# Structure Analysis of Friends Circles

By

Kavana Anand
kkavanaanand@cs.stonybrook.edu

Swijal Patil
swapatil@cs.stonybrook.edu

# INDEX

## 1. Introduction

With advent of sites like Facebook, LinkedIn, Twitter etc., Social Networks have gained vast popularity. Number of users participating in these networks is huge and still growing. These networks are ubiquitous. The discovery of close-knit social circles in these networks is of fundamental and practical interest. This project is about discovery of these circles and determining the reason for formation of these circles in an individual's network which we call it as ego network. Project is aimed to cluster the individual's ego network into possible social circles and analyze the structure of each circle and abstract the common parameters and derive why a particular circle was formed and label the circle and verify with the ground truth.

To start with, it is essential to obtain the required raw datasets that are available and useful for this project. All the datasets used in this project is explained in section 2 of this report. Once all the required datasets are obtained, implementation continues with clustering the network using efficient clustering algorithm as described in section 4 and constructing an algorithm to name these clusters which is explained in section 5 and section 6. End results of this project after several testings and modifications and retesting turns out to match really well with the ground truth and has been verified as explained in section 7. While developing this project, resources that were referred are mentioned in section 8.

## 2. Dataset

We have used number of datasets obtained from different resources for different modules of the project.

- **Mutual friends links**

  First among them is the mutual friend's link which is obtained from Facebook application Give Me My Data API. This data provides all the links present between ego and his/her friends. This is later processed and converted into adjacency matrix which is then used for mapping ego network and its clustering.

- **Ego's profile details**

  Next data is ego's profile which is obtained using Graph API of Facebook. This data details about the ego from which attributes such as name, current location, hometown, education and work have been used in categorizing the ego's network into possible social circles/clusters.

- **Ego's groups information and its members**

  Along with the above data we also retrieve group's information using the Graph API of Facebook which gives names of all the groups to which the ego belong to and all the members of the group. This is later manually modified to retain only members who are friends with the ego. This is data is also used in categorizing the network also later when labelling the cluster.

- **Friends details**

  One of the most important datasets that is used in this project is friend's details. This data is obtained from StatCrunch. This data provides details about ego's every friend's name, current location, hometown and education. This is used while labelling the cluster when a node at random is picked and determining why he/she belongs to particular circle.

## 3. Methodology

This project is developed in two main modules. First module involves mapping the ego network of the ego using the adjacency matrix and clustering the network using kMeans clustering which divides the network into k social circles efficiently based on the links in the matrix. Second module involves deriving the possible social circles for the given ego network using ego's profile data and then constructing the algorithm which takes the formed circles and possible names of the circles derived as input and runs series of comparisons for randomly picked nodes taking around average of five samples for every circle and decide why those nodes were part of the same circle and finally label the circle with appropriately based on the result obtained.

## 4. Clustering

To start with clustering, we need the appropriate adjacency matrix which depicts the links between the ego and his/her friends. From the dataset of mutual friend links we obtained, we run the *getAdjacencyMatrix.java* and produce the adjacency matrix A. This project uses kMeans Cluster algorithm to cluster the network. Following are the details of how social circles are formed which is the output of the program *kMeans.m*

- Diagonal matrix D is computed which contains degree of each node in the network and Laplacian matrix is calculated by L = D - A
- Eigen values and vectors of the Laplacian matrix L are computed. Then kMeans algorithm with k = 5 or k = 6 (depending on the possible clusters an ego can have) is applied to these values which divides the network into k social circles
- Graph is plotted in 3D using 2nd, 3rd and 4th eigen vectors and from this graph kMeans result can be clearly seen with formation of 5 different clusters each represented by different color

## 5. Analysis of friends circle

- **Buckets creation:**

After we had clusters labelled with serial numbers, our task is to analyze the friend data to label the clusters identified. Initially we started creating some random buckets with the ego's properties as the clusters that will be labelled will be related to ego's properties. We created buckets based on ego's information such as current_location, hometown, education which can comprise of multiple values such as undergrad college, grad university, school and so on, work where ego's has worked. It can also have multiple values. Now as the buckets are created they are made as a container to map the remaining details to one of them.

- **Mapping groups data:**

After identifying buckets our next step was to map all the group names to one of the appropriate bucket with some similar characteristics. So we considered each group name initially and then checked whether any of the bucket name contains part of group name, if so then we have mapped the group name under that bucket. If in case there is no match found with respect to name then we explored the description of the group and tried to find out any string matching to bucket name. If it does then map it to appropriate bucket else there is no match for that particular group as per our project and we will not consider the group for our project analysis. Mapping groups will help us to identify group members who are ego's friends related to which bucket.

- **Assumptions:**

For our project to label the clusters we have to go with random number of samples. To label the cluster properly, we have to consider the analysis based on few random samples rather than based on single one. So we have considered 5 random samples of 10 friends each for a particular cluster to label. Based on the majority labelling we will label the cluster but if in case there is no majority then we will pick more random samples and try it again.

