Complex Network

Home work4

Dr.Rahmani

Sara Sadat Younesi – 98533053

Question :

- **Consider the three network datasets of previous assignments**

– If you want, you may also select other datasets.

• Find communities in each network

• Report “modularity” of the communities.

Answer :

 # Commuinity

Functions for computing and measuring community structure.

The community subpackage can be accessed by using networkx.community, then accessing the functions as attributes of community. For example:

Community detection algorithms overview

While humans are very good at detecting distinct or repetitive patterns among a few components, the nature of large interconnected networks makes it practically impossible to perform such basic tasks manually. Groups of densely connected nodes are easy to spot visually, but more sophisticated methods are needed to perform these tasks programmatically. Community detection algorithms are used to find such groups of densely connected components in various networks.

M. Girvan and M. E. J. Newman have proposed one of the most widely adopted community detection algorithms. According to them, groups of nodes in a network are tightly connected within communities and loosely connected between communities.

Practical applications

Because networks are an integral part of many real-world problems, community detection algorithms have found their way into various fields, ranging from social network analysis to public health initiatives.

A known use case is the detection of terrorist groups in social networks by tracking their activities and interactions. In a similar fashion, groups of malicious bots can be detected on online social platforms.

Community detection can be used to study the dynamics of certain groups that are susceptible to epidemic diseases. Other types of diseases can be studied in a similar fashion to discover common links among patients.

One of the most recent use cases, community evolution prediction, involves the prediction of upcoming changes in a network structure.

Community detection techniques

There are two main types of community detection techniques, agglomerative and divisive.

Agglomerative methods generally start with a network that contains only nodes of the original graph. The edges are added one-by-one to the graph, while stronger edges are prioritized over weaker ones. The strength of an edge, or weight, is calculated differently depending on the specific algorithm implementation.

On the other hand, divisive methods rely on the process of removing edges from the original graph iteratively. Stronger edges are removed before weaker ones. At every step, the edge-weight calculation is repeated, since the weight of the remaining edges changes after an edge is removed. After a certain number of steps, we get clusters of densely connected nodes, a.k.a. communities

Community detection algorithms in NetworkX

Name	Description
Girvan-Newman algorithm	The Girvan-Newman algorithm detects communities by progressively removing edges from the original network.
Fluid Communities algorithm	The algorithm is based on the simple idea of fluids interacting in an environment, expanding and pushing each other.
Label Propagation algorithm	Label propagation is a semi-supervised machine learning algorithm that assigns labels to previously unlabeled data points.
Clique Percolation algorithm	The algorithm finds k-clique communities in a graph using the percolation method.
Kernighan-Lin algorithm	This algorithm partitions a network into two sets by iteratively swapping pairs of nodes to reduce the edge cut between the two sets.

girvan_newman

`girvan_newman(G, most_valuable_edge=None)` [source]

Finds communities in a graph using the Girvan–Newman method.

Parameters:

G : *NetworkX graph*

most_valuable_edge : *function*

Function that takes a graph as input and outputs an edge. The edge returned by this function will be recomputed and removed at each iteration of the algorithm.
If not specified, the edge with the highest `networkx.edge_betweenness_centrality()` will be used.

Returns:

iterator

Iterator over tuples of sets of nodes in `G`. Each set of node is a community, each tuple is a sequence of communities at a particular level of the algorithm.

Notes

The Girvan–Newman algorithm detects communities by progressively removing edges from the original graph. The algorithm removes the "most valuable" edge, traditionally the edge with the highest betweenness centrality, at each step. As the graph breaks down into pieces, the tightly knit community structure is exposed and the result can be depicted as a dendrogram.

2. # Modularity

Modularity-based communities

Functions for detecting communities based on modularity.

Modularity-based communities

Functions for detecting communities based on modularity.

`greedy_modularity_communities`(G[, weight, ...])	Find communities in G using greedy modularity maximization.
`naive_greedy_modularity_communities`(G[, ...])	Find communities in G using greedy modularity maximization.

