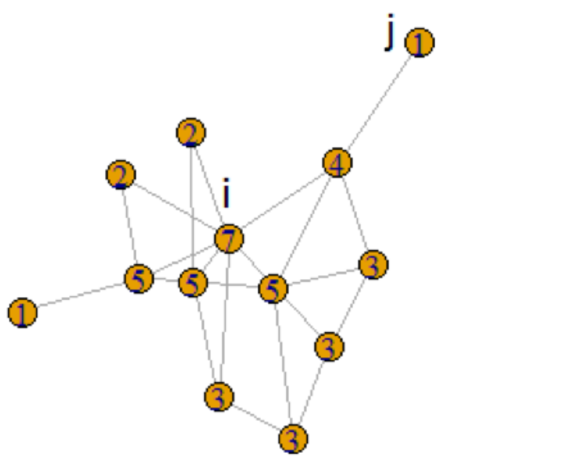
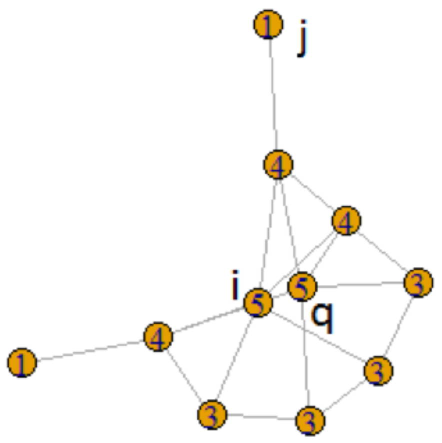
Homework 3: Graphs

CS 201 Data Structures II
Habib University
Spring 2020

Due: 1830h on Friday, 5 March



(a) i is the most popular vertex in the network with a degree of 7.



(b) This network has 2 most popular vertices, i and q, each having a degree of 5.

Figure 1: Two networks are shown with the nodes in each labeled with its degree.

The world we live in is an inter-connected system and many aspects of it can be represented by a graph, or a *network*. A good example is the World Wide Web (WWW) which is an inter-connected system of web-pages in which each vertex represents a page and an edge represents a direct link between two web pages.

Network Science is the study of the networks that appear in different domains, e.g. in social, technological, and biological disciplines. It finds application in many fields including sociology, physics, mathematics, biology, economics, and computer science. The aim is to identify underlying common principles in the formation of networks and explore why they vary in some cases. This is done through exploration of networks to study structural patterns and to identify variations.

The field has undergone tremendous growth especially after ground-breaking discoveries of *small-world* and *scale-free* properties in network models. Traditional graph theoretic properties take on added significance in network science. For example, the degree of a vertex can be interpreted as the *popularity* of the vertex–in a social network, a person with a higher value of the degree is a person with a lot of friends. Similarly, *clustering coefficient* is used to quantify how closely vertices are connected to each other within a neighborhood. The *degree distribution* of a graph describes the structure of the graph, i.e. whether it has a few prominent vertices or *hubs*.

In this assignment, we are going to explore (and implement) some of the measures used in network science and apply them to a few popular datasets.

1 Measures

We look at 5 measures: *degree centrality*, *clustering coefficient*, *average neighbor degree*, *Jaccard similarity*, and *popular distance.*

Degree Centrality *Centrality* provides a measure of the *importance* of a vertex or edge. *Degree centrality* uses the degree of a vertex, i.e. the number of connections of the vertex, as the measure. To compare across networks of different sizes, i.e. with different number of vertices, the degree is normalized by the total number of vertices in the network.

$$C_D(i) = \frac{k_i}{n-1}$$

where i is the subject vertex, k_i is its degree, and n is the total number of vertices in the network.

