Homework 1 (10 points, Due Feb 15)

Solutions

Question 1 (2.0 points: 0.25/each): Please use your own language to briefly explain the following concepts:

Social networks:
A social network is a network representing interactions between members. It can be represented as a graph network with each node denoting an individual and edges between two nodes denoting their relationships, such as friendships, kinships, and citation relationship.

Undirected graph:
An undirected graph is a graph where edges have no orientation (or direction). So an edge (a,b) is identical to an edge (b,a).

Adjacency matrix:
A square matrix to represent a graph, with each row/column denoting a node and the non-diagonal entry a_{ij} denoting the relationship between node a_i and a_j. The size of the matrix is n by n, where n is the total number of nodes in the network.

Network Diameter:
The diameter of a network is the largest shortest-path distance between node pairs inside the network.

Centrality score of a node:
Centrality score of a node measures the importance of the node in the network. This can be evaluated from different aspects. For example, degree centrality score uses the node degree as the measure to calculate the importance of each node, whereas closeness centrality score use the distance between a node and all other nodes in the network to calculate the importance of each node.

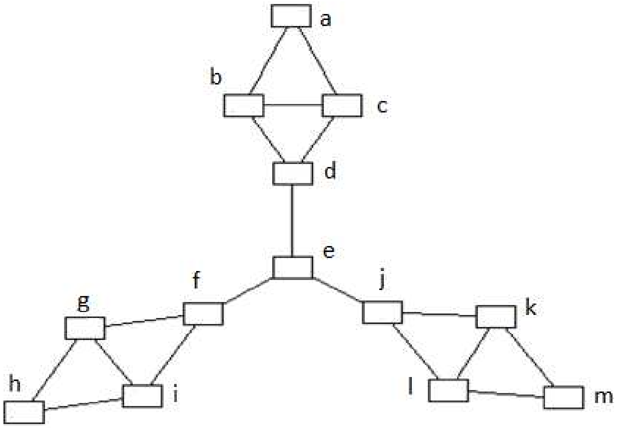
Random Walk:
Random walk denotes a process which random selects a succeeding step to continue the walk. In a graph (or a network), a random walk starts from a node, and continuously moves to the next node by randomly select an out-link, and repeats this process as walk continues.

Random Graph:
A random graph is a graph model which assumes that each pair of node has an equal probability of establishing a linkage.

Power-Law Distribution:

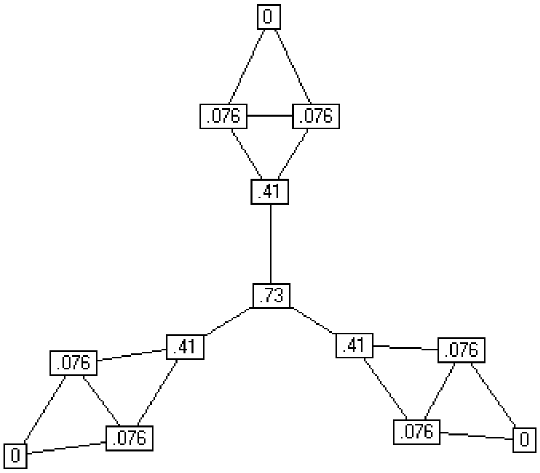
Power-law distribution (a.k.a scale-free distribution) is a special mathematical relationship observed in social networks any other man-made complex networks. This distribution model shows that one variable (such as the frequency of nodes) follows a power distribution with respect to another variable (such as the degree of the node).

Question 2 (1.5 points): In the following network, please calculate the Betweenness Centrality scores [0.5 pt], Closeness Centrality score [0.5 pt], and Eigen Vector based centrality scores [0.5 pt] for every nodes in the network (please show your solutions)

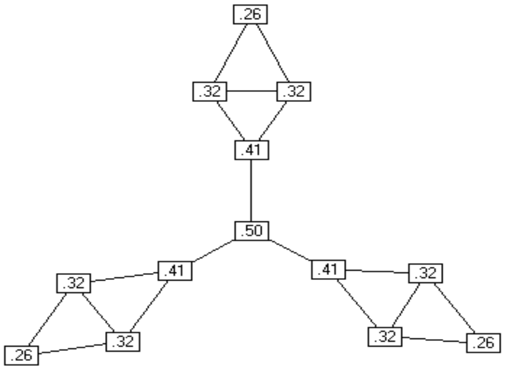


Solutions:

Between ness



Closeness:

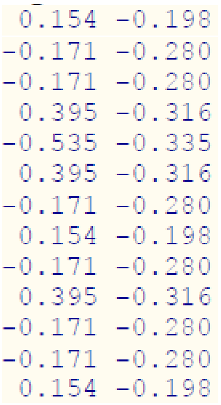


Adjacency matrix

	a	b	c	d	e	f	g	h	i	j	k	l	m
a	0	1	1	0	0	0	0	0	0	0	0	0	0
b	1	0	1	1	0	0	0	0	0	0	0	0	0
c	1	1	0	1	0	0	0	0	0	0	0	0	0
d	0	1	1	0	1	0	0	0	0	0	0	0	0
e	0	0	0	1	0	1	0	0	0	1	0	0	0
f	0	0	0	0	1	0	1	0	1	0	0	0	0
g	0	0	0	0	0	1	0	1	1	0	0	0	0
h	0	0	0	0	0	0	1	0	1	0	0	0	0
i	0	0	0	0	0	1	1	1	0	0	0	0	0
j	0	0	0	0	1	0	0	0	0	0	1	1	0
k	0	0	0	0	0	0	0	0	0	1	0	1	1
l	0	0	0	0	0	0	0	0	0	1	1	0	1
m	0	0	0	0	0	0	0	0	0	0	1	1	0

The largest eigenvalue is 2.833 (which corresponds to the 2nd eigen vector).