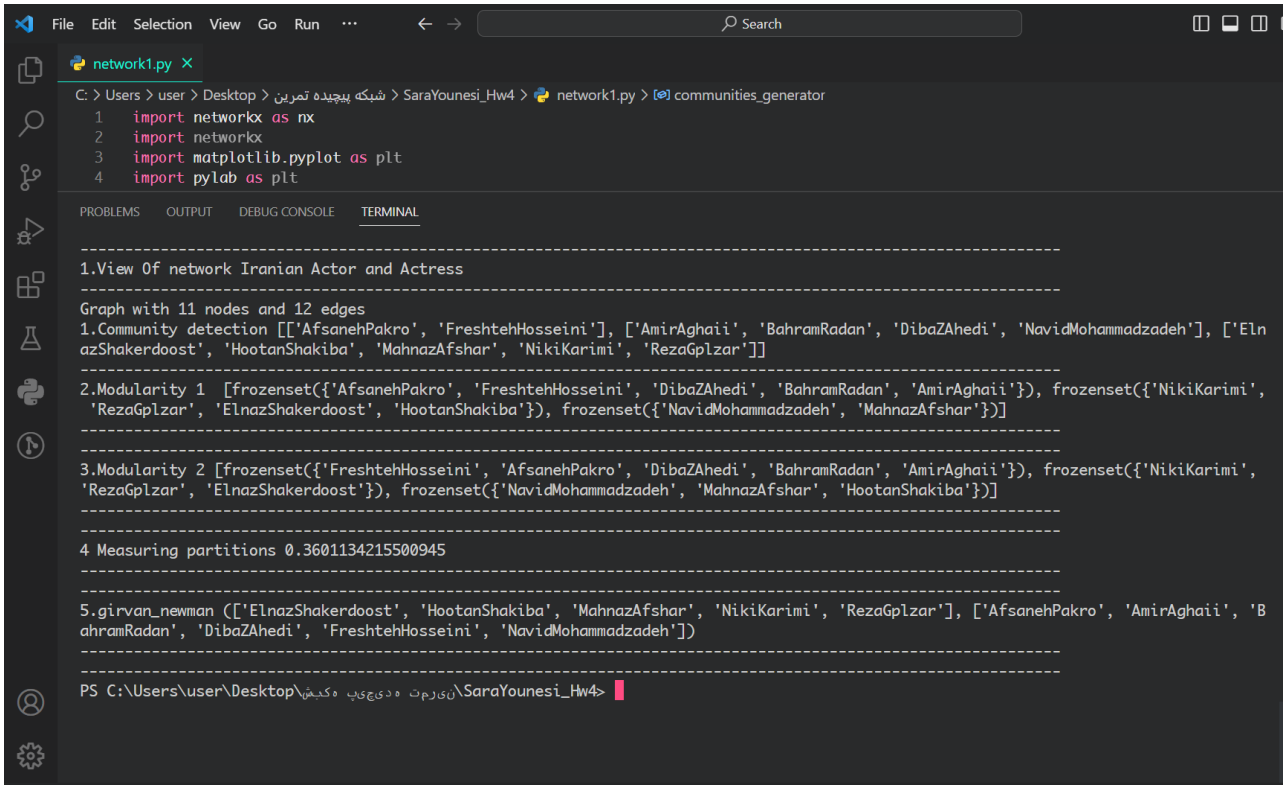
Code and Formula Use

```
communities_generator = nx.community.girvan_newman(G)
top_level_communities = next(communities_generator)
```

```
next_level_communities = next(communities_generator)


communities_generator = community.girvan_newman(G)



print("-------------------------------------------------------------------------
---------------------------------------")
print("1.View Of network Iranian Actor and Actress")
print("-------------------------------------------------------------------------
---------------------------------------")
print(G)
print('1.Community detection', sorted(map(sorted, next_level_communities)))
print("-------------------------------------------------------------------------
---------------------------------------")
print('2.Modularity 1 ', nx.community.greedy_modularity_communities(G))
print("-------------------------------------------------------------------------
---------------------------------------")
print("-------------------------------------------------------------------------
---------------------------------------")
print('3.Modularity 2', nx.community.naive_greedy_modularity_communities(G))
print("-------------------------------------------------------------------------
---------------------------------------")
print("-------------------------------------------------------------------------
---------------------------------------")
print('4 Measuring partitions', nx.community.modularity(G,
nx.community.label_propagation_communities(G)))
print("-------------------------------------------------------------------------
---------------------------------------")
print("-------------------------------------------------------------------------
---------------------------------------")
print('5.girvan_newman', tuple(sorted(c) for c in
next(communities_generator)))
print("-------------------------------------------------------------------------
---------------------------------------")
print("-------------------------------------------------------------------------
---------------------------------------")
```

