Learning Social Circles

Group 10: Divya Rathore, Shruti Tripathi

Executive Summary

In today's data-driven world, social networking sites generate a large amount of data of users and their friends. To manage these data users create social circles manually. As this task becomes increasingly tedious with the growing number of friends a user has, we have followed a machine learning task of creating a user's circle and identifying the features that define the circles. We develop a model for detecting circles that identifies all the members belonging to the circle of a user. It also identifies the features that are common between those users, which tells us what puts them in the same social circle. We converted the egonet files into graphs and used a graph theory method called cliques to identify connected subgraphs that would form social circles of the user's friends. The accuracy of the circles created is further improved by filtering out circles whose members do not have any feature in common.

Introduction

Social networking sites provide an option to manually categorize personal networks into Social Circles, which help users organize their personal social networks. All major social networks provide such functionality, for example, ‘circles’ on Google+, and ‘lists’ on Facebook and Twitter. Each circle consists of a subset of a particular user's friends. Such circles may be disjoint, overlap, or be hierarchically nested.

The motive of this project is to identify social circles automatically based on the network connections of the users and their friends. User’s friends create an overwhelming amount of data and to cope with the information users need to categorize friends into social circles. So far, this is being done manually. The purpose is to avoid the laborious task of constructing the circles and updating it whenever a user’s network grows. More precisely, for a given user and their personal social network, our goal is to identify social circles, which would be a subset of that user’s friend list.

The circles derived can overlap heavily or can be hierarchically nested in larger ones. Also, these circles are not only densely connected but its members also share common properties or traits.

Data Description

The dataset was collected from the website for the competition platform of Kaggle: https://www.kaggle.com/

The data consist of

- A set of users each of whose circles must be inferred
- A list of the user's friends
- Anonymized Facebook profiles of each of those friends
- A network of connections between those friends (their "ego network")

We were able to obtain ego-networks, users and circles from three major social networking sites: Facebook, Google+, and Twitter. The dataset was designed from publicly accessible data.

Website	Ego-networks	Users	Circles
Facebook	10	4,039	193
Google+	133	106,674	479
Twitter	1000	81,362	4869

The ego-networks contains nodes within the range of 10 to 4,964 nodes.

While collecting data from Facebook, the users were asked to identify the social circles their friends belonged to such as common universities, hometown or relatives.

The 133 Google+ users were those who had shared at least two circles, and whose network information was publicly accessible at the time of the data crawl.

The size differences between these datasets simply reflect the availability of data from each of the three sources.

Egonets:

The total of 110 egonet files and each file in this directory contains the ego-network of a single Facebook user, i.e., a list of connections between their friends. Each file userId.egonet contains lines of the form

UserId: Friends
1: 4 6 12 2 208
2: 5 3 17 90 7

where

- 'User' is the target individual whose network is the focus of the task
- 'Friend' is the friend of the target individual (user) that needs to be placed within a circle.
- The node -adjacency lists indicate that User 1 is friends with Friend 4, Friend 6, etc.
- Edges are undirected in Facebook and it can be assumed that every ID in the file is friends with the user.
- The owner of the egonet file is friends with all the users within the file however, he/she won't appear in that file because of the basic assumption that the user never appear in their own social circles.