## 6. Algorithm

As per the assumptions made earlier algorithm will take 5 random samples of 10 friends each and compare the friends' data with the available buckets to map and get the majority for labelling.

```
For each cluster {
    For each sample out of 5 {
        For each friend out of 10 from the sample {
            For each attribute of friend's data {
                If attribute is matched with bucket names or any of the group
                names then add that friend into the bucket if not already
                added
        }
        Find the name of the bucket with highest number of mappings
    }
    Label the cluster with the bucket name which has the highest mappings overall
    from 5 samples
    If more than one bucket recorded same number of times then redo sampling
}
```

For each cluster to be labelled, we have the number of friends and details of friends that are related to that particular cluster. So we retrieve a sample of 10 friends at random using a random function shuffle which shuffles the array at random and gives top 10 friends list as output. Now for each friend we need to associate it to a particular predefined bucket with respect to its parameters present in dataset. We have different parameters related to friend data such as hometown, current_location, education which can comprise of multiple values such as undergrad college, graduate university and school.

Initially we check directly if any of the buckets value matches the current selected parameter. If yes then we check whether this particular friend is already mapped to that bucket or not. If not then we will map the friend to that bucket. If incase the bucket names does not help to map the friend to any of the bucket then we move to groups data to map the things. We take list of all the groups where the friend is a member and then for each group we check to which bucket it belongs to and map the friend to that bucket. Mapping of groups to the buckets is already done as mentioned in *section 5*. Note that a particular friend can be mapped to multiple buckets but cannot map to same bucket multiple times in a single sample run. Now that all the friends from the sample are mapped to some or the other buckets, we will select the bucket with maximum number of friends mapped. If in case there is no majority then we will void the sample and try on new sample. Similarly for 5 samples we will have buckets list with majority of friends mapped. Now consider the bucket with majority of mappings recorded. Incase no majority found try on additional samples. The label of the cluster will be the bucket mapped majority of times.

## 7. Results

We conducted the experiments on Facebook data of a particular ego. Initially we have worked on our ego data and then later we tried the same on few of other people data. Data collected for a particular ego is as mentioned below:

- **kMeans Clustering**

  Following 3D graph *Figure 1* plotted after applying the kMeans clustering algorithm to the adjacency matrix of n nodes where n=746 and k=5 clearly shows that 5 different clusters are formed with cluster 1 having 223, cluster 2 having 32, cluster 3 having 82, cluster 4 having 397 and cluster 5 having 12 nodes out of total 746 nodes each represented by different color.

