**Detecting Tight Communities in Facebook**

**Presented By**:

Ramya Bettachari (rxb154830)

Chris Vinn Mathew (cxm154130)

Mahesh Narayan Krishnamurthy (mxk166930)

Aditi Venugopalan (axv162530)

**INDEX**

1 Introduction and Problem Description 2 Dataset Description 3 Related Work 4 Data Preprocessing

5 Work Methodology & Proposed Solution 6 Experiment and Results 7 Future Work 8 Conclusions 9 References

**1 Introduction and Problem Description**

Social networks have become a medium for people to express their opinions and share information. Especially, with more and more people embracing this powerful medium all over the world, it has become an integral part in all fields ranging from politics to education. People with similar ideologies and interests tend to involve in similar activities and be a part of similar communities in social media as well. Thus, catering to customer specific interests have become an area of business interest. This means that there arises a need to understand the people’s behaviour, activities, their associations and their interest groups, which would further help businesses like Facebook, Twitter, Google, etc. to customize their web pages specific to each individual’s requirements. Detecting tight communities also helps us in identifying hidden relationships between the users which are not explicitly available. We can use this information to analyze patterns of similar users or hidden networks / societies.

This project aims at detecting tight communities in Facebook. The Facebook data contains 10 ego-networks consisting of 88234 edges and 4039 nodes. The people in the Facebook social network are represented as nodes and the connection between two people are represented as edges in the graph that represents the entire Facebook data.

We have also tried finding the tightly coupled communities in the Twitter dataset. The Twitter data contains 1000 ego-networks consisting of 4869 circles and 81362 users. The people in the Twitter social network are represented as nodes and the connection between two people are represented as edges in the graph that represents the entire Twitter data.

# 

# 

# 

# **2 Dataset Description**

**Facebook Dataset :**

This dataset consists of 'circles' (or 'friends lists') from Facebook. Facebook data was obtained from the following link:<http://snap.stanford.edu/data/egonets-Facebook.html>

|  |  |
| --- | --- |
| Dataset distribution | |
| Nodes | 4039 |
| Edges | 88234 |
| Nodes in largest WCC | 4039 (1.000) |
| Edges in largest WCC | 88234 (1.000) |
| Nodes in largest SCC | 4039 (1.000) |
| Edges in largest SCC | 88234 (1.000) |
| Average clustering coefficient | 0.6055 |
| Number of triangles | 1612010 |
| Fraction of closed triangles | 0.2647 |
| Diameter (longest shortest path) | 8 |
| File | Description   |  |  | | --- | --- | | 90-percentile effective diameter | 4.7 | |
| [facebook.tar.gz](http://snap.stanford.edu/data/facebook.tar.gz) | Facebook data (10 networks, anonymized) |
| [facebook\_combined.txt.gz](http://snap.stanford.edu/data/facebook_combined.txt.gz) | Edges from all egonets combined |
| [readme-Ego.txt](http://snap.stanford.edu/data/readme-Ego.txt) | Description of files |

For our project, we are considering the facebook\_combined.txt.gz file and facebook.tar.gz file. The combined text file contains 88234 edges and 4039 nodes. The facebook.tar.gz file consists of ten ego networks and their circles, edges, etc. We will be working on the circles and the combined text file. In combined text file, each line consists of a pair of nodes indicating that they are friends with each other. These edges are undirected, i.e., it implies that if A is friends with B, then B is friends with A well.

