Assignment 2 - Social Circles in Facebook communities.

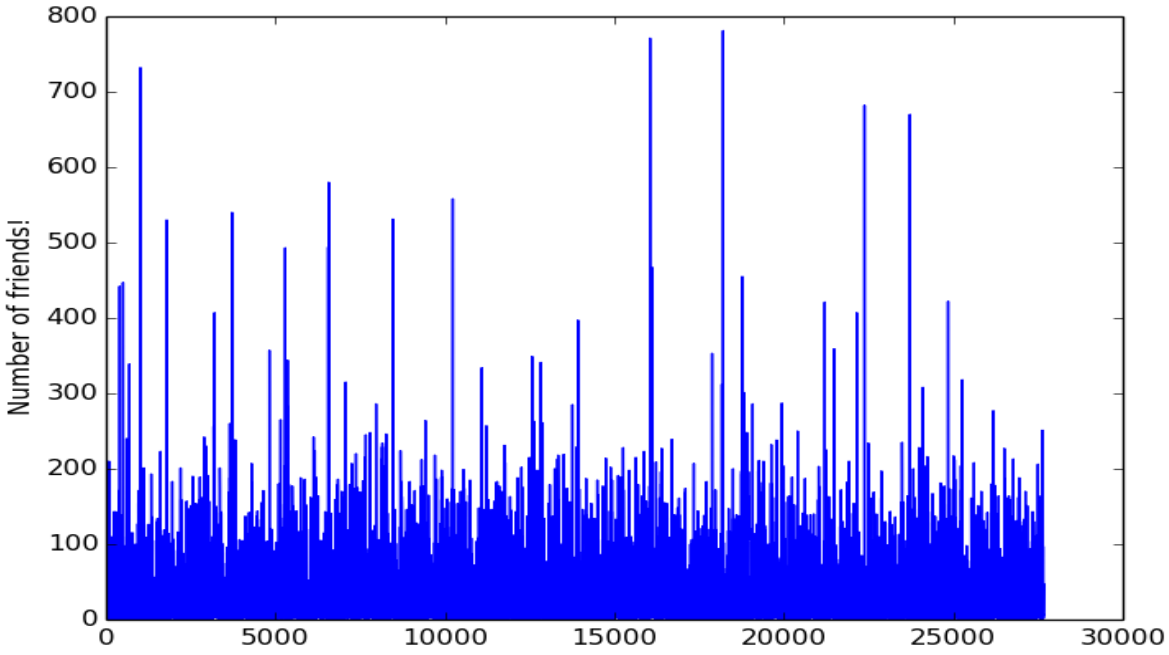
Aditya Bansal A98100081,
Nitesh Anandan A10229989,
Chawanwat Sean Techavatnavisal A09277200

1.Introduction

For years we have been using social networks online to keep up with friends and connect with people from all over the world. We add all kinds of people to our networks; friends, family, relatives, coworkers, etc. Our network of friends can be further segregated into even more concise categories: friends from college, from the same hometown, people with similar interests. These different communities are called Social Circles and they are the basis of our work in this assignment.

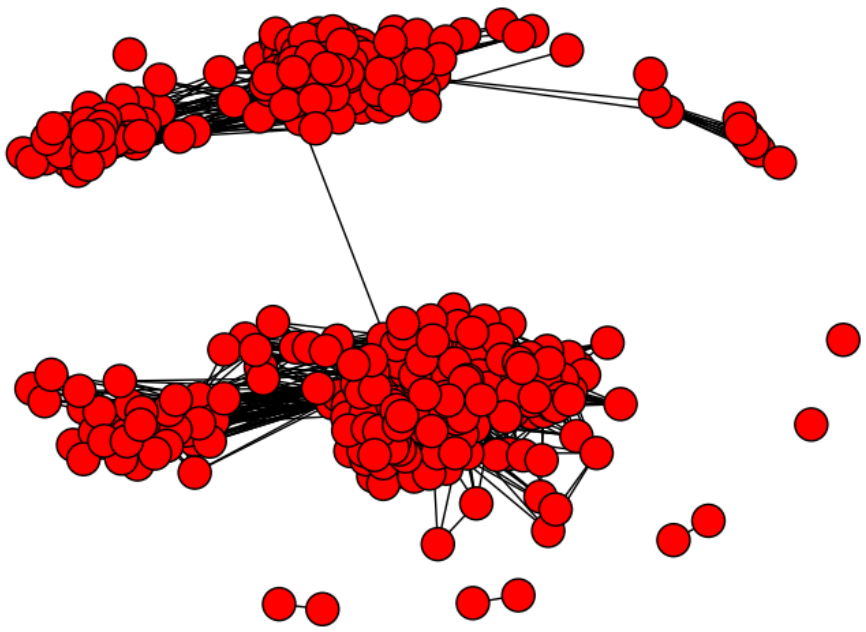
The dataset we used for this task was taken from a Kaggle competition, *Learning Social Circles in Networks*. The dataset provided some users' "egonets," a network representation of the user's friendships. It also provided some hashed features for each user such as hometown and name. Based on these given data sets, we want to construct models that will display the social circles that can be formed by manipulating and rearranging them.