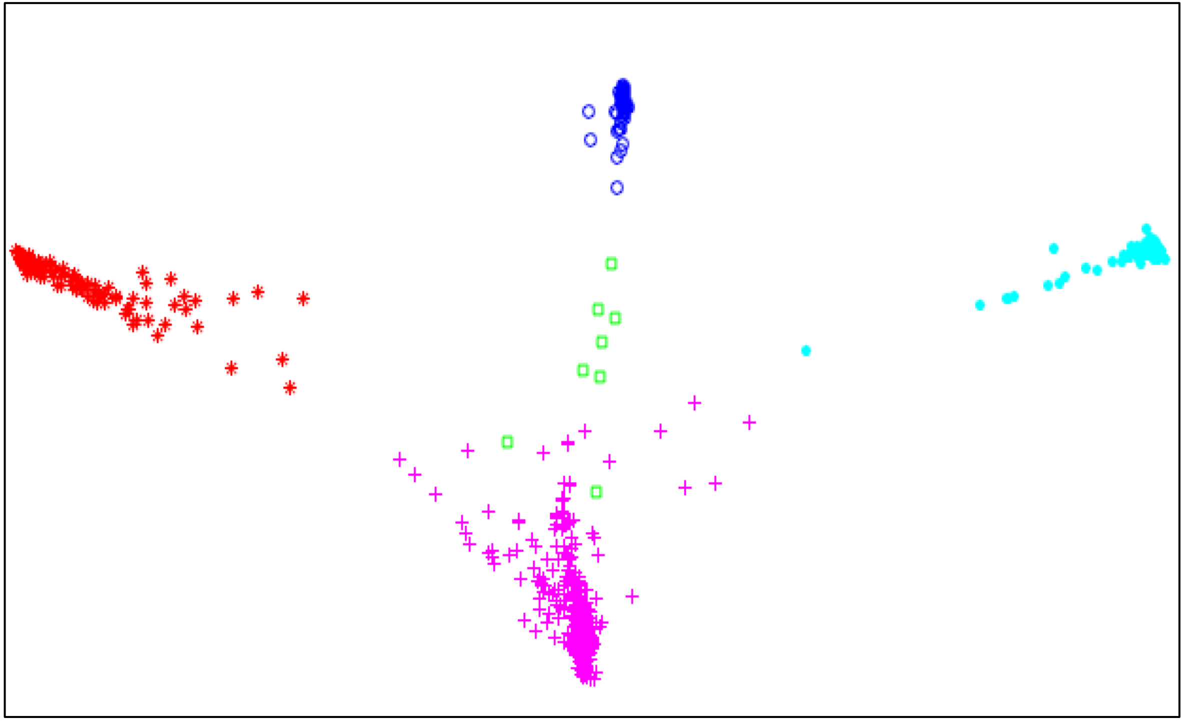

*Figure 1*

- **Ego data**

Current_location - Stony Brook New York, Hometown - Shahada, Colleges - PCCOE, SBU, SAMEMS, Work - TCS.

- **Group data**

Here we have collected all the groups of the ego along with the members of groups which are friends of that ego. There were around 18 groups each consisting of around 55 member friends on an average.

- **Friend data**

Here ego had around 746 friends and data related to each friend was collected and stored. Data stored was current_location, hometown, colleges attended.

- **Buckets formation**

Based on Ego data, we have classified parameters into 6 different buckets as follows:
Hometown - SHAHADA, Undergraduate - PCCOE, Work - TCS, Graduation - SBU, Current Location - STONY BROOK, School - SAMEMS.

- **Mapping Groups**

Based on buckets all the 18 groups were mapped to one or some bucket. Note group can map to only one bucket.

- **Generating random samples**

5 random samples of 10 friends each were generated using random function and given as input to the algorithm one sample at a time.

- **Labelling**

On analyzing all the samples for a given ego, we found the results as mentioned below:

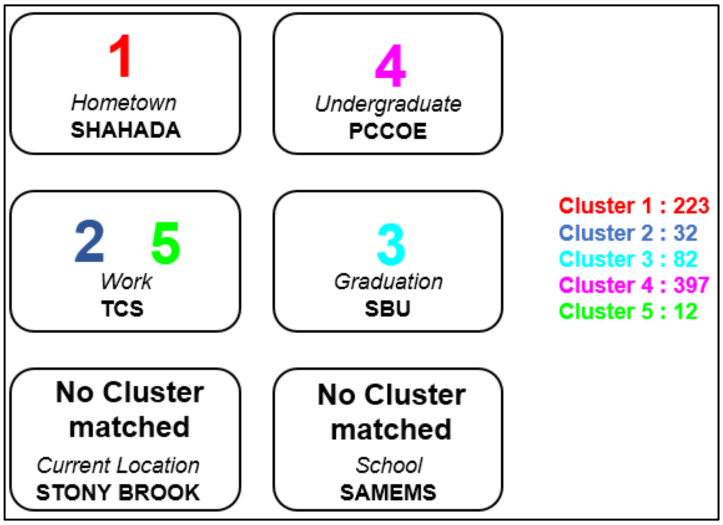

*Figure 2*

Here we can see that the 5 clusters that we got during the kMeans clustering were labelled.
Cluster 1 consisting of 223 friends was labelled to ego's Hometown – SHAHADA
Cluster 2 consisting of 32 friends was labelled to ego's Work – TCS
Cluster 3 consisting of 82 friends was labelled to ego's Graduation – SBU
Cluster 4 consisting of 397 friends was labelled to ego's Undergraduate – PCCOE
Cluster 5 consisting of 12 friends was labelled to ego's Work - TCS

On verifying with the ground truth labels that we deduced initially, all the clusters are matched perfectly for what community they belong. On closely analyzing we found that some of the friends were getting hashed to different buckets due to the multiple facts. Multiple groups' description contained information such that they matched with the bucket's name which was not intended to. Another fact is due to incomplete updates of user they were not able to match the appropriate bucket. For example, many of ego's friends didn't update their current_location as STONY BROOK who actually does live in here but they had their Education updated as SBU which made them all classify to Graduation bucket rather than going to current_location bucket. Hence there was no cluster mapped to current_location. Another instance is School – SAMEMS bucket where no cluster was mapped. This was due to the fact that all the people who belongs to ego's school are from the same hometown as of ego and hence all of them got mapped to hometown bucket leaving school bucket with no cluster found. Both of the no cluster found buckets corresponds to the fact that they were rarely chosen as majority of the friends' mapped bucket.

Another interesting observation is that the bucket TCS was mapped to 2 clusters. On further analyzing this we found that ego had worked at 2 different locations for same firm TCS. But as our clustering is based on mutual friends' links, there were 2 different clusters found due to 2 different groups of ego for same firm depending on the location.

## 8. References

[1] http://www.informatik.uni-hamburg.de/ML/contents/people/luxburg/publications/Luxburg07_tutorial.pdf

[2] https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm

[3] http://www.statcrunch.com/frienddata/

[4] https://apps.facebook.com/give_me_my_data/

[5] https://developers.facebook.com/tools/explorer/