Eigen Vectors: (first, second)

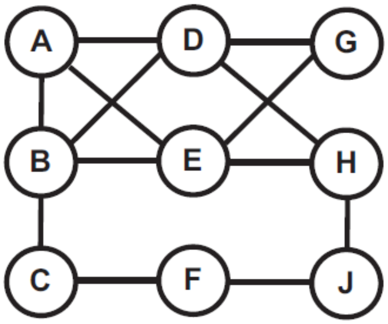


```
 0.154 -0.198
-0.171 -0.280
-0.171 -0.280
 0.395 -0.316
-0.535 -0.335
 0.395 -0.316
-0.171 -0.280
 0.154 -0.198
-0.171 -0.280
 0.395 -0.316
-0.171 -0.280
-0.171 -0.280
 0.154 -0.198
```

The corresponding centrality score:

	a	b	c	d	e	f	g	h	i	j	k	l	m
$C_E(i)$	0.198	0.280	0.280	0.316	0.335	0.316	0.280	0.198	0.280	0.316	0.280	0.280	0.198

Question 3 (2.5 pts): In the following network, please explain how to use adjacency matrix and the power of adjacency matrix to find diameter of the network (show your solution 1 pt). Please draw degree distribution of the network [0.25 pt], calculate clustering coefficient for very nodes in the network [0.25 pt]. Please also calculate the edge density [0.25 pt] and the clustering coefficient of the whole network [0.25]. Please explain why clustering coefficient is smaller than the edge density [0.25 pt]. Please find the node with the highest betweenness score (please show your solution [0.25 pt])



Solutions:

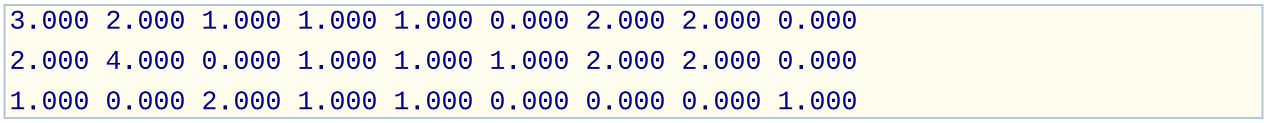
Finding diameter of the network.

Construct Adjacency matrix:

M=

	A	B	C	D	E	F	G	H	J
A	0	1	0	1	1	0	0	0	0
B	1	0	1	1	1	0	0	0	0
C	0	1	0	0	0	1	0	0	0
D	1	1	0	0	0	0	1	1	0
E	1	1	0	0	0	0	1	1	0
F	0	0	1	0	0	0	0	0	1
G	0	0	0	1	1	0	0	0	0
H	0	0	0	1	1	0	0	0	1
J	0	0	0	0	0	1	0	1	0

M^2=



```
3.000 2.000 1.000 1.000 1.000 0.000 2.000 2.000 0.000
2.000 4.000 0.000 1.000 1.000 1.000 2.000 2.000 0.000
1.000 0.000 2.000 1.000 1.000 0.000 0.000 0.000 1.000
```

```
1.000 1.000 1.000 4.000 4.000 0.000 0.000 0.000 1.000
1.000 1.000 1.000 4.000 4.000 0.000 0.000 0.000 1.000
0.000 1.000 0.000 0.000 0.000 2.000 0.000 1.000 0.000
2.000 2.000 0.000 0.000 0.000 0.000 2.000 2.000 0.000
2.000 2.000 0.000 0.000 0.000 1.000 2.000 3.000 0.000
```
0.000 0.000 1.000 1.000 1.000 0.000 0.000 0.000 2.000

M+M^2=

```
3 2 1 1 1 0 2 2 0
2 4 0 1 1 1 2 2 0
1 0 2 1 1 0 0 0 1
1 1 1 4 4 0 0 0 1
1 1 1 4 4 0 0 0 1
0 1 0 0 0 2 0 1 0
2 2 0 0 0 0 2 2 0
2 2 0 0 0 1 2 3 0
```
0 0 1 1 1 0 0 0 2

There is still zero elements on the off-diagonal elements. So we continue to obtain M^3

