

**Department of Master of Computer Applications**

**MCAE24 Social Network Analysis**

**Case Study – Description**

**Title: Social Circle- Twitter**

**By**

**(1MS22MC016) Komal S Kallanagoudar**

**(1MS22MC057) Yamuna S**

**Description of the Case Study**

**Problem Statement:**

We define a novel machine learning task of identifying users’ social circles. We pose the problem as a node clustering problem on a user’s ego-network, a network of connections between friends.

**Objective:**

The objective of this case study is to Conduct network analysis to identify nodes (Twitter users) and edges (connections between users) within each ego network, develop a model or algorithm for detecting social circles within each ego network, taking into account the network structure and user profile information and allow for the possibility of overlapping circles, as Twitter users can belong to multiple social groups or communities simultaneously.

Also learn an effective circle-specific user profile similarity metric that quantifies the similarity between users within the same circle. The goal is to discover these circles given only the network between the ego’s friends.

**Dataset Description:**

From the dataset we obtained profile and network data from 10 ego-networks, consisting of 193 circles and 4,039 users.

**Methodology:**

1. **Data Collection:** Given a single user with personal social network, our goal is to identify their circles, each of which is a subset of her friends. Circles are user-specific as each user organizes her personal network of friends independently of all other users to whom they are not connected.
2. **Data Preprocessing:** Before diving into analysis, we had to roll up our sleeves and clean up the data. We combed through it to deal with any missing values, duplicates, and ensure uniform data formats. We also selected the most pertinent attributes to make the dataset sleek and agile for analysis.
3. **Centrality Computation:** The heart of our study lay in computing various centrality measures for each user in the dataset. These measures included degree centrality, which gauged the number of connections a user had, betweenness centrality which looked at friendship between user and his/her friends.
4. **Network Visualization:** To truly appreciate the intricate web of relationships among users, we employed the 'nx\_draw' function from the networkx library. This nifty tool conjured up vivid visualizations of the twitter user’s network. In these visuals, each node represented a user, and the edges between them depicted their connections.
5. **Importing data set:**

import pandas as pd #For reading dataset files

import networkx as nx #For network creation/analysis

import matplotlib.pyplot as plt #For plotting graphs

%matplotlib inline

df = pd.read\_csv('twitter\_combined.txt.gz')

df.info()

df.head()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 2420765 entries, 0 to 2420764

Data columns (total 1 columns):

# Column Dtype

--- ------ -----

0 214328887 34428380 object

dtypes: object(1)

memory usage: 18.5+ MB

|  |  |
| --- | --- |
|  | 214328887 34428380 |
| 0 | 17116707 28465635 |
| 1 | 380580781 18996905 |
| 2 | 221036078 153460275 |
| 3 | 107830991 17868918 |
| 4 | 151338729 222261763 |

1. **Subset:**

subset=df.sample(n=100)

print (subset.head())

print (subset.tail())

nx.draw\_networkx(subset)

|  |  |  |
| --- | --- | --- |
|  | 214328887 | 34428380 |
| 2145181 | 17817210 | 15433693 |
| 158298 | 83533045 | 18823495 |
| 1190052 | 61730210 | 260846408 |
| 1869171 | 37214134 | 42361118 |
| 1985346 | 42523468 | 80797776 |
|  | 214328887 | 34428380 |
| 539610 | 1035491 | 8417802 |
| 1579065 | 285359479 | 496438192 |
| 949191 | 18811197 | 250303199 |
| 1491563 | 118534712 | 62513246 |
| 1239906 | 204317520 | 208132323 |

1. Importing data set (based on user):

import pandas as pd

df = pd.read\_csv('twitter\_combined.txt.gz',sep=" ",header=None,names=["User1","User2"]);

df.info()

df.tail()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 2420766 entries, 0 to 2420765

Data columns (total 2 columns):

# Column Dtype

--- ------ -----

0 User1 int64

1 User2 int64

dtypes: int64(2)

memory usage: 36.9 MB

|  |  |  |
| --- | --- | --- |
|  | User1 | User2 |
| 2420761 | 99841247 | 154263215 |
| 2420762 | 99841247 | 194403468 |
| 2420763 | 99841247 | 180165101 |
| 2420764 | 99841247 | 253509115 |
| 2420765 | 99841247 | 463410501 |

1. **Subset of User:**

subset=df.sample(n=100)

print (subset.head())

print (subset.tail())

|  |  |  |
| --- | --- | --- |
|  | User1 | User2 |
| 1969178 | 275003449 | 15811779 |
| 1591178 | 106955278 | 28149277 |
| 1234512 | 77618627 | 204317520 |
| 1377091 | 123480187 | 149276260 |
| 1710152 | 18021714 | 24004172 |
|  | **User1** | **User2** |
| 450863 | 17526132 | 16186995 |
| 1393123 | 536893070 | 302374422 |
| 1064810 | 18220380 | 14855064 |
| 1483093 | 190286462 | 143540685 |
| 1031734 | 131537442 | 212247905 |