**Twitter Dataset :**

This dataset was obtained from the link : http://snap.stanford.edu/data/egonets-Twitter.html

|  |  |
| --- | --- |
| Dataset distribution | |

|  |  |
| --- | --- |
| Nodes | 81306 |
| Edges | 1768149 |
| Nodes in largest WCC | 81306 (1.000) |
| Edges in largest WCC | 1768149 (1.000) |
| Nodes in largest SCC | 68413 (0.841) |
| Edges in largest SCC | 1685163 (0.953) |
| Average clustering coefficient | 0.5653 |
| Number of triangles | 13082506 |
| Fraction of closed triangles | 0.06415 |
| Diameter (longest shortest path) | 7 |
| 90-percentile effective diameter | 4.5 |

Twitter data is obtained from 1,000 ego-networks, consisting of 4,869 circles and 81,362 users. The ego-networks have 81306 nodes and the directed edges indicate person a follows person b. The set of circles for each node is all the groups the user is a part of and each circle is a set of nodes indicating all the users of that group and their connection. These circles were obtained by using the features mentioned in the feature name file.

We used the file that contained all the edges i.e. all the connections between every user in the Twitter data to construct a graph of the social network with each user as a node and the connection of that user with other users as an edge in the graph.

# **3 Related Work**

Identifying relationships among social media users has turned out to be one of the hot topics of research in the field of Computer Science. The amount of information one can extract from these relationships is priceless and is necessary to shape any successful business strategies. Great research work has been going on in this field in top institutions such as Carnegie Mellon and Stanford Universities.

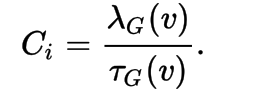
# **4 Dataset Preprocessing**

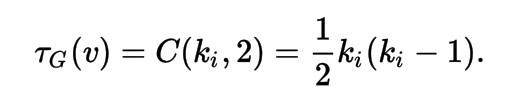
The combined textfiles from twitter and facebook datasets were pre-processed. The unwanted text contents from these files were removed and then fed as input to the RDD in Spark.

# **5 Work Methodology & Proposed Solution**

We started with the Facebook dataset and tried identifying tightly connected communities. This problem translates to identifying connected components, strongly connected components, and cliques in the graph. In any undirected graph, a connected component is a subgraph in which any two vertices are **connected** to each other by paths, and which is **connected** to no additional vertices in the main graph. Hence, we identified the connected components in the graph. Then, we tried identifying the strongly connected components. A graph is said to be **strongly connected** if there is a path between all pairs of vertices. A **strongly connected component** (SCC) of a directed graph is a maximal **strongly connected** subgraph. But, since the dataset was so huge, it didn’t run to completion and kept running out of memory. We also tried the same in Databricks, UTD Cluster, Cloudera and Sandbox. But no cluster had enough memory to compute the same. Then, we tried finding cliques in the graph. A **clique** is a subset of vertices of an undirected **graph,** such that any pair of vertices from the subset are connected by an edge. But since, it’s an NP-Complete problem, we couldn’t compute cliques of size greater than three in the Facebook dataset. So, we proceeded by identifying the **clustering coefficient** of each node, a measure of the degree to which nodes in a graph tend to **cluster** together.The **local clustering coefficient** of anode in agraph quantifies how close itsneighbours are to being aclique.Filtering nodes based on clustering coefficient will remove the outliers. Finding connected components on the filtered nodes will give tightly coupled components. Thus, it helped us in identifying the class of nodes that are tightly coupled.

