have not been around for very long, research in the area is young. Nobody can deny that a lot of information is available, whether to the general public or to the companies themselves. One of the key elements of a social network, defined earlier, is the presence of a personal profile [2]. While people fill these out fully with real information, they can just as easily be inaccurate or incomplete.

One area of research could be trying to discover if we can make assumptions about an individual based on their network information. For example, if you found a person who did not fill out where they live, would it be possible to determine that with relative accuracy? This tool can allow the user to make predictions about users who have filled out that information and see how it compares to the true data. Then, once good predictors have been found, they can use this information to fill in the true blanks.

Just as people are involved in different social groups in everyday life, they may have memberships to various social networking websites. Another area of research is how to determine if two users are truly the same person. They may have the same name and age, but could be different people from different parts of the world. Hopefully, this tool can be used to identify characteristics that can predict whether two people who may seem are alike are truly one and the same.

Two principles mentioned earlier that could be tested with this analysis tool are tie strength and homophily. By examining two people, their demographic information, and their link attributes, a researcher could add a new category of tie strength. The strength could be found using empirical evidence and tested against predictions made by this system. Once a good model is found, data about users not involved in the study could be extrapolated. The principle of homophily
could be tested with a dataset containing information about people’s spending habits for a given product. By examining the networks, one could see if a person is more likely to buy a product based on the actions of their friends.

Another area this tool could be helpful in is target marketing. The study by Hill et al. [6] showed that people connected to a previous buyer of a telecommunications service were 3-5 times more likely to buy that service. To start, with the network generation capability, creating potential groups to target would be simple by using the former customers as base users. Also, adding a category marking their likelihood to buy the product, as predicted by their networks, could allow a company to narrow their search even more.

Social networks continue to grow, providing more and more information about the connections we choose to make with other people. While many of the information may be meaningless, there is likely much to learned from the networks people are involved with. This network analysis tool will allow people to look at the available social network information and hopefully draw meaningful conclusions about the data.

7. REFERENCES