1. **Edge List:**

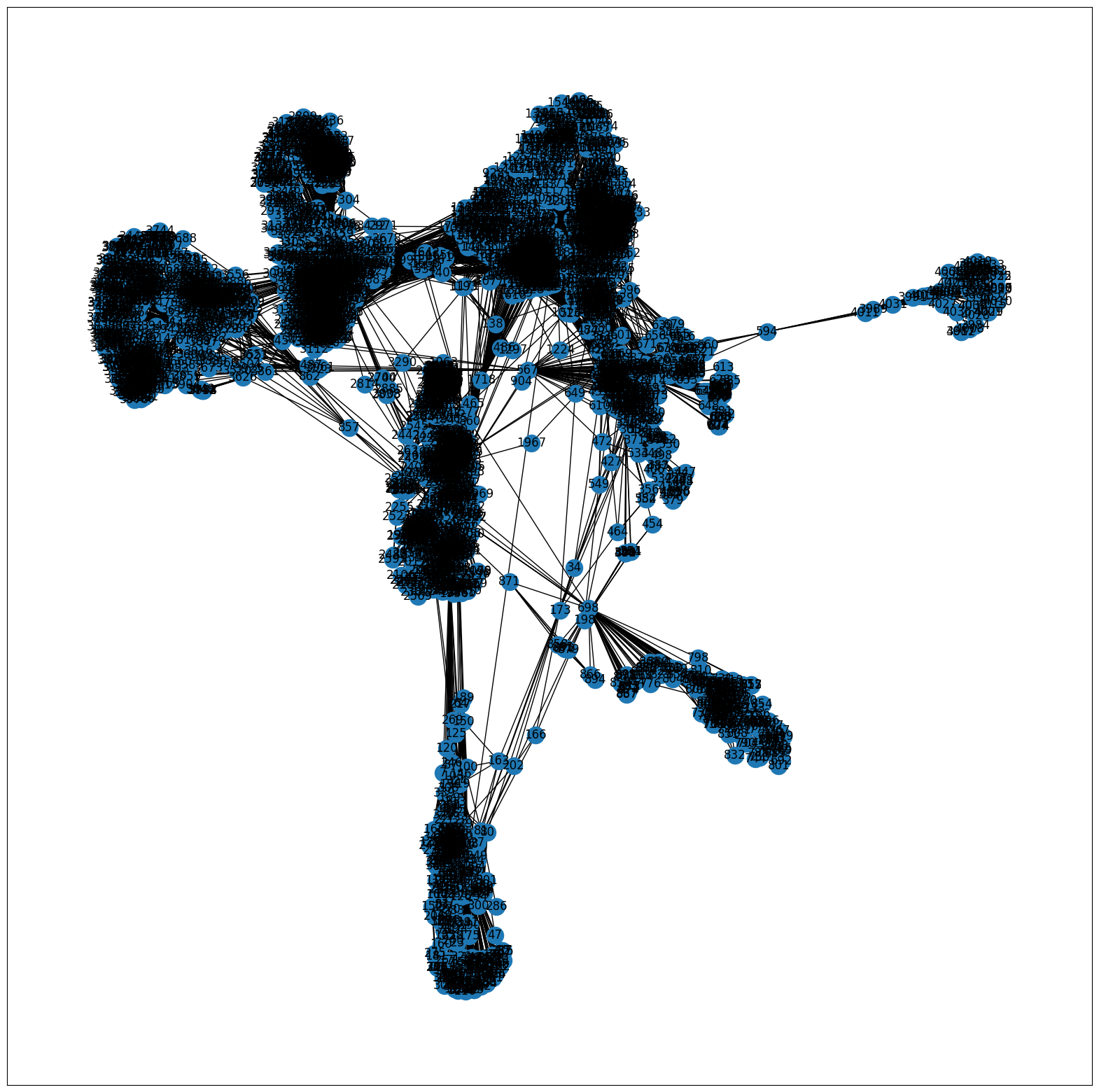
import networkx as nx

import matplotlib.pyplot as plt

G\_t=nx.read\_edgelist('twitter\_combined.txt.gz',create\_using=nx.Graph(),nodetype = int)

plt.figure(figsize=(7,7))

nx.draw\_networkx(G\_t)



1. **Description of Data:**

print (df.describe())

|  |  |  |
| --- | --- | --- |
|  | 214328887 | 34428380 |
| count |  | 2420765 |
| unique |  | 1768149 |
| top | 43003845 | 40981798 |
| freq |  | 78 |

1. **First and Last 5 records:**

sg=df.sample(n=15)

print (sg.head())

print (sg.tail())

|  |  |  |
| --- | --- | --- |
|  | 214328887 | 34428380 |
| 479506 | 28201743 | 14287830 |
| 1369324 | 20471349 | 322822476 |
| 343373 | 115710058 | 48037533 |
| 2052291 | 43472595 | 39614123 |
| 1424229 | 3332291 | 812126 |
|  | 214328887 | 34428380 |
| 2057775 | 14949810 | 1715051 |
| 659884 | 216665512 | 76027335 |
| 1950592 | 23371095 | 56903665 |
| 292540 | 15989218 | 18197944 |
| 195366 | 185457171 | 366795440 |

1. Betweenness Centrality:

pos = nx.spring\_layout(df)

betCent = nx.betweenness\_centrality(df, normalized=True,

endpoints=True)

node\_color = [20000.0 \* df.degree(v) for v in df]

node\_size = [v \* 10000 for v in betCent.values()]

plt.figure(figsize=(20,20))

nx.draw\_networkx(df, pos=pos, with\_labels=False,node\_color=node\_color,node\_size=node\_size )

plt.axis('off');



1. **Sorted List:**

sorted(betCent, key=betCent.get, reverse=True)[:5]

[107, 1684, 3437, 1912, 1085]

1. **Edge List:**

FG = nx.from\_pandas\_edgelist(df, source='User1', target='User2')

nx.draw\_networkx(FG)