M^3

=

```
4.000  6.000  2.000  9.000  9.000  1.000  2.000  2.000  2.000
 6.000  4.000  5.000 10.000 10.000  0.000  2.000  2.000  3.000
 2.000  5.000  0.000  1.000  1.000  3.000  2.000  3.000  0.000
 9.000 10.000  1.000  2.000  2.000  2.000  8.000  9.000  0.000
 9.000 10.000  1.000  2.000  2.000  2.000  8.000  9.000  0.000
 1.000  0.000  3.000  2.000  2.000  0.000  0.000  0.000  3.000
 2.000  2.000  2.000  8.000  8.000  0.000  0.000  0.000  2.000
 2.000  2.000  3.000  9.000  9.000  0.000  0.000  0.000  4.000
```
 2.000 3.000 0.000 0.000 0.000 3.000 2.000 4.000 0.000

M+M^2+M^3=

```
7  8  3 10 10  1  4  4  2
 8  8  5 11 11  1  4  4  3
 3  5  2  2  2  3  2  3  1
10 11  2  6  6  2  8  9  1
10 11  2  6  6  2  8  9  1
```

```
 1  1  3  2  2  2  0  1  3
 4  4  2  8  8  0  2  2  2
 4  4  3  9  9  1  2  3  4
 2  3  1  1  1  3  2  4  2
```

There is still two off-diagonal elements have zero values, so we calculate M^4

M^4=

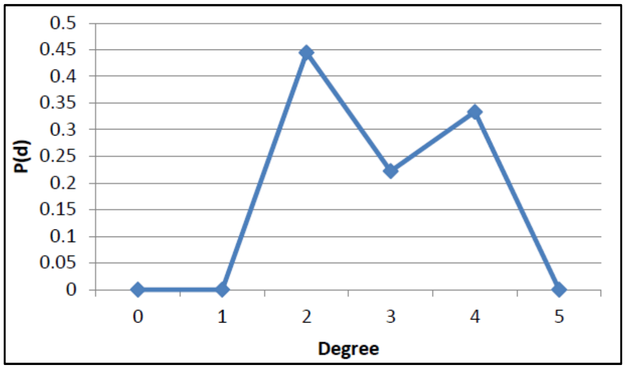
```
24.000 24.000  7.000 14.000 14.000  4.000 18.000 20.000  3.000
24.000 31.000  4.000 14.000 14.000  8.000 20.000 23.000  2.000
 7.000  4.000  8.000 12.000 12.000  0.000  2.000  2.000  6.000
14.000 14.000 12.000 36.000 36.000  1.000  4.000  4.000 11.000
14.000 14.000 12.000 36.000 36.000  1.000  4.000  4.000 11.000
 4.000  8.000  0.000  1.000  1.000  6.000  4.000  7.000  0.000
18.000 20.000  2.000  4.000  4.000  4.000 16.000 18.000  0.000
20.000 23.000  2.000  4.000  4.000  7.000 18.000 22.000  0.000
 3.000  2.000  6.000 11.000 11.000  0.000  0.000  0.000  7.000
```

M+M^2+M^3+M^4=

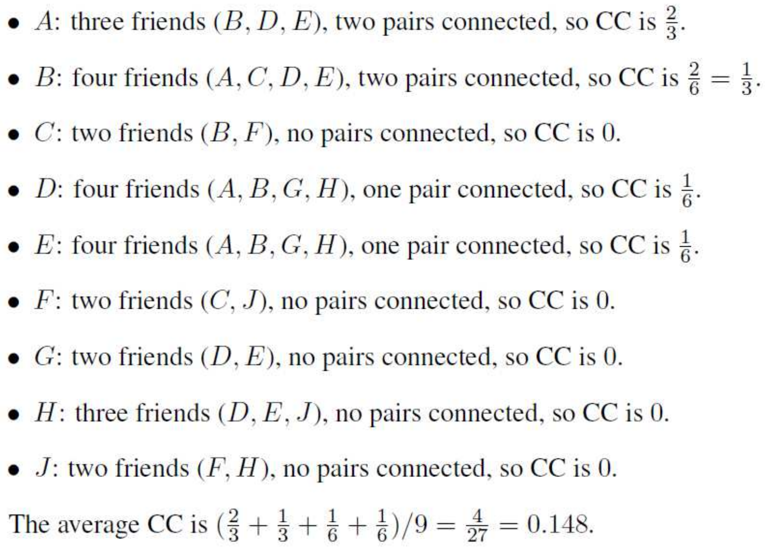
```
31 32 10 24 24  5 22 24  5
32 39  9 25 25  9 24 27  5
10  9 10 14 14  3  4  5  7
24 25 14 42 42  3 12 13 12
24 25 14 42 42  3 12 13 12
 5  9  3  3  3  8  4  8  3
22 24  4 12 12  4 18 20  2
24 27  5 13 13  8 20 25  4
 5  5  7 12 12  3  2  4  9
```

Now, all off-diagonal elements are non-zero. It means that the maximal shortest distance path between node pairs is 4. Therefore, the diameter of the network is 4.

