SOCIAL NETWORK ANALYSIS PROJECT REPORT

Team Members :

MD Wazid Ansari (2K19/AE/037)

Bhavishya (2K19/SE/

Abstract

We describe our work in the collection and analysis of massive data describing the connections between participants to online social networks.A large sample comprising thousands of connections, have been collected; the data is anonymous and organized as an undirected graph.We describe a set of tools that we developed to analyze specific properties of such socialnetwork graphs, i.e., among others, degree distribution, centrality measures, scaling laws and distribution of friendship.

I. INTRODUCTION

A Social network is defined as a network of relationships or interactions, where the nodes consist of people or actor, and the edges or archs consist of the relationships or interactions between these actors. Social networks and the techniques to analyse them existed since decades. There can be several type of social networks like email network, telephone network, collaboration network. But recently online social networks like Facebook, Twitter, LinkedIn, MySpace etc have been developed which gained popularity within very short amount of time and gathered large number of users. Facebook is said to have more than 500 million users in 2010.

The field of social networks and their analysis has evolved from graph theory, statistics and sociology and it is used in several other fields like information science, business application, communication, economy etc. Analysing a social network is similar to the analysis of a graph because social networks form the topology of a graph.

In this paper, some graph analysis tools for the analysis of large online social networks are discussed and compared.

II. SOCIAL NETWORK ANALYSIS

Social network analysis (SNA) is the methodical analysis of social networks. Social network analysis views social relationships in terms of network theory, consisting of nodes (representing individual actors within the network) and ties (which represent relationships between the individuals, such as friendship, kinship, organizational position, sexual relationships, etc.).

Analysis tasks of social networks includes following:

- Discovering the structure of social network
- Finding various attribute values for the network- Ex.radius, diameter, centrality, betweenness, shortest paths, density etc
- Finding communities in the social network
- Visualizing the whole or part of the social network

III. LITERATURE REVIEW

- Zachary, in his PhD thesis,formalized the first model of a real-life social network. Then,social networks started attracting the interest of different sciences, including Computer Science.
- Degree centrality was proposed by Professor Linton, which reflects local properties of the network, and the main consideration is the node itself and the neighbors properties. Although the calculation of degree centrality is simple, it has some deficiencies. #### Betweenness centrality, closeness centrality, and eigenvector centrality reflect the global property of networks. Among them,
  - betweenness centrality mainly considered the shortest path through the node.
  - Closeness centrality measures the difficulty to reach the other node.
  - Eigenvector centrality mainly considered the status and prestige in the networks using the composition of the reputation of other nodes to reflect the influence of the node for the entire network.
  - K-shell centrality reflects the nodes location within networks to measure node communication capacities .

Several works has been done on various social networks to analyse and discover various kinds of relationships and information [7][8][9][10].

IV. OBJECTIVES

- Importing Datasets and describing them.
- Working appropriate centrality measures, clustering coefficients (both local and global) and reciprocity and transitivity.
- Inferences from the output of above values.
- Information Cascading effect in the network.

V. METHODOLOGY

For our project purpose we used certain analyzing tools to analyze out dataset

- Language - Python
- Tool - Jupyter Notebook
- Library - networkx

Description of Datasets:

Social circles: Facebook

Dataset information :
This dataset consists of 'circles' (or 'friends lists') from Facebook. Facebook data was collected from survey participants using this Facebook app. The dataset includes node features (profiles), circles, and ego networks.

Facebook data has been anonymized by replacing the Facebook-internal ids for each user with a new value. Also, while feature vectors from this dataset have been provided, the interpretation of those features has been obscured. For instance, where the original dataset may have contained a feature "political=Democratic Party", the new data would simply contain "political=anonymized feature 1". Thus, using the anonymized data it is possible to determine whether two users have the same political affiliations, but not what their individual political affiliations represent.

In [0]:
```
# Importing used libraries for this social network graph study
import networkx as nx
import matplotlib.pyplot as plt
import csv
import random
import matplotlib.colors as mcolors
import random
import pandas as pd
import datetime
import io
import array,re,itertools
import numpy as np
import math
```

In [0]:
```
import warnings
import matplotlib.cbook
warnings.filterwarnings("ignore",category=matplotlib.cbook.mplDeprecation)
```

Reading Facebook Social Circle Dataset and displaying it's features.