Results Of 5 Networks

1.View Of network Iranian Actor and Actress



2.View Of network IUST Student

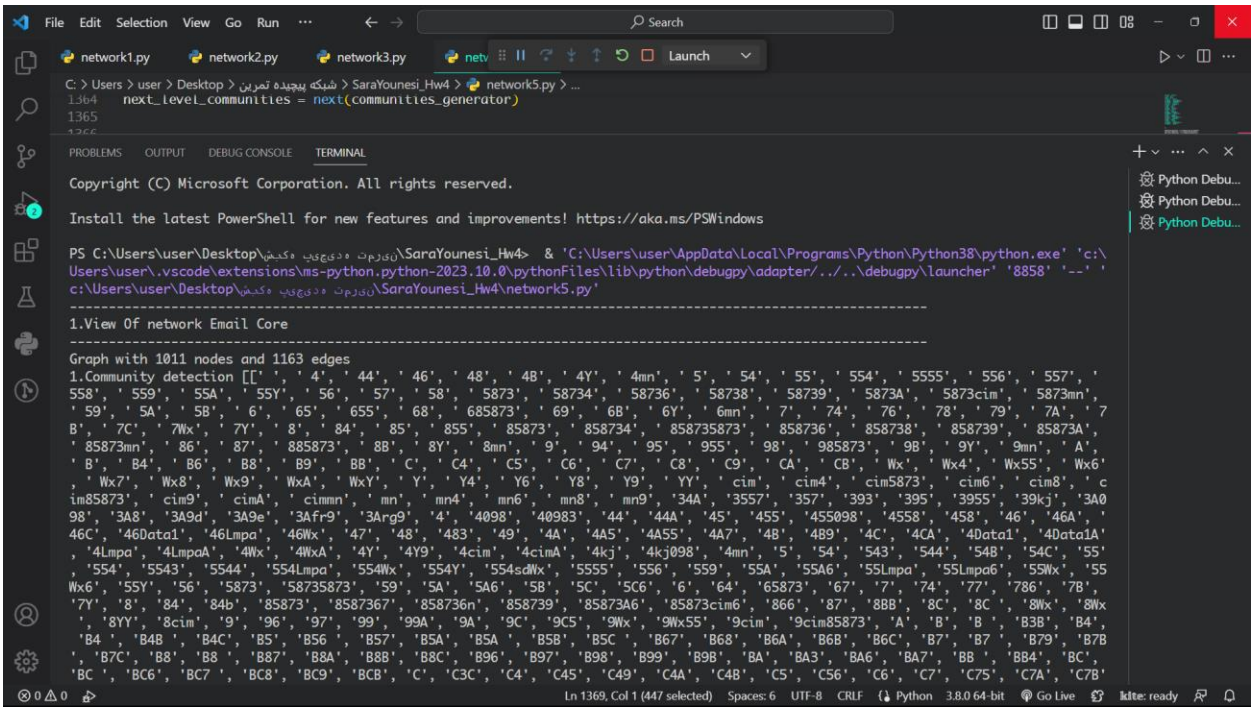
3.View Of network IUST Professors

```
29  print("----------")

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

--------------------------------------------------------------------------------
1.View Of network Professor IUST
--------------------------------------------------------------------------------
Graph with 14 nodes and 16 edges
1.Community detection [['Dr,maleki', 'Dr.Rashidi', 'Dr.Smali'], ['Dr.Ashtiani', 'Dr.Khanjari', 'Dr.Rahmani', 'Dr.etemadi', 'Dr.movahe
di'], ['Dr.Entezari', 'Dr.Jahed', 'Dr.Kangavari', 'Dr.Minaii', 'Dr.Mozaieni', 'Dr.Parsa']]
--------------------------------------------------------------------------------
2.Modularity 1  [frozenset({'Dr.movahedi', 'Dr.Rahmani', 'Dr.Khanjari', 'Dr.etemadi', 'Dr.Ashtiani'}), frozenset({'Dr.Mozaieni', 'Dr.
Rashidi', 'Dr,maleki', 'Dr.Entezari', 'Dr.Smali'}), frozenset({'Dr.Parsa', 'Dr.Jahed', 'Dr.Minaii', 'Dr.Kangavari'})]
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
3.Modularity 2 [frozenset({'Dr.movahedi', 'Dr.Rahmani', 'Dr.Khanjari', 'Dr.etemadi', 'Dr.Ashtiani'}), frozenset({'Dr.Mozaieni', 'Dr.R
ashidi', 'Dr,maleki', 'Dr.Entezari', 'Dr.Smali'}), frozenset({'Dr.Parsa', 'Dr.Jahed', 'Dr.Minaii', 'Dr.Kangavari'})]
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
4 Measuring partitions 0.4112426035502958
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
5.girvan_newman (['Dr,maleki', 'Dr.Ashtiani', 'Dr.Khanjari', 'Dr.Rahmani', 'Dr.Rashidi', 'Dr.Smali', 'Dr.etemadi', 'Dr.movahedi'], ['
Dr.Entezari', 'Dr.Jahed', 'Dr.Kangavari', 'Dr.Minaii', 'Dr.Mozaieni', 'Dr.Parsa'])
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
PS C:\Users\user\Desktop\ﻩﮐﮑﺶ ﻲﻴﭽﻳﺩﻩ ﺕﻣﺭﻱﻧ\SaraYounesi_Hw4>

0   Ln 23, Col 40  Spaces: 6  UTF-8  CRLF  Python  3.8.0 64-bit  Go Live
```