Clustering Coefficient *Clustering coefficient* is used to quantify the connectivity of vertices in a graph. The *local* clustering coefficient for an individual vertex, i, can be used to determine whether a graph is a *small-world network* and is calculated as

$$C_i = \frac{L_i}{\frac{k_i(k_i-1)}{2}}$$

where k_i is the number of neighbors of i and L_i is the total number of edges between the k_i neighbors of i. The denominator indicates the maximum possible edges among the k_i neighbors. If all the neighbors are connected with each other, C_i will be 1, and if there are no common friends among the neighbors, C_i will be 0. The average clustering coefficient of a network with n vertices is the average of all the local clustering coefficients.

$$\overline{C} = \frac{1}{n}\sum_i C_i$$

Average Neighbor Degree The degree of a vertex describes how famous or popular it is in the network. *Average neighbor degree* tells whether the vertex is surrounded by famous vertices. For a vertex, i, it is calculated as

$$K_i = \frac{1}{|N(i)|}\sum_{j\in N(i)} k_j$$

where $N(i)$ is the set of neighbours of i and k_j is the degree of node j.

Jaccard Similarity *Jaccard Similarity* is used to determine the similarity of two vertices to each other. Vertices i and j are more similar if they have more common friends. Jaccard similarity between 2 vertices, i and j, is calculated as:

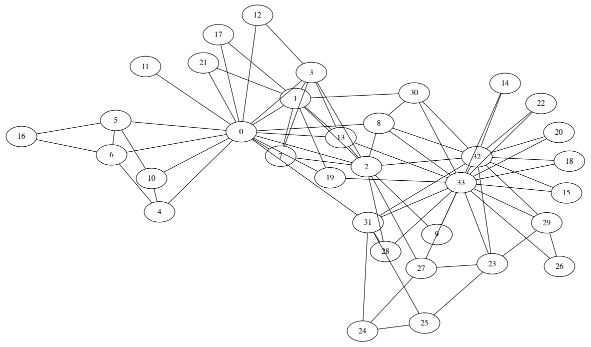
$$J(i,j) = \frac{|N(i)\cap N(j)|}{|N(i)\cup N(j)|} = \frac{|N(i)\cap N(j)|}{|N(i)|+|N(j)|-|N(i)\cap N(j)|}$$

where $N(i)$ and $N(j)$ are the sets of neighbors of i and j respectively.

Popular Distance Another measure to find the importance of a vertex in a network is its distance from the most popular vertex of the network. The *popular distance* of a vertex i is the length of the shortest path from i to the vertex with the maximum degree. If the network contains more than one vertex with maximum degree then the minimum shortest path is considered. For example, the popular distance of vertex j in Figure 1a is 2 as that is the length of the shortest path from j to i. In Figure 1b, the popular distance of j is again 2 as it is at a distance of 2 from each of i and q which are the 2 most popular vertices in the network.

2 Datasets

For this homework, we use three different datasets: “Zachary’s Karate Club”, “Coauthorships in Network Science”, and “High Energy Physics - Theory collaboration network”. For the figures below, zoom into the figure with your PDF viewer to see more detail.



Zachary’s Karate Club is probably the most presented network at network science conferences! The network represents social ties among the members of a karate club at a university. Each vertex represents a member of the club and an edge between two vertices indicates that the corresponding members are friends. The network is undirected and unweighted.

Coauthorships in Network Science is a collaboration network of coauthorships between scientists in the area of network theory and experiments. Each vertex represents an author and an edge between vertex i and vertex j indicates that the corresponding authors have co-authored a paper. The network is undirected and unweighted. See Figure 2 for a visualization.

High Energy Physics - Theory collaboration network is a collaboration network of coauthorships between scientists in the area of High Energy Physics – Physics. The network is constructed from the e-print arXiv for the papers submitted to High Energy Physics - Theory category. The network is undirected and weighted where the weight of an edge represents the number of papers co-authored by both authors involved in the edge. Edge weights are normalized. See Figure 3 for a visualization.

3 Tasks

Your task is to go over the provided files and fill in the required functionality.

`networks.py` contains network operations that operate on a `Graph`. All methods currently contain `pass` which will have to be replaced by you. The exception is `visualize` which is implemented. When this function executes successfully, the visualization is automatically stored in the files `Graph`.gv and `Graph.gv.pdf`. Do not add these to your repository.

`graphs.py` contains a complete implementation of `Edge` which represents an undirected edge and the interface of `Graph`. You have to implement the `Graph` interface using 3 different classes `SetGraph`, `AdjacencyMatrix`, and `AdjacencyList` that store a graph respectively as 2 sets, an adjacency matrix, and an adjacency list. Do not alter the code already provided.