Degree Distribution



Clustering Coefficient for each node:



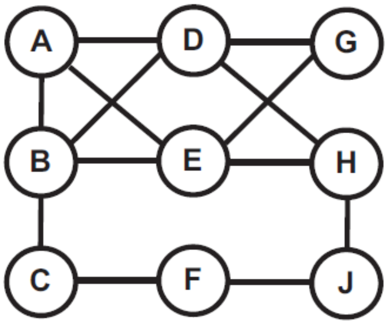
- A: three friends (B, D, E), two pairs connected, so CC is $\frac{2}{3}$.
- B: four friends (A, C, D, E), two pairs connected, so CC is $\frac{2}{6} = \frac{1}{3}$.
- C: two friends (B, F), no pairs connected, so CC is 0.
- D: four friends (A, B, G, H), one pair connected, so CC is $\frac{1}{6}$.
- E: four friends (A, B, G, H), one pair connected, so CC is $\frac{1}{6}$.
- F: two friends (C, J), no pairs connected, so CC is 0.
- G: two friends (D, E), no pairs connected, so CC is 0.
- H: three friends (D, E, J), no pairs connected, so CC is 0.
- J: two friends (F, H), no pairs connected, so CC is 0.

The average CC is $(\frac{2}{3} + \frac{1}{3} + \frac{1}{6} + \frac{1}{6})/9 = \frac{4}{27} = 0.148$.

Clustering Coefficient of the whole network: 0.148

Edge Density=13/(9*(9-1)/2)=13/36=0.361

Node B has the highest betweenness score: 7.417



Node B betweeneness score for node pairs:

(A,C): 1/1; (C,D): 1/1; (A,F):1/1

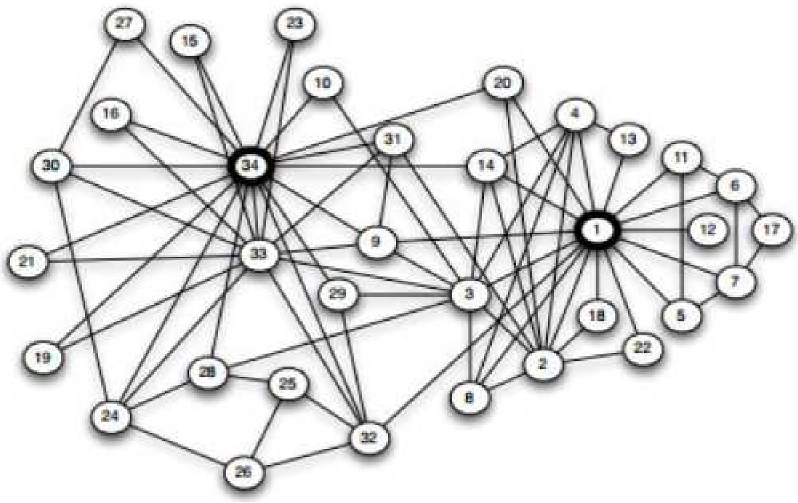
(C,E): 1/1; (E,F):1/2; (D,F): 1/2

(C,H): 2/3; (C,G): 2/2; (D,E): 1/4

(F,G): 2/4=0.5;

Question 4 (2.0 pts) The following network shows a small benchmark "Zachary's karate club" social network which contains "friendships between 34 members of a karate club at a US university in the 1970s".

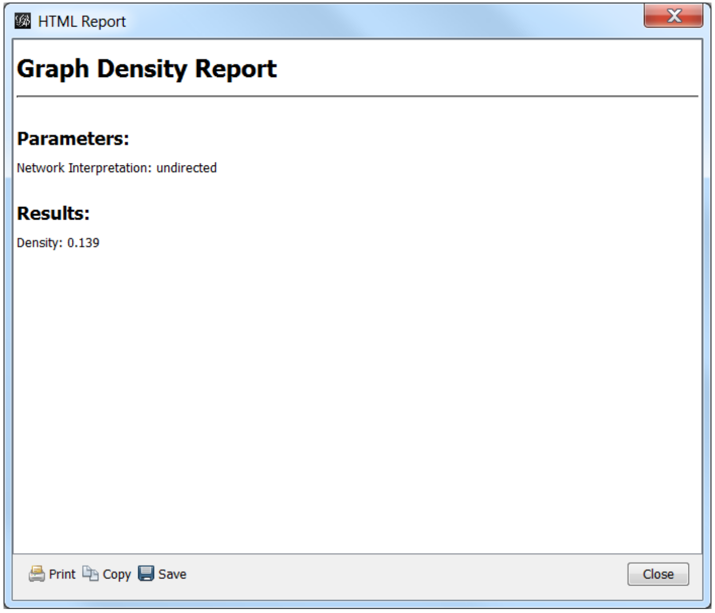
1. Please calculate the edge density and the clustering coefficient of the whole network, and analyze why clustering coefficient is larger (or smaller) than the edge density [0.5 pt].
2. Please calculate the average distance between any two pairs of nodes [0.5 pt], and report the Diameter of the network [0.5 pt].
3. Please find the node(s) with the highest closeness centrality score [0.5 pt]