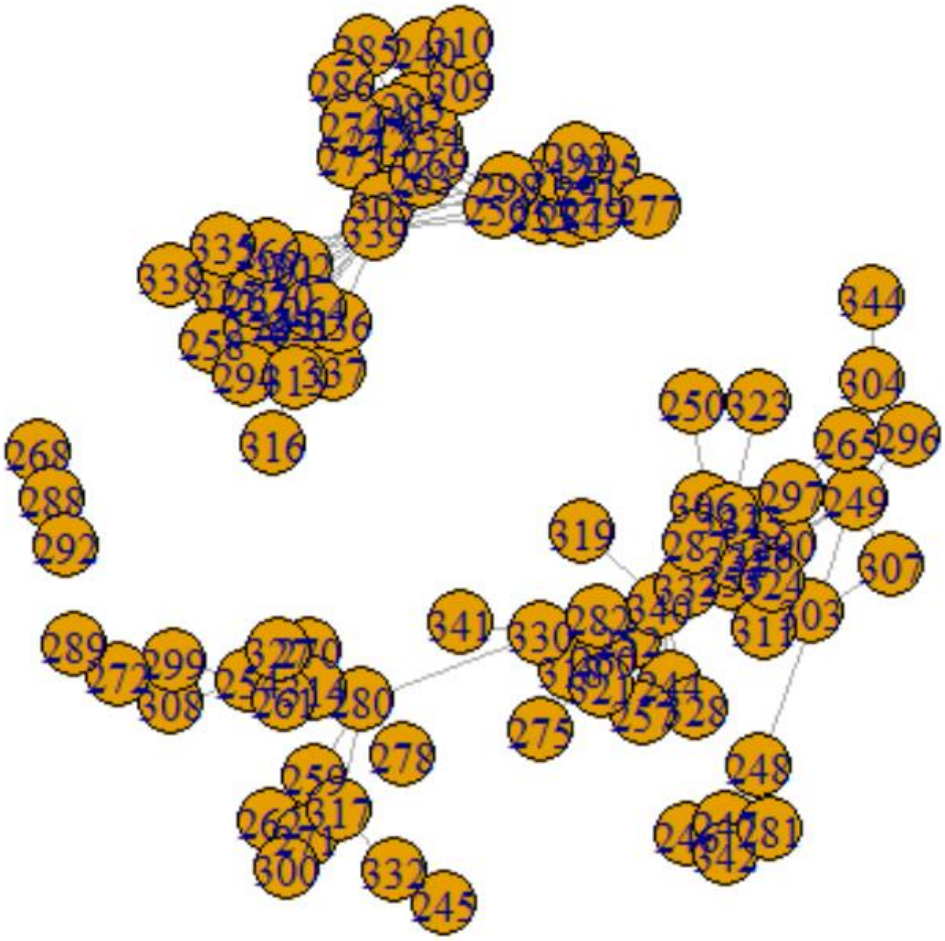

Figure 1: 239.egonet

Figure 1 represents a graph plot constructed from one of the egonet files i.e. file- 239.egonet. The graph represents the network connections of the user: 239. We can notice the network looks scattered which is because the user 329, which is connected to all the nodes within the file is not

included in the plot. However, we can clearly see some clusters forming which could potentially represent some of the overlapping or hierarchically nested personal circles of the user, whose egonet file it is. Our task is to determine all such circles for all the egonets provided in the dataset.

Features:

The features file contains features for all the users. Each line is of the form
UserId feature1 feature2 feature3 ...

Each line starts with the id of a user. There is one line for every user and all of their friends. The remainder of the line is the set of features for which the user has hashed values.

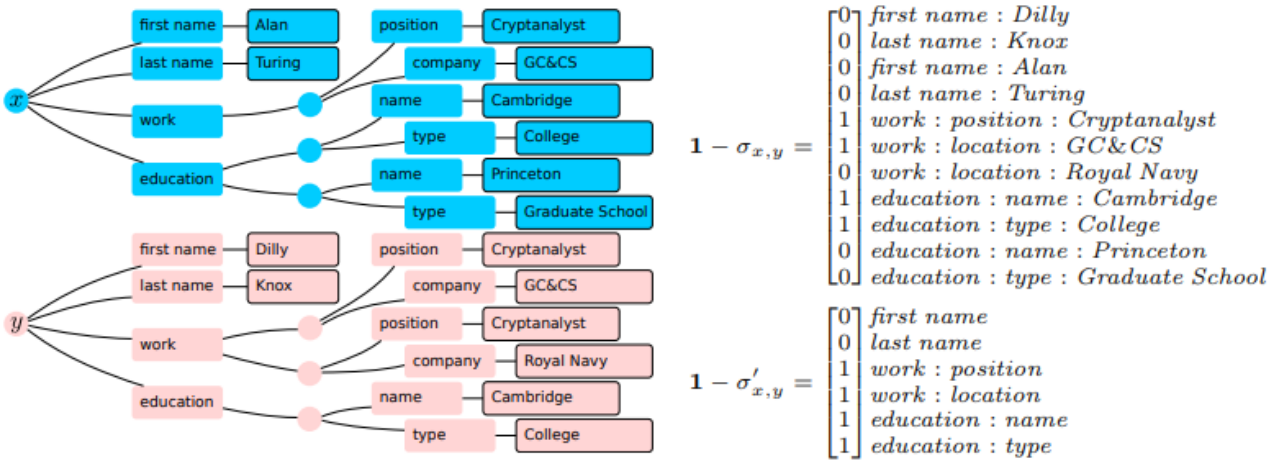

Figure 2: Tree Structure of Features

Examples of trees for two users x (blue) and y (pink) are shown at left in Figure 2. These user profiles are tree-structured, and features were constructed by comparing paths in those trees. Two schemes for constructing feature vectors from these profiles are shown at right:

(1) The right side of the figure shows binary indicators. These are used for measuring the difference between leaves in the two trees.

e.g. 'work→position→Cryptanalyst' appears in both trees.