In [0]:
```
# Taking (Social Circles)Facebook Dataset in graph and displaying it's features.
g_fb = nx.read_edgelist('/content/drive/My Drive//facebook_combined.txt', nodetype=int
print (nx.info(g_fb))
```

```
Name:
Type: DiGraph
Number of nodes: 4039
Number of edges: 88234
Average in degree:  21.8455
Average out degree:  21.8455
```

In [0]:
```
# Sampling Social Circles Facebook Dataset
result = []
F = open('/content/drive/My Drive/Colab Notebooks/Dataset/facebook_combined_Sample.txt
with open('/content/drive/My Drive/Colab Notebooks/Dataset/facebook_combined.txt') as
    data = fh.readlines()
    for line in data:
        if line:
            words = line.split()
            result.append(words)
sample = random.sample(result, 20000)

for item in sample:
    F.write(item[0] + "\t" + item[1] + "\n")
print ("Sampled Dataset Information:")
G_sampled = nx.read_edgelist('/content/drive/My Drive/Colab Notebooks/Dataset/faceboo
print (nx.info(G_sampled))
```

```
Sampled Dataset Information:
Name:
Type: DiGraph
Number of nodes: 3671
Number of edges: 17784
Average in degree:   4.8445
Average out degree:   4.8445
```

VI. ANALYSIS

Working on centrality measures, clustering coefficients on samples for both datasets.

In [0]:
```
# Social Circles Facebook Dataset Sample
pos = nx.spring_layout(G_sampled)

# Degree Centrality
degCent = nx.degree_centrality(G_sampled)

# Betweeness Centrality
betCent = nx.betweenness_centrality(G_sampled, normalized=True, endpoints=True)

# Eigen vector centrality
eig_cen=nx.eigenvector_centrality(G_sampled,1000000) # 1000000-->precision
key_max = max(eig_cen.keys(), key=(lambda k: eig_cen[k]))
print('Maximum eigen-vector centrality:',eig_cen[key_max],end=" ")
print('at node =',key_max)

# Katz centrality
l=nx.katz_centrality(G_sampled, alpha=0.1, beta=1.0, max_iter=1000, tol=1.0e-6, nstart
l_max = max(l.keys(), key=(lambda k: l[k]))
print('Maximum Katz centrality:',l[l_max],end=" ")
print('at node =',l_max)

# Clustering Coefficients
print("Clustering Coefficient of Sample (GLobal)")
cc1_sampled = nx.average_clustering(G_sampled)
print (cc1_sampled)
#print("Clustering Coefficient of Sample (Local)")
c1_sampled = nx.clustering(G_sampled)
#print (c1_sampled)

# Reciprocity and Transitivity
print ("Reciprocity of Facebook Sample")
r = nx.overall_reciprocity(G_sampled)
print(r)
print ("Transitivity of Facebook Sample")
t = nx.transitivity(G_sampled)
print(t)
```

```
Maximum eigen-vector centrality: 0.6912190320792618 at node = 2654
Maximum Katz centrality: 0.19985693904023555 at node = 2624
Clustering Coefficient of Sample (GLobal)
0.049808151514214145
Reciprocity of Facebook Sample
0.0
Transitivity of Facebook Sample
0.04155047406581149
```

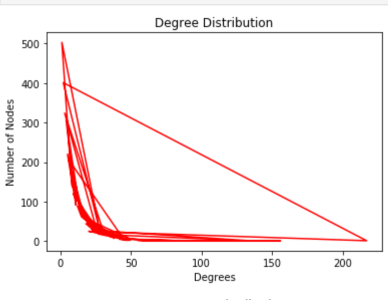
Identifing nodes with maximum in-degrees and list of some nodes with their in-degrees for both dataset samples.