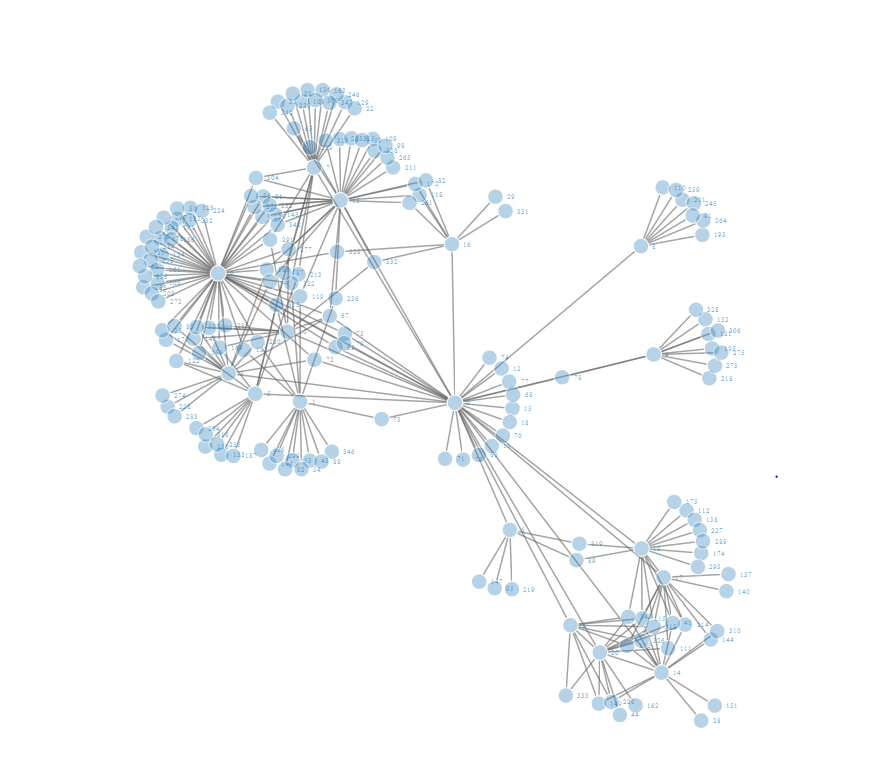
**Formula to find the clustering coefficient ‘C*i*’ at node *i*  with *ki* neighbours :**

