Social Media Workshop

Assignment-03

NetworkX Lab

NetworkX is a Python language software package for the creation, manipulation, and study of the structure, dynamics, and function of complex networks . NetworkX is used by developers to process and perform functionalities on various social media applications.

With NetworkX you can load and store networks in standard and nonstandard data formats, generate many types of random and classic networks, analyze network structure, build network models, design new network algorithms, draw networks, and much more.

You need to install two Python packages using pip install, Packages are- networkx, Matplotlib **(take .whl file of this from resources from study material)**

**Practice following python codes using above mentioned packages and check the results of programs.**

**Sample code 1:shortest path code:**

```
import networkx as nx
import matplotlib.pyplot as plt
G = nx.karate_club_graph()
pos = nx.spring_layout(G)
nx.draw(G,pos,node_color='k')
# draw path in red
path = nx.shortest_path(G,source=14,target=16)
path_edges = zip(path,path[1:])
nx.draw_networkx_nodes(G,pos,nodelist=path,node_color='r')
nx.draw_networkx_edges(G,pos,edgelist=path_edges,edge_color='r',width
=10)
plt.axis('equal')
plt.show()
```

**Sample code 2: Read data from a csv file: (author.csv file is given in resources folder on study material**

```
import csv
import networkx as nx
import matplotlib.pyplot as plt

with open('author.csv','rt') as authorIn: #reads in the author-
network file
 authorIn = csv.reader(authorIn)
 headers = next(authorIn)
 authors = [row for row in authorIn]
 uniqueauthors = list(set([row[0] for row in authors])) #takes the
reduantices of the superhero networks
 id=list(enumerate(uniqueauthors))# creates a list of tuples with
unique ids and their names for each superhero in the network
 keys = {name: i for i, name in enumerate(uniqueauthors)} #creates a
dictionary(hash map) that maps each id to the superhero names
 links = [] #creates a blank list
 for row in authors: #maps all of the names in the csv file to their
id number
  try:
   links.append({keys[row[0]]: keys[row[1]]})
  except:
   links.append({row[0]: row[1]})
 G = nx.Graph() #creates a graph
 authorNodeId=[] #takes source and target edges
 for row in id:
   authorNodeId.append(row[0])
   G.add_nodes_from(authorNodeId)#creates nodes for the graph.
 for node in links:#loops through each link and changes each
dictionary to a tuple so networkx can read in the information
   edges = node.items()
G.add_edge(*edges[0])#takes the tuple from the list and unpacks the
tuples
nx.draw_networkx(G)
plt.show(G)
```

**Show labels in the network using the below given code:**

```
pos=nx.spring_layout(G)
for p in pos:  # raise text positions
        pos[p][1] += 0.7
nx.draw_networkx_labels(G, pos)
```

**Sample code 3: Circular Tree network Sample Code:**

```
import networkx as nx
import matplotlib.pyplot as plt
try:
    from networkx import graphviz_layout
except ImportError:
    raise ImportError("This example needs Graphviz and either
PyGraphviz or Pydot")
G=nx.balanced_tree(3,5)
pos=nx.graphviz_layout(G,prog='twopi',args='')
plt.figure(figsize=(8,8))
```

```
nx.draw(G,pos,node_size=20,alpha=0.5,node_color="blue",
with_labels=False)

plt.axis('equal')

plt.savefig('circular_tree.png')

plt.show()
```

**Sample code 4: Ego network Sample Code:**

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
"""
Example using the NetworkX ego_graph() function to return the main
egonet of
the largest hub in a Barabási-Albert network.
"""
__author__="""Drew Conway (drew.conway@nyu.edu)"""

from operator import itemgetter
import networkx as nx
import matplotlib.pyplot as plt


if __name__ == '__main__':
    # Create a BA model graph
    n=1000
    m=2
    G=nx.generators.barabasi_albert_graph(n,m)
    # find node with largest degree
    node_and_degree=G.degree()

(largest_hub,degree)=sorted(node_and_degree.items(),key=itemgetter(1)
)[-1]
    # Create ego graph of main hub
    hub_ego=nx.ego_graph(G,largest_hub)
    # Draw graph
    pos=nx.spring_layout(hub_ego)

nx.draw(hub_ego,pos,node_color='b',node_size=50,with_labels=False)
    # Draw ego as large and red

nx.draw_networkx_nodes(hub_ego,pos,nodelist=[largest_hub],node_size=3
00,node_color='r')
    plt.savefig('ego_graph.png')
    plt.show()
```