There were about 27666 unique User Id's in the dataset. The average number of friends for each user was around 28.9093472132, or round of to 29 because fraction's can't exist in this case. A distribution of number of friends for each user has been provided:



From this, we could infer how big the average size of a social circle for a particular user would be. By using the NetworkX library, we also created ego graphs for each user. (Graph of the go network). One sample is shown below:

Ego Network for userId 10395:



Now that we have this dataset, we arrive to the intuition that each user belongs to different social circles, and there is very little interconnectivity between these circles. However, there is a lot of interconnectivity within these circles, and of course, there are a few outliers.

2. Predictive Task

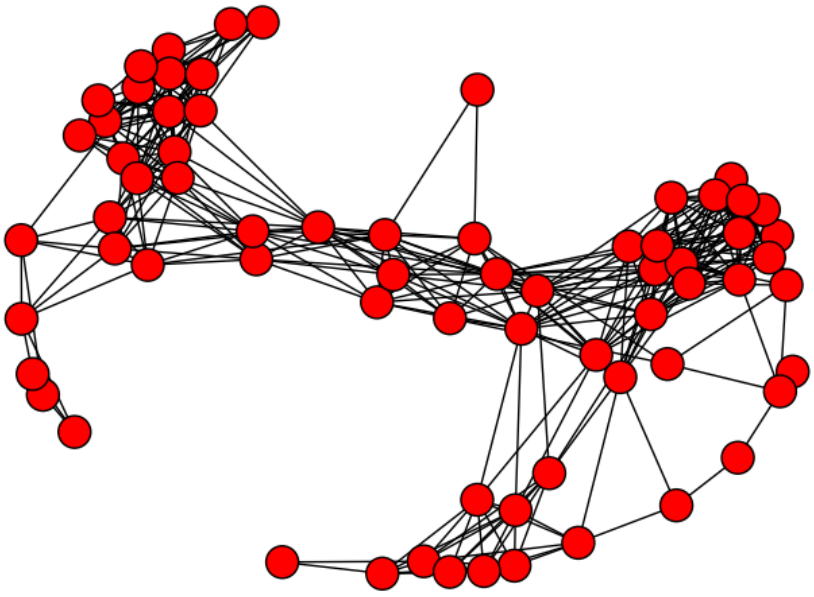
We want to predict what communities would form or what circles can be created using the given friendships. To evaluate this model, we utilized the Min Edit Distance to compute how far our circles are from the actual data. Each addition/removal of a friend to/from a circle counts as 1 edit. Since the Kaggle competition had ended, we could not check our solutions properly. We did however modify the given metric file to calculate the Min Edit Distance between our predictions and the Ground Truth (.circle files).

Models:

We learned a lot about social communities in class. Communities contain many defining aspects to it such as: one or bidirectional relationships, outliers, true positives, false positives, etc. Some models that we want to try are connected components, clique percolation, and clustering.

The baseline provided on Kaggle was just to get the strongly connected components for test users, and put them in a circle. This was what we used as a baseline. Our target was to beat it much as we did in Assignment 1, where we were able to obtain a respectable amount of increase in prediction accuracy for both the training and test sets.

Our first model was to run Clique Percolation on the ego networks of each user in the test data. A sample max clique graph is shown :



Using this, we were trying to figure out what value we should use for k clique percolation. This graph did not show much, so we decided to play with different values for k.

We then decided to give k-means clustering a shot. We ran through the features.txt file and used all entries' hometown info to cluster the nodes, since it is quite likely that people of the same hometown may be friends on social networks [5]. We thought that with this info at hand we may be able to predict how well connected the users were within a hometowns. From there we cross-checked the Egonets to see if people belonging to the same hometown were in-fact well connected or not but after having done that, we ended up with one cluster per hometown. We noticed that this was somewhat of a dead end considering every user would end up in just one circle.

After a bit more research we found that a notably different clustering algorithm, Spectral Clustering, might give us a better chance to predict social circles. As K-Means is a way to cluster based on proximity, some research showed that Spectral Clustering is a way to cluster based on "Affinity" and connectivity (like graphs). [2]

We tried to implement Spectral Clustering on various aspects of the ego networks, and in various ways.