Where

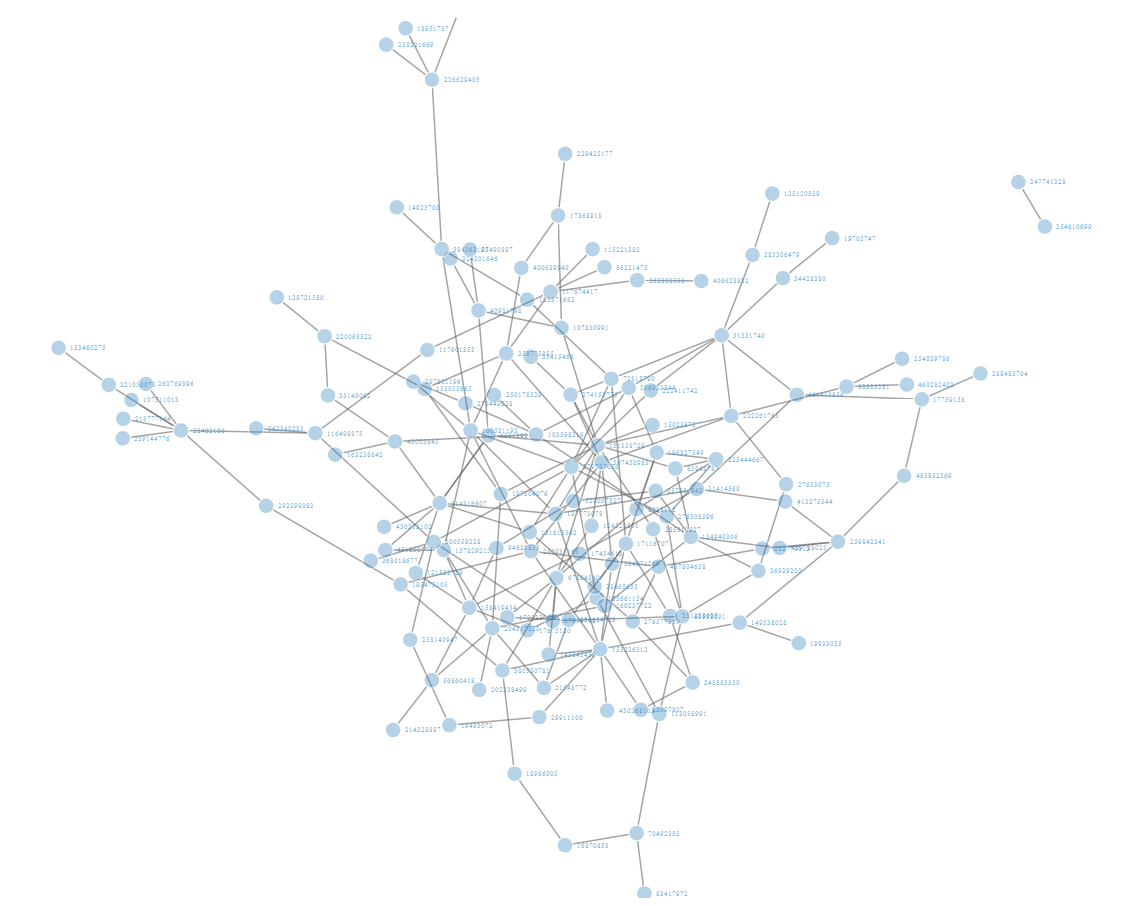




Then, we started working on the Twitter dataset and followed a similar approach. We identified the connected and strongly connected components in the Twitter dataset. Then, we also identified the clustering coefficients, and thereby, identified the tightly coupled nodes.



**A part of the facebook dataset**



**A part of the twitter dataset**

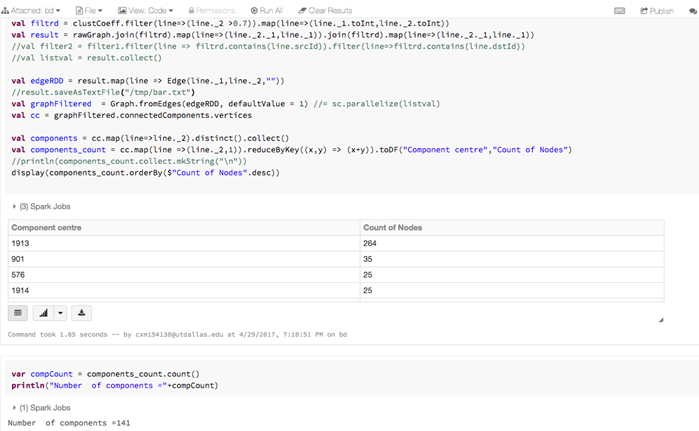
**6 Experiment and Results**

**Results for the Facebook Dataset:**

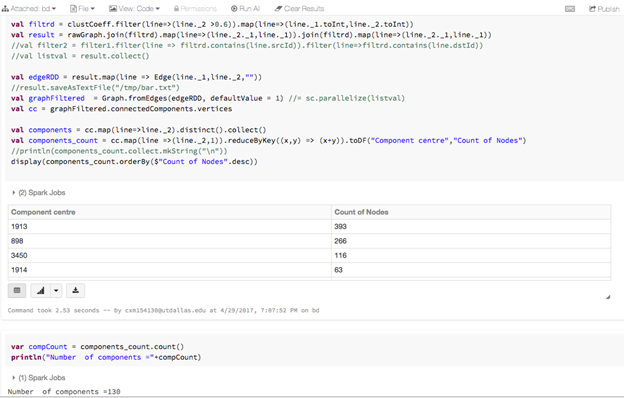
|  |  |
| --- | --- |
| **CLUSTERING COEFFICIENT THRESHOLD** | **Number of Connected Components** |
| 0.7 | 141 |
| 0.6 | 130 |
| 0.5 | 80 |
| 0.4 | 47 |
| 0.3 | 30 |
| 0.2 | 19 |