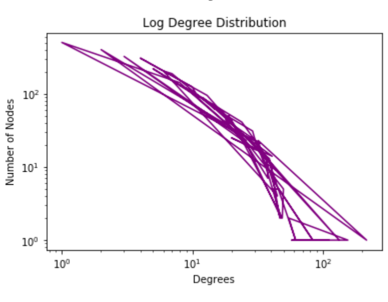
In [0]:
```
# Social Circles Facebook Dataset Sample
dict1 = G_sampled.in_degree()
print("The nodes with maximum in-degree are :")
i = max(nx.in_degree_centrality(G_sampled),key=(nx.in_degree_centrality(G_sampled)).ge
val = dict1[i]

print("Node   inDegree")
for p,r in dict1:
    if(r == val):
        print(str(p) + "\t" + str(r))
print(" ")
print("List of some nodes with their in-degree:")
w = 1
for p,r in dict1:
    if(w <= 5):
        print(str(p) + "\t" + str(r))
        w = w+1
```

```
The nodes with maximum in-degree are :
Node    inDegree
2624    56

List of some nodes with their in-degree:
2348    22
2638    44
2917    4
3129    6
525     13
```

Showing Degree Distribution and Log Degree Distribution for both dataset samples.

In [0]:
```
# Social Circles facebook Dataset sample
d = dict()
for x,y in nx.degree(G_sampled):
    if(y not in d):
        d[y]=1
    else:
        d[y]+=1
plt.plot(d.keys(),d.values(),"red")
plt.xlabel("Degrees")
plt.ylabel("Number of Nodes")
plt.title("Degree Distribution")
plt.show()
plt.loglog(d.keys(),d.values(),"purple")
plt.xlabel("Degrees")
plt.ylabel("Number of Nodes")
plt.title("Log Degree Distribution")
plt.show()
```





Information Cascading effect in the network.

In [0]:
```
# Social Circle Facebook Dataset Sample (Information Cascading effect)
print (nx.info(G_sampled))
```

```
Name:
Type: DiGraph
Number of nodes: 3671
Number of edges: 17784
Average in degree:   4.8445
Average out degree:   4.8445
```

An information cascade is defined as a piece of information or decision being cascaded among a set of individuals, where

- individuals are connected by a network and
- individuals are only observing decisions of their immediate neighbors (friends).

The **independent cascade model (ICM)** that can be utilized to model information cascades. Variants of this model have been discussed in the literature. Below assumptions for this model include the following:

- The network is represented using a directed graph. Nodes are actors and edges depict the communication channels between them. A node can only influence nodes that it is connected to.
- Decisions are binary – nodes can be either active or inactive. An active nodes means that the node decided to adopt the behavior, innovation, or decision.
- A node, once activated, can activate its neighboring nodes.
- Activation is a progressive process, where nodes change from inactive to active, but not vice versa.

Reuirements :

- Diffusion graph G(V, E), set of initial activated nodes A(0) , activation probabilities P v,w

In [0]:
```
initial_activated_nodes = random.randint(1,500)
nodes = list(G_sampled.nodes)
print(initial_activated_nodes)
```

```
295
```

Set of initial activated nodes is mentioned below:-

In [0]:
```
activated_nodes = set()
while(len(activated_nodes) != initial_activated_nodes):
    rand_idx = random.randint(1,125)
    activated_nodes.add(nodes[rand_idx-1])
print(activated_nodes)
```

In [0]:
```
remaining_activation = list(activated_nodes)
```

Now, we'll implement **Independent Cascade Model (ICM)** as mentioned in Zafarani to activate remaining nodes taking 0.2 as activation probability P v,w.

In [0]:
```
while(len(remaining_activation)):
    node = remaining_activation[0]
    remaining_activation.remove(node)
    nbrs = G_sampled.neighbors(node)
    for child in nbrs:
        if child not in activated_nodes:
            prob = random.uniform(0,1)
            if prob < 0.2:
                activated_nodes.add(child)
                remaining_activation.append(child)
```

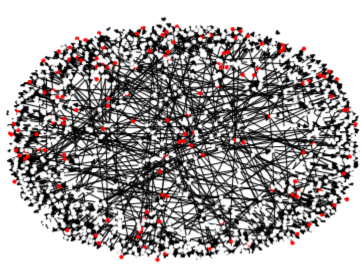
In [0]:
```
print("Initially Randomly Activated Nodes : " + str(initial_activated_nodes))
print("Number of nodes finally activated by applying ICM = " + str(len(activated_nodes
```

```
Initially Randomly Activated Nodes : 387
Number of nodes finally activated by applying ICM = 144
```

Coloring of nodes to label them as activated and inactive.