4..View Of network Facebook Relations

```
File  Edit  Selection  View  Go  Run  ···   Search

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

s\lib\python\debugpy\adapter/../..\debugpy\launcher' '8624' '--' 'c:\Users\user\Desktop\ﻩﮐﮑﺶ ﻲﻴﭽﻳﺩﻩ ﺕﻣﺭﻱﻧ\SaraYounesi_Hw4\network4.py
'
--------------------------------------------------------------------------------
1.View Of network FaceBook Relationship
--------------------------------------------------------------------------------
Graph with 1050 nodes and 1175 edges
1.Community detection [[' ', ' 55', ' 5555', ' 556', ' 557', ' 558', ' 559', ' 55Relation1', ' 55Y', ' 55secondrel', ' 5873', ' 58736
', ' 58738', ' 58739', ' 5873Center12im', ' 5873Relation1', ' 5873mn', ' 5873secondrel', ' 5ID1', ' 6', ' 65', ' 655', ' 68', ' 68587
3', ' 69', ' 6ID1', ' 6Y', ' 6mn', ' 7', ' 76', ' 78', ' 79', ' 7Center12', ' 7Relation1', ' 7Y', ' 7firstqx', ' 7secondrel', ' 8', '
 85', ' 855', ' 85873', ' 858735873', ' 858736', ' 858738', ' 858739', ' 85873Relation1', ' 85873mn', ' 85873secondrel', ' 86', ' 87'
, ' 885873', ' 8ID1', ' 8Y', ' 8mn', ' 8secondrel', ' 9', ' 95', ' 955', ' 98', ' 985873', ' 9ID1', ' 9Y', ' 9mn', ' 9secondrel', ' C
enter128', ' Center12ID1', ' Center12Relation1', ' Center12im', ' Center12im5873', ' Center12im6', ' Center12im8', ' Center12im85873'
, ' Center12im9', ' Center12imRelation1', ' Center12immn', ' Center12imsecondrel', ' ID1', ' ID16', ' ID18', ' ID19', ' ID1ID1', ' Re
lation1', ' Y', ' Y6', ' Y8', ' Y9', ' YY', ' Ysecondrel', ' firstqx', ' firstqx55', ' firstqx6', ' firstqx7', ' firstqx8', ' firstqx
9', ' firstqxRelation1', ' firstqxY', ' firstqxsecondrel', ' mn', ' mn6', ' mn8', ' mn9', ' mnsecondrel', ' secondrel', ' secondrel6'
, ' secondrel8', ' secondrelY', ' secondrelmn', ' secondrelsecondrel', '3557', '357', '393', '395', '3955', '39kj', '3Relation1098',
'3Relation18', '3Relation19d', '3Relation19e', '3Relation1fr9', '3Relation1rg9', '3secondrelRelation1', '55', '5555', '556', '559', '
55LmpRelation1', '55LmpRelation16', '55Relation1', '55Relation16', '55Y', '55firstqx', '55firstqx6', '55secondrel', '55secondrel3', '
55secondrelLmpRelation1', '55secondrelY', '55secondrelfirstqx', '55secondrelsdfirstqx', '55secondrelsecondrel', '56', '5873', '587358
