Assignment 2: Network measures

Date of issue: 07.03.2019

Rana Hussein: `rana.hussein@unifr.ch`

Task 1. Given two disconnected social networks in Figure 1. Consider joining them together to obtain a connected network component. Which user should *Phil* be friends with, in order for him to achieve the highest Betweenness Centrality score in the constructed connected network? Explain your answer.

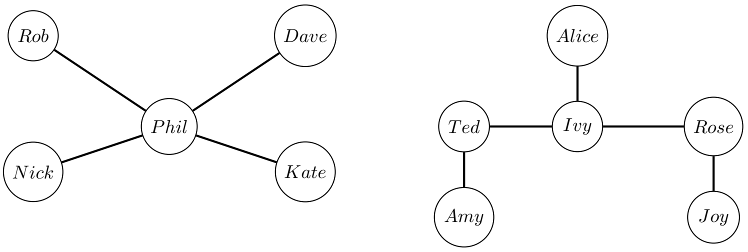

Figure 1 - Disconnected social networks

Task 2. Assume the following social network in Figure 2, in which each node represents a user and links represent friendships between them.

1. Calculate the Jaccard similarity between user *U5* and all other users. Which is the most structurally equivalent user to *U5*?
2. Modify the graph by performing one edge operation, such that *U3* becomes the most structurally equivalent user to *U5*. An edge operation is either an edge addition or a removal.

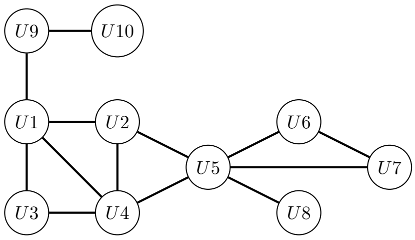

Figure 2 - Social network

Task 3. Implement a Python script using NetworkX library, and perform the following tasks:

1. Download and load the social network *email-Eu-core.txt.gz* from `http://snap.stanford.edu/data/email-Eu-core.html` into your script.The dataset represents a network of email data between co-workers. Nodes represent members of an institution and edges represent email communications between them.

2. Use the Page Rank centrality function in *NetworkX*, to print the node with the highest centrality score.
3. Use the Eigenvector centrality function in *NetworkX*, to print the centrality score for each node.

Task 4. Consider a network for instant messaging communication shown in Figure 3, nodes represent users and links represent communication between them. Implement a script to perform the following tasks:

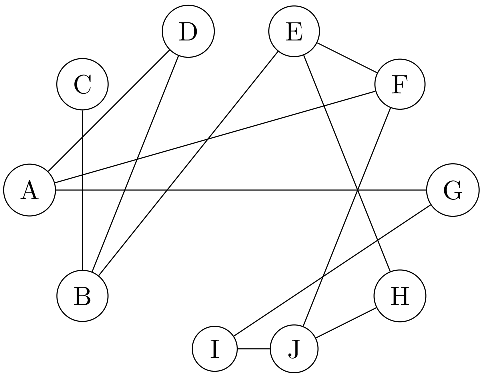

Figure 3 - Messaging communication network

1. Use the Adamic Adar similarity function in *NetworkX* python library, to print the similarity scores for each pair of nodes.
2. Implement the Closeness centrality measure by doing the following:
   - Implement a function to calculate the minimal path length between two nodes.
   - Implement Closeness centrality using the minimal paths.

Assignment delivery notes:

- This assignment should be uploaded to ILIAS by 13.03.2019