**Scala code and output:**

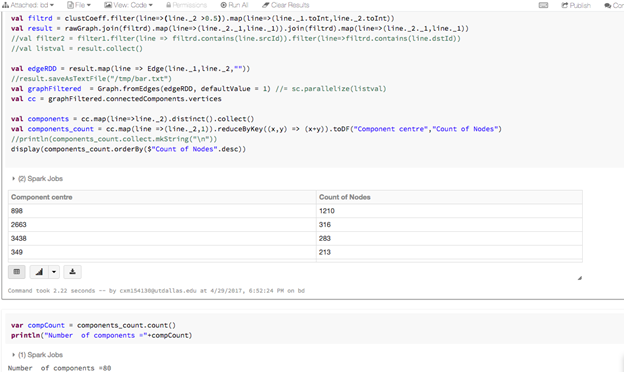
For nodes with clustering coefficient threshold > 0.7



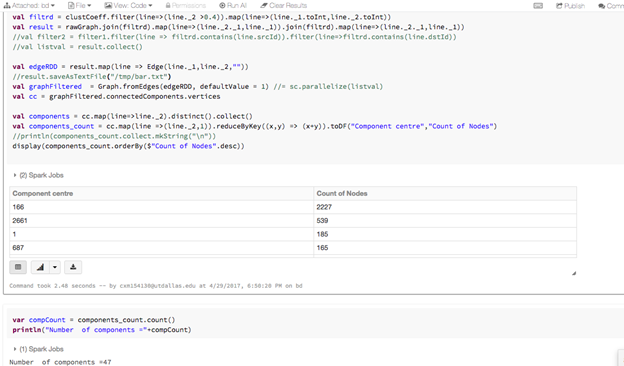
For nodes with clustering coefficient hold > 0.6



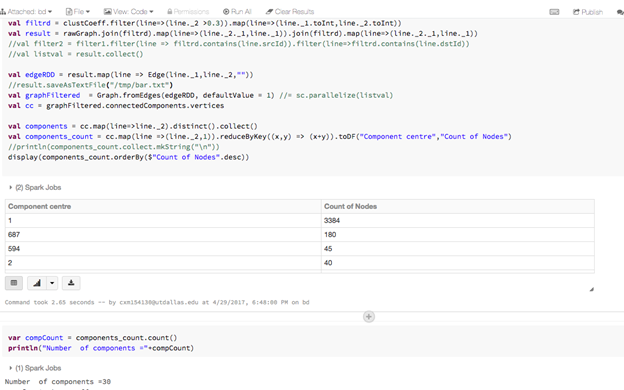
For nodes with clustering coefficient threshold > 0.5



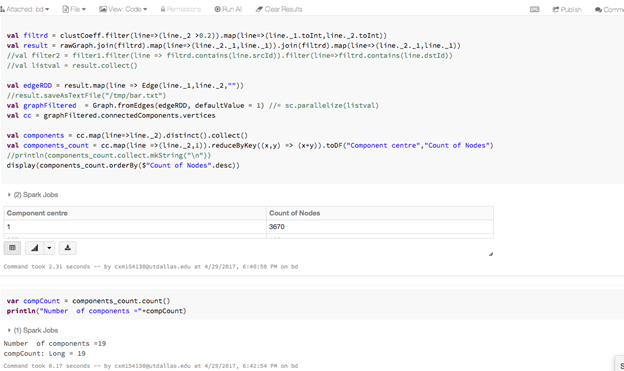
For nodes with clustering coefficient threshold > 0.4