Also go over the files in the `datasets` folder. Each file represents a network as en `edge list`. Each line contains the 2 endpoints of an edge. Each vertex is represented by an integer. For some networks, each line also contains a weight as a floating point number. Note that the vertex numbers need not begin at 0 and need not be contiguous. A Graph is instantiated with a `str` instance representing the content of such a file.

You may use the given networks to test your code.

Credits

This homework and related files are courtesy of Muhammad Qasim Pasta.

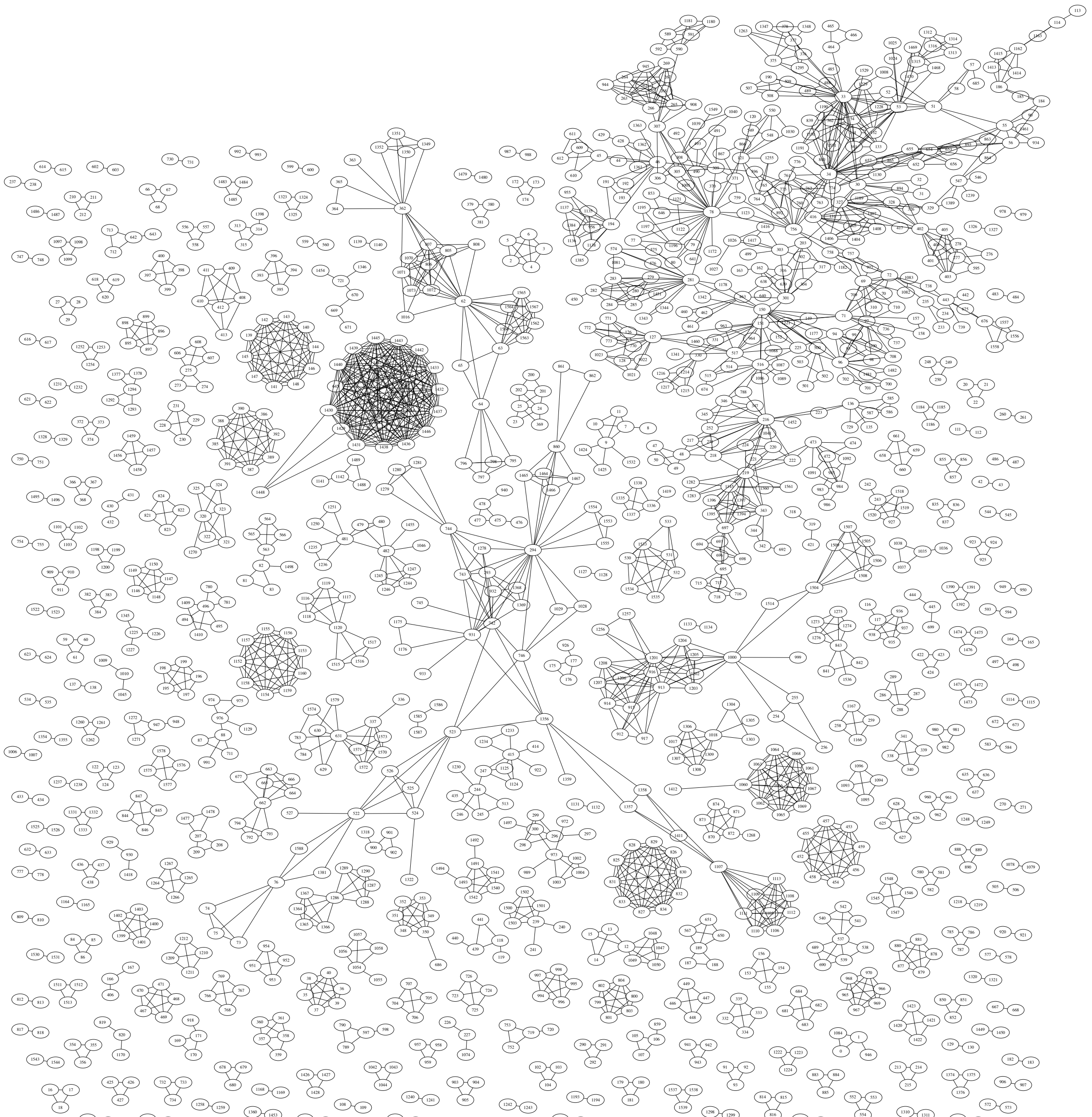


Figure 2: Coauthorships in Network Science

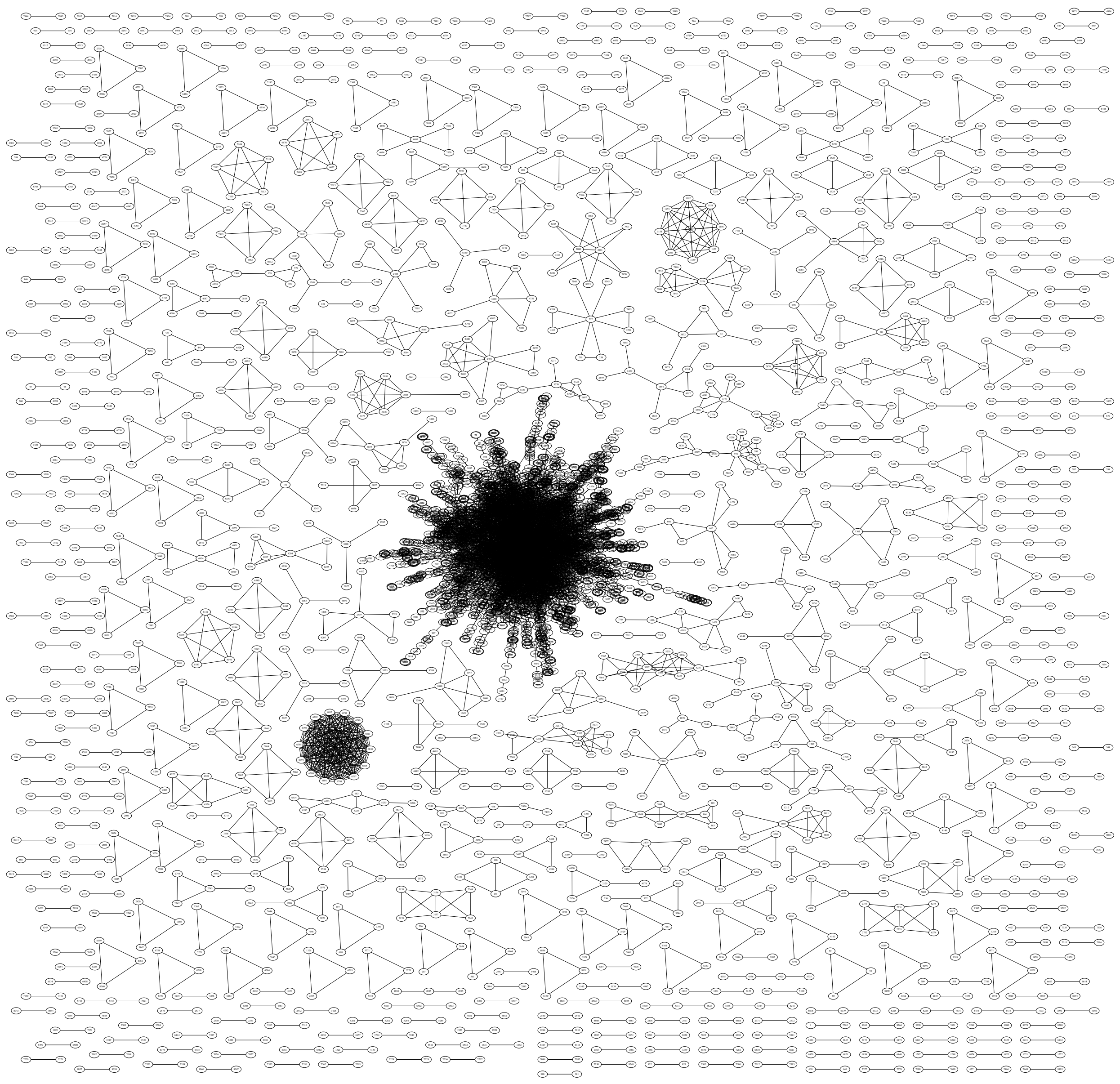


Figure 3: High Energy Physics - Theory collaboration network