73', '59', '5Center12', '5Center126', '5ID1', '5Relation1', '5Relation16', '5firstq', '5secondrel', '5secondrel3', '5secondrelCenter1
2', '5secondrelID1', '5secondrelsecondrel', '6', '65873', '67', '6d', '6k', '6l', '6sa', '6secondrel', '6z', '7', '77', '786', '7ID1'
, '7Y', '7ere', '7fgjk', '7secondrel', '7z', '7zdfz', '8', '85873', '8587367', '858736n', '858739', '85873Center12im6', '85873Relatio
n16', '87', '8Center12', '8Center12 ', '8Center12im', '8ID1ID1', '8YY', '8firstqx', '8firstqx ', '8secondrel', '8secondrelID1', '9',
'96', '97', '99', '99Relation1', '9Center12', '9Center125', '9Center12im', '9Center12im85873', '9Relation1', '9firstqx', '9firstqx55'
, 'Center12', 'Center123Center12', 'Center125', 'Center1256', 'Center126', 'Center127', 'Center1275', 'Center127ID1', 'Center127Relat
ion1', 'Center128', 'Center1285', 'Center1289', 'Center128ID1', 'Center128Relation1', 'Center128secondrel', 'Center129', 'Center1297'
, 'Center129Center12', 'Center129ID1', 'Center129Relation1', 'Center129secondrel', 'Center12ID1', 'Center12Relation1', 'Center12Relat
ion15', 'Center12Relation16', 'Center12Relation17', 'Center12Relation19', 'Center12Relation1Center12', 'Center12Relation1secondrel',
'Center12qq', 'Center12fv', 'Center12fvd', 'Center12im', 'Center12im6', 'Center12im8', 'Center12im85873', 'Center12im885873', 'Center
12im9', 'Center12im95873', 'Center12imRelation1', 'Center12imRelation16', 'Center12immn', 'Center12imsecondrel', 'Center12imsecondrel
85873', 'Center12imsecondrelmn', 'Center12kl', 'Center12pi', 'Center12secondrel', 'Center12secondrel5', 'Center12secondrel9', 'Center
12secondrelID1', 'Center12secondrelRelation1', 'Center12xCenter12', 'DRelation1tRelation11098Y', 'DRelation1tRelation117', 'DRelation
1tRelation117kj', 'DRelation1tRelation119Y', 'DRelation1tRelation11Relation17', 'DRelation1tRelation11Y', 'DRelation1tRelation11kj6',
 'DRelation1tRelation11secondrel9', 'ID1', 'ID1 ', 'ID13Center12', 'ID13ID1', 'ID15', 'ID156 ', 'ID157', 'ID15Center12 ', 'ID15ID1',

⊗0 ⚠0   Ln 1359, Col 1 (4 selected)  Spaces: 6  UTF-8  CRLF  Python  3.8.0 64-bit  Go Live  Idle: ready
```

5.View Of network Email Core



Source

— NetworkX 3.1 documentation