We first implemented Spectral Clustering on the connected components of each ego network in the test set. This did not yield a very good result (as explained below). So we decided to implement Spectral Clustering on the entire ego network instead of individual connected components. This yielded a much better score.

3. Literatures and Related Works

We started manipulating our data set with clique percolation as it is a method that is well-suited for analyzing the overlapping community structure of networks. For this technique, we consulted a paper named *Clique Percolation in Random Networks* by Derenyi, Palla, and Vicsek [4]. The paper explained clique percolation as very much like a regular edge percolation cluster in an adjacency graph where the vertices represent the k-cliques of the original graph [4]. There also exists an edge between two vertices if the corresponding k-cliques are adjacent. One of the most important aspects of clique percolation is that it can be made stronger by using a more detailed representation of the cliques. By adjusting the k-th threshold in finding cliques, we can identify communities with different levels of aspects such as cohesiveness, locality, and density.

After clique percolation, we then moved onto trying k-means clustering. K-means clustering is a tool that can be used to find groups of respondents, objects, or cases that are similar to one another but different from a different group [5]. K-means is fast, however, all results are dependent on the initial k number of groups that the algorithm uses. It classifies or group nodes based on attributes or features into k number of groups [5]. The grouping process is done by minimizing the distances between data and the corresponding cluster centroid. The limitation of k-means is in its cluster model. The clusters are expected to be of similar sizes so that assignments to the nearest clusters will be correct and if the sizes are wrong, then overlapping can occur.

Next we attempted Spectral Clustering, which is the idea of clustering by affinity of the connectedness of our dataset. This process involves a few steps, the first of which is to construct an adjacency matrix which would represent our graph [2]. Then we compute the eigenvalue decomposition of the graph by identifying the eigenvectors. Then we partition our nodes by looking at the various eigenvectors and figure out the different sets the nodes belong to.

Professor McAuley's paper on the topic was also consulted and examined to understand the concepts better. In his work, *Learning to Discover Social Circles in Ego Networks* [3], Professor McAuley elaborated on ways to utilize machine learning techniques such as assigning weights to each data to quantify its significance relative to other data as well as representing different aspects of our dataset in the form of a tree where each level of it contains increasingly specific information [3].

While McAuley found that his experiment performed much better than the baseline he set, he also noted that the experiment performed best on *complete* data such as data with accurate and plentiful labels and mutual ties between nodes instead of a one-directional tie. This works in our favor as our dataset also contain nodes with mutual ties albeit some of them are not

explicitly stated.

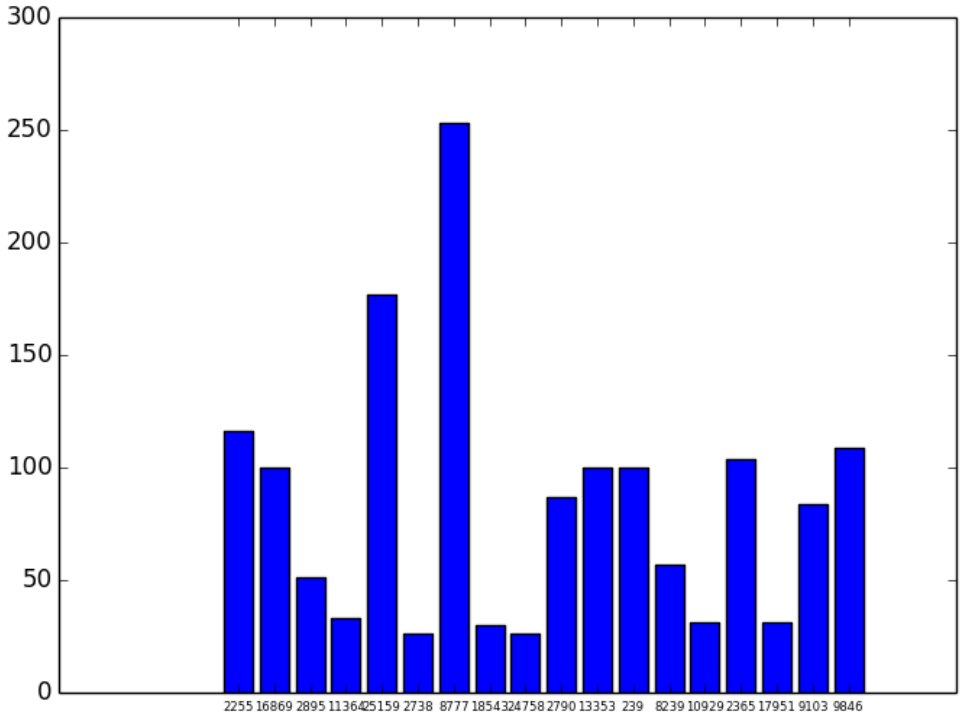
For general inspiration and broad overview of various methods and experiments already done on the topic of community detection such as graph construction, spectral clustering, and inferring dynamic community behavior, we consulted a publication by the Massachusetts Institute of Technology named *Social Network Analysis with Content and Graphs* [1]. It describes techniques that are used to quantify and provide concrete result dataset such as relational probability data tree, relational dependency networks, and collective classification. It also expands into the versatility of applying multiple algorithms such as infomap and modularity optimization [1]. What was helpful to us was their explanation of the nuances of community dynamics. Tasks such as behavior analysis and prediction and identity and pattern-of-life analysis are possible using community detection. Another important aspect of community dynamics is that it is ever-changing, and this fluidity allows us to leverage the temporal aspects of social network analysis.