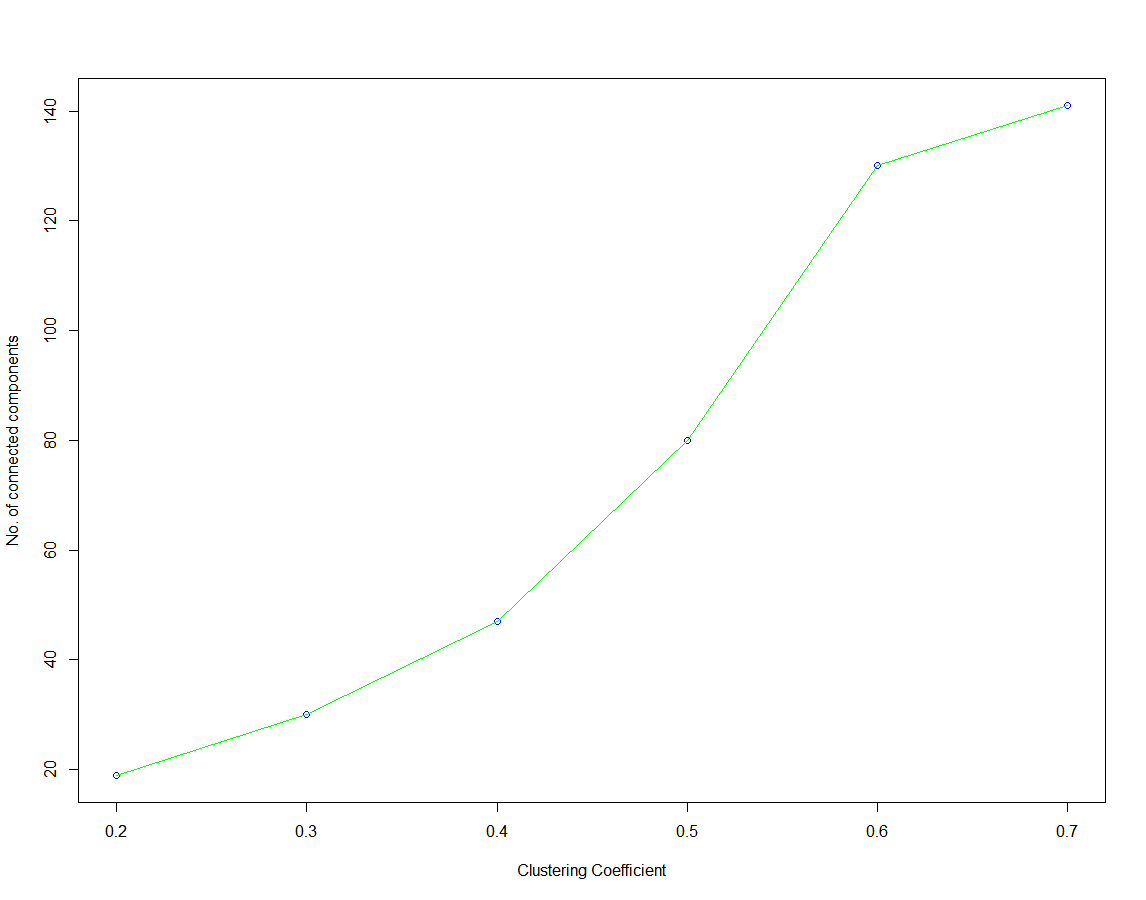
For nodes with clustering coefficient threshold > 0.3



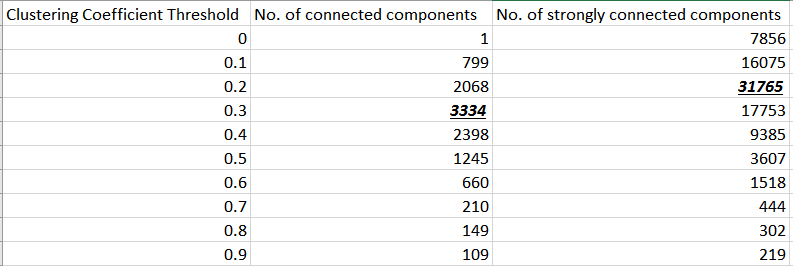
For nodes with clustering coefficient threshold > 0.2