(2) (bottom right) shows summation of the leaf nodes in the first scheme, maintaining the fact that the two users worked at the same institution, but discarding the identity of that institution.

In order to anonymize the data, the feature files were encoded and the values were hashed to numbers so each line in the final file looks like

```
8 last_name;8 hometown;name;2 hometown;id;2 birthday;7
```

which would mean that user John (encoded as 8) has the last name John (encoded 8) and hometown name as ABC City (encoded as 2) and so on.

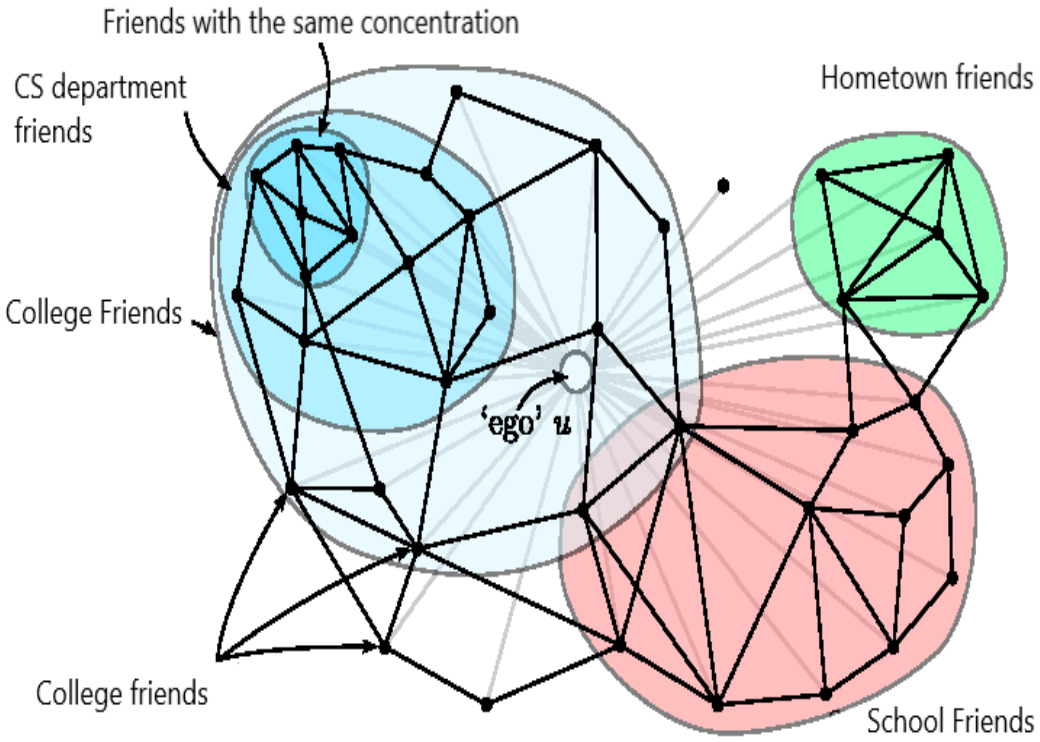

Figure 3: Different circles based on different features

Figure 3 represents how Different circles can be defined by the features constructed by comparing the profile trees for the users within those circles and overlapping circles can form stronger connections within weaker ones.

The network connections between the nodes are not enough to define potential personal circles, there needs to be some common properties or aspects between the nodes of that circles to get a more precise prediction of circles. We cannot simply formulate this as a simple clustering problem because it would fail to capture individual aspects of users' communities. So, we aim to discover circle memberships and to find common properties around which circles form.