- Red = activated
- White = inactive

In [0]:
```
colored = []
for nodes in G_sampled.nodes():
    if nodes in activated_nodes:
        colored.append('red')
    else:
        colored.append('white')
nx.draw(G_sampled, with_labels = False, node_size= 10 , node_color = colored)
```



VII. INFERENCE

The dataset graph contains number of edges low as compared to general graph. That means our graph have low density.

Dataset :
Type: Graph
Number of nodes: 4039
Number of edges: 88234
Average degree : 43.6910
For each dataset, we have plotted degree distribution and log distribution, from that we come to a conclusion that both directed graphs have many nodes with very less degrees.

For studing these directed graphs, Firstly we have made simple random sample from given dataset and then we calculated Degree Centrality, Eigen vector centrality, Katz centrality and Betweenness Centrality. Moreover, we have calculated Clustering Coefficient (both local and global), Reciprocity and Transitivity for each graph which can be seen above.

Using above plots, we have identified hubs in these real graphs. Hubs are the nodes which are less in number but have high degrees.

VIII. REFERENCES

- [1] Measurement and Analysis of Online Social Networks by Alan Mislove,Massimiliano Marcon,Krishna P. Gummadi, Max Planck Institute for Software Systems.
- [2] For a historical overview of the development of social network analysis,see: Carrington, Peter J. & Scott, John (2011). "Introduction". The Sage Handbook of Social Network Analysis. SAGE. p. 1. ISBN 978-1-84787-395-8.
- [3] http://www.facebook.com/notes/facebook/500millionstories/409753352130
- [4] David Combe, Christine Largeron, El˝od Egyed-Zsigmond and Mathias Géry, “A comparative study of social network analysis tools”, International Workshop on Web Intelligence and Virtual Enterprises 2(2010)
- [5] M A huisman, MAJ van duijn, “ Software for social network analysis”, pages 270-316, 2004
- [6] Graph and Network Analysis Dr. Derek Greene Clique Research Cluster, University College Dublin, Web Science Doctoral Summer School 2011.
- [7] Monclar, Rafael Studart, et al. "Using social networks analysis for collaboration and team formation identification." Computer Supported Cooperative Work in Design (CSCWD), 2011 15th International Conference on. IEEE, 2011.
- [8] Nadeem Akhtar, Hira Javed, Geetanjali Sengar, "Analysis of Facebook Social Network", IEEE International Conference on Computational Intelligence and Computer Networks (CICN), 27-29 September, 2013, Mathura, India.
- [9] Zelenkauskaite, Asta, et al. "Interconnectedness of complex systems of internet of things through social network analysis for disaster management." Intelligent Networking and Collaborative Systems (INCoS), 2012 4th International Conference on. IEEE, 2012.
- [10] Li, Jianfeng, Yan Chen, and Yan Lin. "Research on traffic layout based on social network analysis." Education Technology and Computer(ICETC), 2010 2nd International Conference on. Vol. 1. IEEE, 2010.
- [11] Online Social Networks: Measurement, Analysis, and Applications to Distributed Information Systems by Alan E. Mislove, Houston, Texas
- [12] Social Networks Overview: Current Trends and Research Challenges” November 2010 Coordinated by the —NextMEDIA CSA.
- [13] Business Application of Social Network Analysis BASNA-2013 www.basna.in
- [14] International network of Social Network Analysis INSNA www.insna.org
- [15] Networkx http://Networkx.lanl.gov/index.html
- [16] Gephi https://gephi.org
- [17] Pajek vlado.fmf.uni-lj.si/pub/networks/pajek
- [18] IGraph IGraph.sourceforge.net
- [19] snap.stanford.edu/data