We think collective classification has great potential with the data set that we are working with as it allows us to make prediction with a certain amount of accuracy is for situations that we have very little specific information about. Collective classification [1] is possible when many individual class labels are unknown but are connected via social or organizational ties. This allows us to propagate predictions about one individual to other related individuals and it is done through using individual conditional models such as Bayes' rule built for each distinguishing feature of each node. Collective classification is also the basis of the relational probability tree which is related to the tree that Professor McAuley constructed in his paper.

4. Results and Analysis

Our model that gave us a starting point was clique percolation. Since we did not have the test set, we used the training set as the test data as well.
Basically what the clique percolation model did was find k clique communities (k varied for different users) for the test set users, using their egonet. Then, we put each community in a separate circle. Since we didn't have a test set, we split up the training data to give us test set.

We then took the test data, and converted it into a different format so that we could use the Kaggle metric python file (which calculated the Min Edit Distance). Finally, we calculated the edit distances between our predictions, and the test files.
We used 18 Training files as Test files, and the loss turned out to be "1515" using clique percolation using clique size 5.
Below is a graph that shows how the user loss varies for the test users.



We also played with different clique sizes, but they gave a bigger loss than size 5.

The baseline (connected components gave a loss of 1795) for this particular model. Hence, we can conclude that our model made a significant jump over the baseline provided.

The results from our attempts at K-Means clustering weren't too promising. Since we ended up with one circle per hometown, we didn't have a way to find out how well connected the users of a particular hometown were given that not all userID's had an Egonet. We did some research and tried Spectral Clustering as well, which gave better results.

As we read from Tom Denton's post, we wanted to implement Spectral Clustering on the dataset. This gave a score that was slightly lower than the baseline (1724), but was nowhere close to beating clique percolation. Looking at the social circles formed, we noticed that is was creating a bigger number of circles than clique percolation, and also the circles were smaller too.

To counter this, instead of running Spectral Clustering on the connected components, we ran it on the entire adjacency matrix of the egonet of the particular user, and printed the different clusters as individual labels.
This gave us a score of 1575.

Final Thoughts

We were expecting Spectral Clustering to give us a better result than Clique Percolation. Our assumption was that, since the test set that we had was so small, it could have been a coincidence that Clique Percolation was giving a better result than Clustering. Our expectation turned out to be correct and special clustering gave us the best end result compared to other methods that we tried.

With more time, we expect that we would be able to improve on our model by incorporating the different features for each userId. Our team had a lot of fun working on this assignment. One thing that we think would be a good idea to work on in the future is to come up with different techniques to calculate the errors generated. It would also have been preferable if we were able to test our solutions on the kaggle board as well to gauge how we were doing as we progressed.

Works Cited

[1] Social Network Analysis with Content and Graphs William M. Campbell, Charlie K. Dagli, and Clifford J. Weinstein
https://www.ll.mit.edu/publications/journal/pdf/vol20_no1/20_1_5_Campbell.pdf

[2] Spectral Clustering: A quick overview
https://charlesmartin14.wordpress.com/2012/10/09/spectral-clustering/

[3] Learning to Discover Social Circles in Ego Networks
http://i.stanford.edu/~julian/pdfs/nips2012.pdf

[4] Clique Percolation in Random Networks
http://lanl.arxiv.org/pdf/cond-mat/0504551v1.pdf

[5] Analysis of Social Networking Sites Using K-Mean Clustering Algorithm
http://interscience.in/IJCCT_Vol3Iss3/paper20.pdf