Feature	Description	Feature	Description
birthday	birthday	location;id	location id

education;classes;description	class description	location;name	location name
education;classes;from;id	class teacher id	middle_name	middle name
education;classes;from;name	class teacher name	name	name
education;classes;id	class id	political	political
education;classes;name	class name	religion	religion
education;classes;with;id	classmates id	work;description	description of work
education;classes;with;name	classmates name	work;employer;id	employer id
education;concentration;id	concentration id	work;employer;name	employer name
education;concentration;name	concentration name	work;end_date	end date of work
education;degree;id	degree id	work;from;id	work id
education;degree;name	degree name	work;from;name	work assignee name
education;school;id	school id	work;location;id	work location id
education;school;name	school name	work;location;name	work location name
education;type	type of education	work;position;id	work position id
education;with;id	education id	work;position;name	work position name
education;with;name	education name	work;projects;description	projects description
education;year;id	education year id	work;projects;end_date	project end date
education;year;name	education year name	work;projects;from;id	work assignee id

first_name	first name	work;projects;from;name	work assignee name
gender	gender	work;projects;id	project id

hometown;id	hometown id	work;projects;name	project name
hometown;name	hometown name	work;projects;start_date	project start date
id	id	work;projects;with;id	project id
languages;id	language id	work;projects;with;name	project name
languages;name	language name	work;start_date	work start name
last_name	last name	work;with;id	coworker id
locale	locality	work;with;name	coworker name
location	residence location		

Table 1: Feature List

Construction of Circles

The ego networks provided in the data represents undirected graphs and connected components of the python package can be used to generate all clustered nodes in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the supergraph. A vertex with no incident edges is itself a connected component. A graph that is itself connected has exactly one connected component, consisting of the whole graph.

Figure 4 shows some connected components of an egonet file (0.egonet) and Figure 4.1 show the output in the form of a list of connected nodes in the file. The graph shown in this illustration has three connected components which provide us with clusters however this approach is not particularly satisfactory and fails to capture individual aspects of users' communities.

Figure 4: Connected Component

```
[{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 27, 28, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58,
59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76,
77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94,
95, 96, 97, 98, 99, 100, 101, 103, 104, 105, 106, 108, 109, 110, 111,
112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125,
126, 127, 128, 129, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140,
141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154,
155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168,
169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182,
183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196,
197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 209, 210, 211,
212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225,
226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238}, {208,
130}]
```

Figure 4.1 List of Connected Component

Cliques:

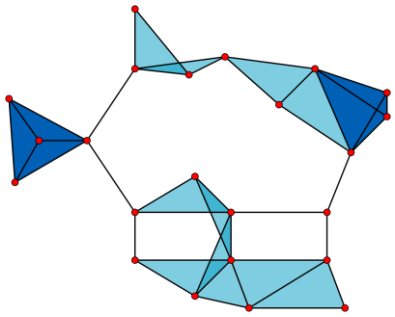

Figure 5: Clique example

A clique of a graph is a subgraph which is a subset of vertices of an undirected graph such that every two distinct vertices in the clique are adjacent; that is, its induced subgraph is complete. The clique of largest possible size is called max-clique. A clique of size k is called k-clique. Cliques are one of the basic concepts of graph theory and are used in many other mathematical problems and constructions on graphs. The python package networkx has an enumerate_all_cliques method which returns all cliques in an undirected graph. This method returns cliques of size k = 1, 2, 3, …, maxDegree - 1., where maxDegree is the maximal degree of any node in the graph.

The minimum clique size we considered was 5 i.e. the circles should have 5 or more members/ nodes in them. In real life, there could be fewer members in a circle but considering the large size of the egonets provided in the dataset, we decided to go with 5. The egonet file has nodes in the range of 10 to 4,964. and calculating cliques for thousands of nodes takes too much time on local systems. Due to the limitations of personal laptops, we decided to limit our computation of circles to the egonets of node size 65 and less.

Output

The desired output contains a list of nodes grouped together in the form of circles of varying sizes.

The derived circles follow certain properties such as the same nodes can be a part of more than one circle. There are overlapping circles and stronger circles are formed within weaker ones.

Figure 6 shows how the derived circle looks like for one egonet file. Similar circles were formed for all the egonet files.