**Plot of clustering coefficient vs no. of connected components**

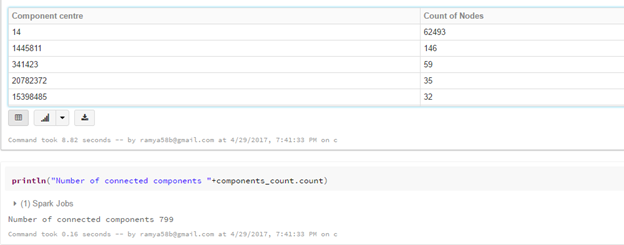


**Results for the Twitter Dataset:**



For nodes with clustering coefficient > 0.1

Connected components:



Strongly connected components:

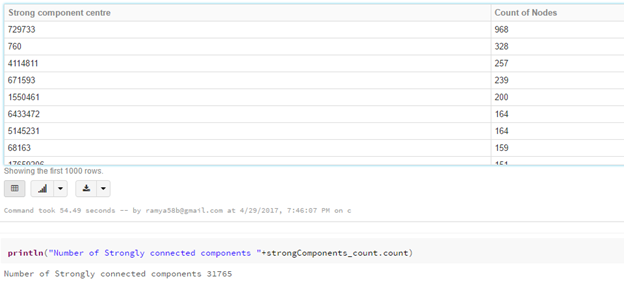


For nodes with clustering coefficient > 0.2

Connected components:

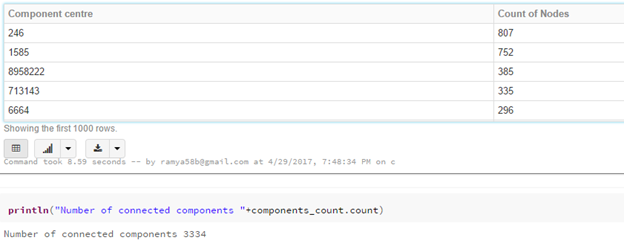


Strongly connected components:



For nodes with clustering coefficient > 0.3

Connected components:



Strongly connected components:

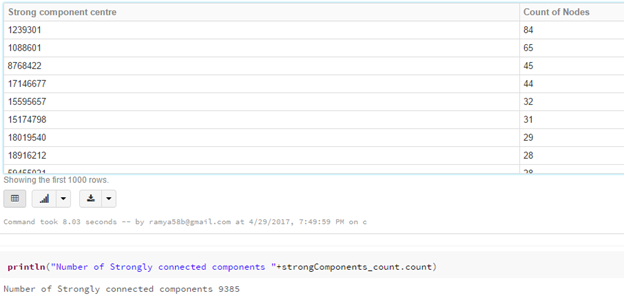


For nodes with clustering coefficient > 0.4

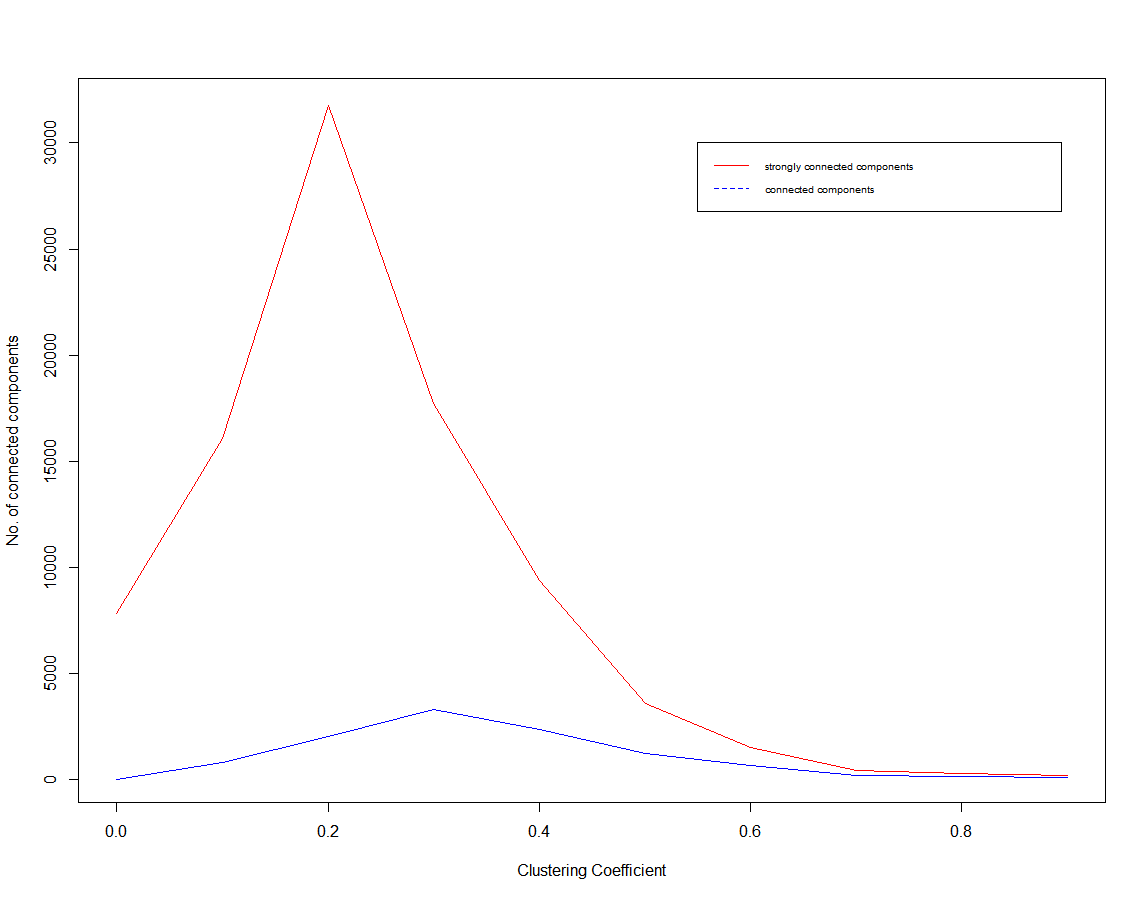
Connected components:



Strongly connected components:



Plot of clustering coefficient vs no. of connected and strongly connected components



From the above results for twitter dataset, it is seen that the number of components increases as the clustering coefficient increases, until a certain point where the number of components starts to decrease. This point is reached at clustering coefficient = 0.3 for connected components and clustering coefficient = 0.2 for strongly connected components, points where the number of components is at the maximum.

**7 Future Work**

Our system provides an easier methodology to find out the degree of tightness in a community, using connected, strongly connected components and cluster coefficients in Facebook and Twitter datasets. But, this might not be a centralized approach that can be followed in all other Social Network platforms. Moreover, we need to find a mechanism to identify cliques of any generic size in the datasets. Therefore, a more generalized approach is needed, that would work with all networks and provides the most optimum results.

**8 Conclusions**

We have identified the tightly coupled communities in the Facebook and Twitter datasets using certain simple techniques in graph theory. We have identified the tight coupling based on only the connection between different nodes in the network. All the results were obtained in a fairly acceptable time-frame.

# 

# **9 References**

1. <http://dept.stat.lsa.umich.edu/~jizhu/pubs/Zhao-PNAS11.pdf>

2. <http://www.cise.ufl.edu/~mythai/files/socialcom11.pdf>

3. <http://spark.apache.org/graphx/>

4. <http://i.stanford.edu/~julian/pdfs/nips2012.pdf>

5. <http://www.cs.princeton.edu/~yingyul/ModelingandDetectingCommunityHierarchies.pdf>

6. <https://www.infoq.com/articles/apache-spark-graphx>

7. <http://www.hamrtech.com/assets/downloads/K-Cliques.pdf>

7. <http://spark.apache.org/docs/latest/graphx-programming-guide.html#connected-components>

8. T. Yoshida. Toward finding hidden communities based on user profiles. In ICDM Workshops, 2010.