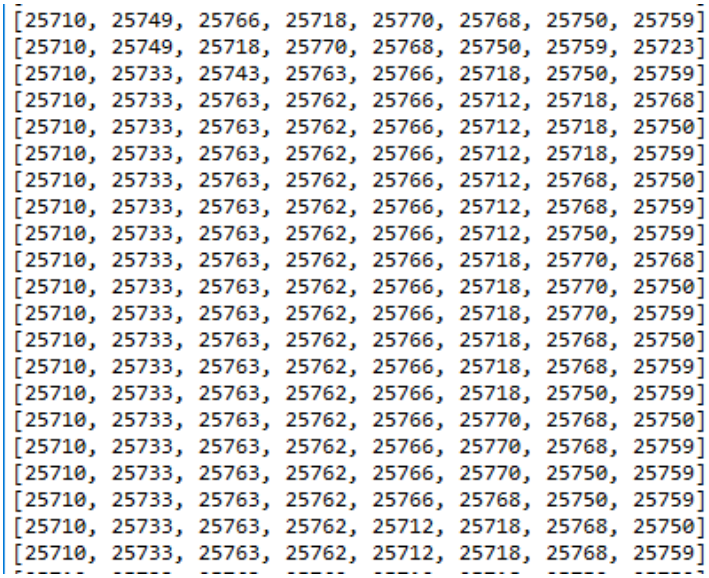

Figure 6: Social Circles

Circles Validations

As discussed above, this project is aiming to not only discover circle memberships but also find common properties around which the circles are formed. To verify the derived circles we combined our results with the feature data provided. For every ID in the dataset, there is a feature string that denotes the features formed by comparing the tree structures of the profiles. By extracting the feature strings for every node in the circle and then comparing them, the common properties of the circles were found, which was then used to verify the circle formations.

To execute the verification of circle, we designed the algorithm using python and stored the respective feature lists for each node within a circle and compared the feature lists of all the nodes to extract the common features, the final output is shown in figure 7.

The results in the final output image (fig 7) are easy to interpret, the first line shown in the result is a circle of five nodes which represents a group of five friends of the same gender, who have had the same type of education and live in the same location. So, we successfully found the common aspects of the derived circles and we also notice that the list of common feature is not limited to only one property, the group can be defined by more than one common feature. So different circles were formed by different aspects and one person/ node can be a part of more than one circle based on different features.

```
[1316, 1334, 1318, 1338, 1327] feature: ['education;type;1', 'gender;0', 'location;id;287', 'location;name;287\n']
[1316, 1334, 1318, 1336, 1354] feature: ['locale;1', 'work;employer;id;965', 'work;employer;name;951', 'location;id;287']
[1316, 1334, 1318, 1336, 1319] feature: ['locale;1', 'work;employer;id;965', 'work;employer;name;951', 'location;id;287', 'location;name;287\n']
[1316, 1334, 1318, 1336, 1345] feature: ['work;employer;id;965', 'work;employer;name;951', 'location;name;287\n']
[1316, 1334, 1318, 1336, 1351] feature: ['work;employer;id;965', 'location;id;287']
[1316, 1334, 1318, 1336, 1327] feature: ['locale;1', 'work;employer;id;965', 'work;employer;name;951', 'location;id;287', 'location;name;287\n']
[1316, 1334, 1318, 1336, 1339] feature: ['locale;1', 'work;employer;name;951', 'location;id;287', 'location;name;287\n']
[1316, 1334, 1318, 1354, 1345] feature: ['education;type;1', 'gender;0', 'work;employer;id;965', 'work;employer;name;951']
[1316, 1334, 1318, 1354, 1351] feature: ['location;id;287', 'work;employer;name;956']
[1316, 1334, 1318, 1354, 1327] feature: ['location;id;287', 'locale;1', 'gender;0', 'work;employer;id;965', 'work;employer;name;951']
[1316, 1334, 1318, 1354, 1339] feature: ['locale;1', 'education;type;1', 'location;id;287']
[1316, 1334, 1318, 1319, 1345] feature: ['education;type;0', 'education;type;1', 'gender;0', 'work;employer;id;965', 'work;employer;name;951']
[1316, 1334, 1318, 1319, 1351] feature: ['location;id;287', 'work;employer;id;970', 'work;employer;name;956', 'education;type;1']
[1316, 1334, 1318, 1319, 1327] feature: []
[1316, 1334, 1318, 1319, 1339] feature: ['locale;1', 'education;type;0', 'education;type;1', 'location;id;287', 'location;name;287\n']
[1316, 1334, 1318, 1345, 1351] feature: ['education;type;1', 'work;employer;id;965']
[1316, 1334, 1318, 1351, 1327] feature: ['location;id;287', 'work;employer;id;970', 'work;employer;name;956', 'education;type;1']
[1316, 1334, 1318, 1351, 1339] feature: ['education;type;0', 'education;type;1', 'location;id;287']
[1316, 1334, 1318, 1327, 1339] feature: ['locale;1', 'education;type;0', 'education;type;1', 'location;id;287', 'location;name;287\n']
[1316, 1334, 1349, 1319, 1345] feature: ['education;type;0', 'education;type;1', 'work;employer;id;965', 'work;employer;name;951', 'location;name;287\r
[1316, 1334, 1349, 1319, 1351] feature: ['education;type;1', 'work;employer;name;951', 'location;id;287']
[1316, 1334, 1349, 1319, 1327] feature: ['education;type;0', 'work;employer;name;951', 'location;id;287']
[1316, 1334, 1349, 1319, 1339] feature: ['education;type;0', 'education;type;1', 'work;employer;name;951', 'location;id;287', 'location;name;287\n']
[1316, 1334, 1349, 1345, 1351] feature: ['education;type;1', 'work;employer;name;951']
```

Figure 7: Final Output

Accuracy or Success Rate

With a motive to apply an approach that could effectively formulate personal social circles for users even when their network grows and to capture individual properties of communities or groups, the basic assumption for each derived circle was that there has to be some common property or aspect among the members of the derived circles.

As shown in fig 7 there are a few unexpected circle outputs which turned out to defy our assumption of at least one common aspect in a group. When calculated, it showed that a total of 9.6% circles contain an empty feature list. Clearly, these values do not hold the required criteria for our output and must be rejected or be considered as an error. So, we derived this formula for calculating the percentage error or the success rate of our analysis.

Percentage error = {(V-total - V-error) /V-total } × 100

where V-total is the total circles derived and the value is 25001

V-error is the number of circles being rejected and the value is 2419

The percentage error found for our analysis is 90.32%

Conclusion & Future Scope

- The purpose of this project is to determine personal social circles with the provided network connections of the users while keeping in mind the individual aspects of the users' communities.
- We derived a total of 25001 circles for all the users in the ego net files. Out of these approximately 10% did not contain a feature list and were eliminated. This concludes that the clique methodology gave us an accuracy of 90.32% in automatically organizing nodes into circles and extracting out the common features in them.
- The circles were given a #tooLittleFriendsInCircleThreshold of 5 IDs and #tooManyNodesThreshold of 65. We noticed that after these thresholds the size of most of the derived circles was in range five to thirteen nodes in a circle and the common features were required to be more than or equal to 1.
- We successfully extracted potential circles that could be updated if a person's social network grows, and these social circles are only limited to profile information being missing or withheld.
- The basic properties of the circle formation were:
    - Every circle should have common properties or 'aspects' among its nodes.
    - Different aspects should lead to the formation of different circles.
    - Many circles are hierarchically nested in larger ones and 'stronger' circles can form within 'weaker' ones
- Due to the overlapping circles and the fact that a single node can be present in more than one circle, we found that finding connected components cannot give satisfactory results for our problem.

In conclusion, our results can be used for suggesting potential social circles to the user and could ease up the process of manual categorization of friends within groups. This would not only help the users to organize their personal circles better but also help them to cope up with the overwhelming information generated by the friends on social networks.

Future Scope

The future scope for this project according to our analysis lies in broadening the list of features considered during circle formation. Most of our data revolve around features including similar

education, location or work. Whereas in the real world there are numerous other factors that play into people forming different social circles such as hobbies, interests, emotions and social agenda. On capturing a wider range of features that define a person, we can come up with an exhaustive list of features that creates more holistic social circles. This will help optimize the formation of social circles better and will be able to bring people together on the basis of multiple features.

References

"Learning Social Circles in Networks," *Retrieved from https://www.kaggle.com/c/learning-social-circles*.

"Cliques - Networkx Documentation," *Retrieved from https://networkx.github.io/documentation/networkx-1.11/reference/algorithms.clique.html*.

"Components - Networkx Documentation," *Retrieved from https://networkx.github.io/documentation/stable/reference/algorithms/component.html*

Julian McAuley, and Jure Leskovec "Learning to Discover Social Circles in Ego Networks" NIPS *2012*

*Hanneman, R. A., & Riddle, M. (2005). "*Introduction to social network methods*". Riverside, California*

*"*Graph Theory- Connected Components & Cliques" *Referred from* https://en.wikipedia.org/wiki/