Zachary’s karate club network

Solutions

1. Edge density: 0.139 (From Gephi)



Graph Density Report

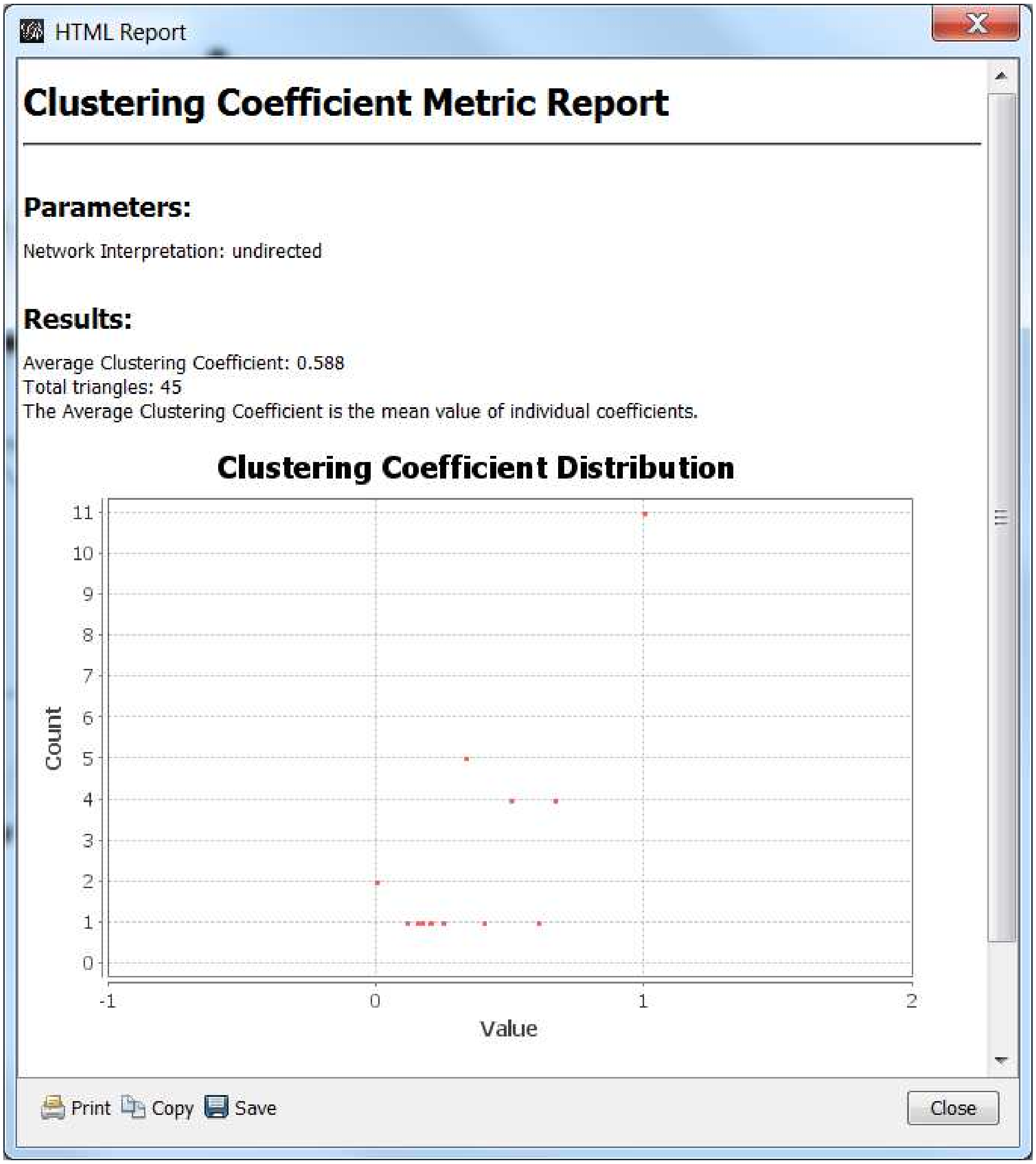
Parameters:

Network Interpretation: undirected

Results:

Density: 0.139

Clustering coefficient: 0.588



Clustering Coefficient Metric Report

Parameters:

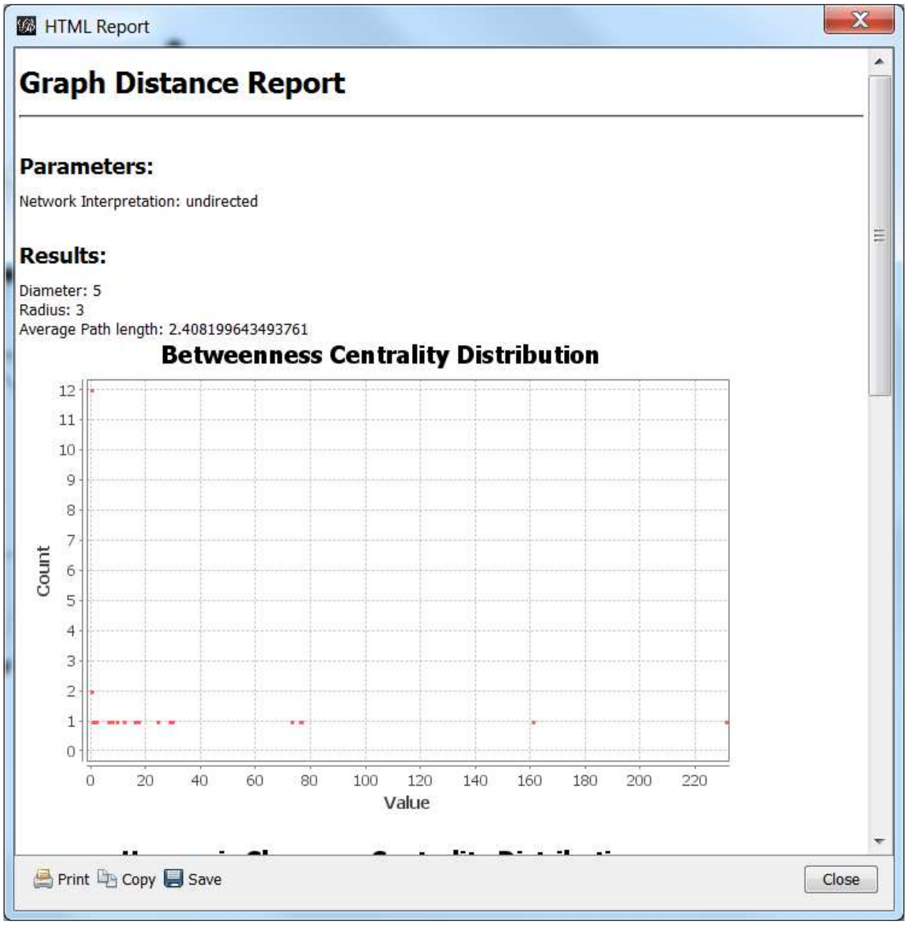
Network Interpretation: undirected

Results:

Average Clustering Coefficient: 0.588
Total triangles: 45
The Average Clustering Coefficient is the mean value of individual coefficients.

Clustering coefficient is greater than edge density, because for each node, its neighbors have a relatively higher probability of being connected, compared to the probability of any two nodes being connected. This is a commonly observed phenomenon in social networks.

2. Average path length: 2.408
Diameter: 5



Graph Distance Report

Parameters:

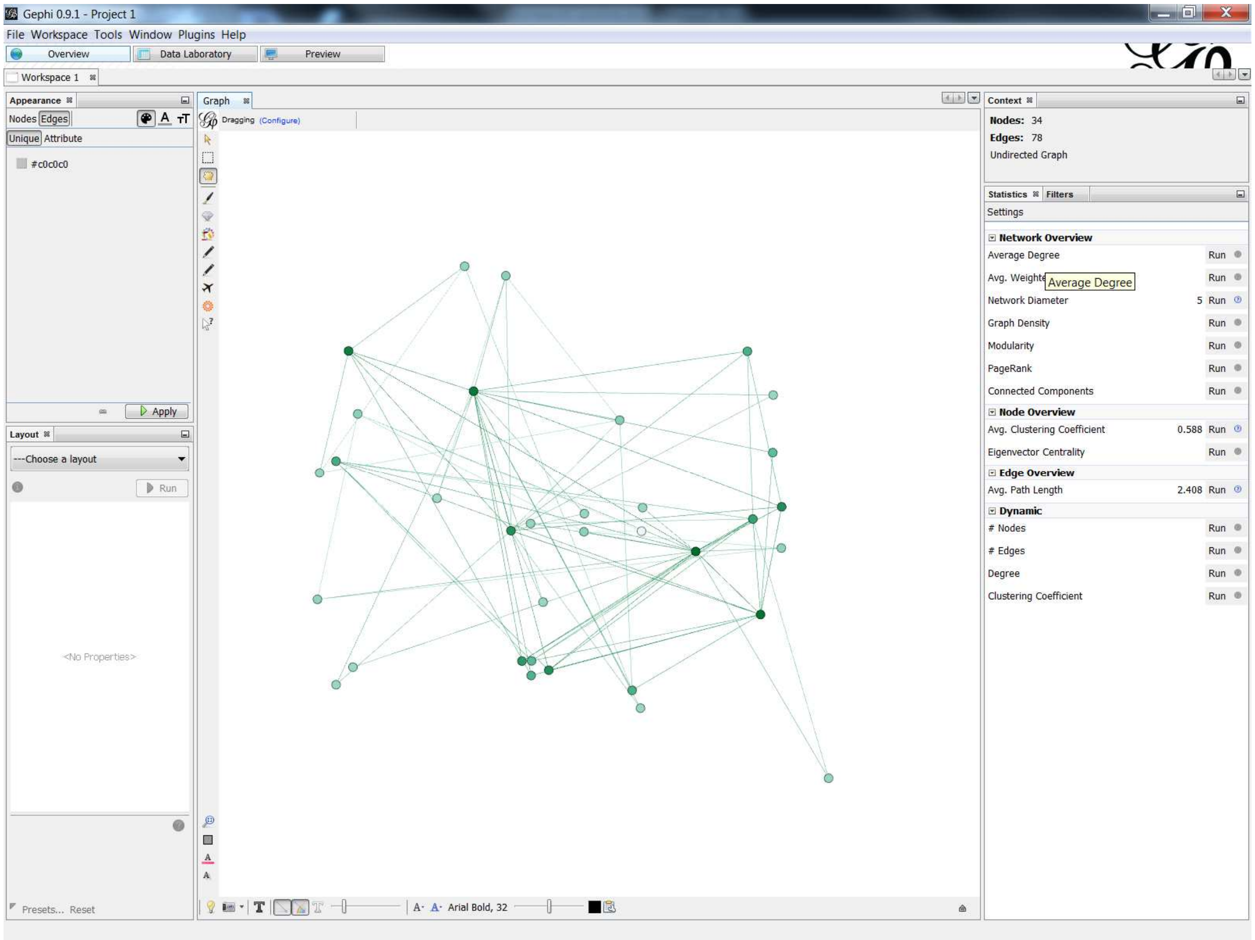
Network Interpretation: undirected

Results:

Diameter: 5
Radius: 3
Average Path length: 2.408199643493761

Betweenness Centrality Distribution

3. Node centrality score (color coded)



Nodes: 34
Edges: 78
Undirected Graph

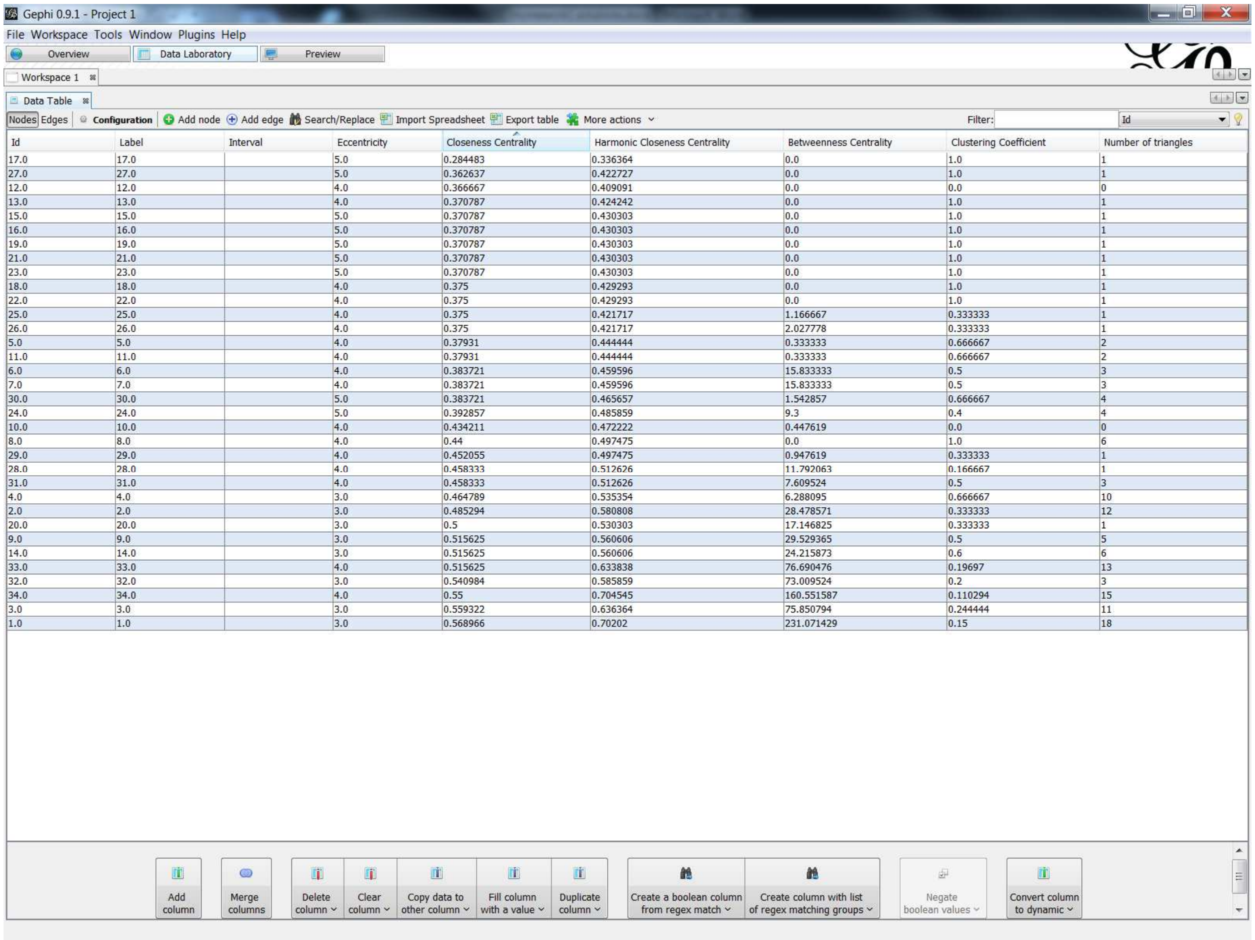
Network Overview
Average Degree
Avg. Weighted Degree
Network Diameter 5
Graph Density
Modularity
PageRank
Connected Components

Node Overview
Avg. Clustering Coefficient 0.588
Eigenvector Centrality

Edge Overview
Avg. Path Length 2.408

Dynamic
Nodes
Edges
Degree
Clustering Coefficient

Spreadsheet:



Id	Label	Interval	Eccentricity	Closeness Centrality	Harmonic Closeness Centrality	Betweenness Centrality	Clustering Coefficient	Number of triangles
17.0	17.0		5.0	0.284483	0.336364	0.0	1.0	1
27.0	27.0		5.0	0.362637	0.422727	0.0	1.0	1
12.0	12.0		4.0	0.366667	0.409091	0.0	0.0	0
13.0	13.0		4.0	0.370787	0.424242	0.0	1.0	1
15.0	15.0		5.0	0.370787	0.430303	0.0	1.0	1
16.0	16.0		5.0	0.370787	0.430303	0.0	1.0	1
19.0	19.0		5.0	0.370787	0.430303	0.0	1.0	1
21.0	21.0		5.0	0.370787	0.430303	0.0	1.0	1
23.0	23.0		5.0	0.370787	0.430303	0.0	1.0	1
18.0	18.0		4.0	0.375	0.429293	0.0	1.0	1
22.0	22.0		4.0	0.375	0.429293	0.0	1.0	1
25.0	25.0		4.0	0.375	0.421717	1.166667	0.333333	1
26.0	26.0		4.0	0.375	0.421717	2.027778	0.333333	1
5.0	5.0		4.0	0.37931	0.444444	0.333333	0.666667	2
11.0	11.0		4.0	0.37931	0.444444	0.333333	0.666667	2
6.0	6.0		4.0	0.383721	0.459596	15.833333	0.5	3
7.0	7.0		4.0	0.383721	0.459596	15.833333	0.5	3
30.0	30.0		5.0	0.383721	0.465657	1.542857	0.666667	4
24.0	24.0		5.0	0.392857	0.485859	9.3	0.4	4
10.0	10.0		4.0	0.434211	0.472222	0.447619	0.0	0
8.0	8.0		4.0	0.44	0.497475	0.0	1.0	6
29.0	29.0		4.0	0.452055	0.497475	0.947619	0.333333	1
28.0	28.0		4.0	0.458333	0.512626	11.792063	0.166667	1
31.0	31.0		4.0	0.458333	0.512626	7.609524	0.5	3
4.0	4.0		3.0	0.464789	0.535354	6.288095	0.666667	10
2.0	2.0		3.0	0.485294	0.580808	28.478571	0.333333	12
20.0	20.0		3.0	0.5	0.530303	17.146825	0.333333	1
9.0	9.0		3.0	0.515625	0.560606	29.529365	0.5	5
14.0	14.0		3.0	0.515625	0.560606	24.215873	0.6	6
33.0	33.0		4.0	0.515625	0.633838	76.690476	0.19697	13
32.0	32.0		3.0	0.540984	0.585859	73.009524	0.2	3
34.0	34.0		4.0	0.55	0.704545	160.551587	0.110294	15
3.0	3.0		3.0	0.559322	0.636364	75.850794	0.244444	11
1.0	1.0		3.0	0.568966	0.70202	231.071429	0.15	18

Therefore: Node 1 or Node 34 has the highest centrality score (depending on which measure was used)

Question 5 (2.0 points) The following URL points to a "coauthorship network" of scientists working on network theory and experiment.

http://networkdata.ics.uci.edu/data.php?id=11

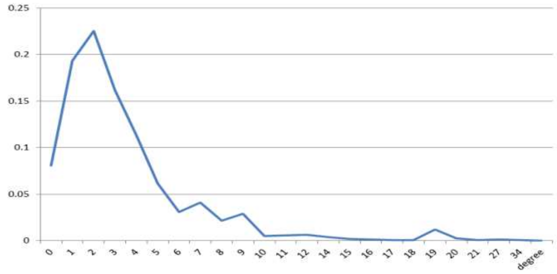
A brief description of the network is given in the "netscience.txt". In "netscience.paj" file (these are text files), you can find nodes and edges between nodes. The names of the scientists (which correspond to the nodes of the networks) are given in "netscience.gml". Please download the dataset and use any program tools you are familiar with to build a network and finish the following tasks:

4. Please report and draw the degree distributions of the whole network [0.25 pt]. Convert the figure to log-log space and validate whether it complies with the power-law distributions [0.25 pt].

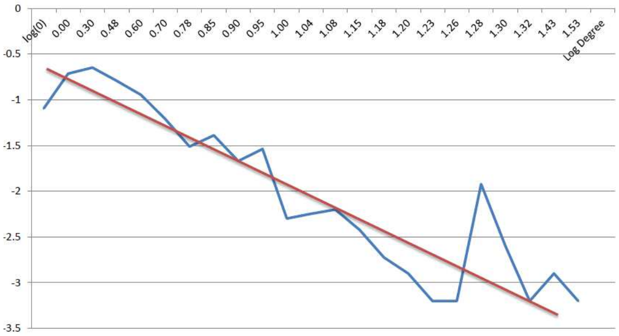
5. Please report cumulative degree distribution of the network [0.25 pt], and convert it to log-log space and validate whether it complies with the power-law distributions [0.25 pt].
6. Please report Rank-Degree distribution of the network [0.25 pt], and convert it to log-log space and validate whether it complies with the power-law distributions [0.25 pt].
7. Please report clustering coefficient and diameter of the network [0.5 pt].

Solution

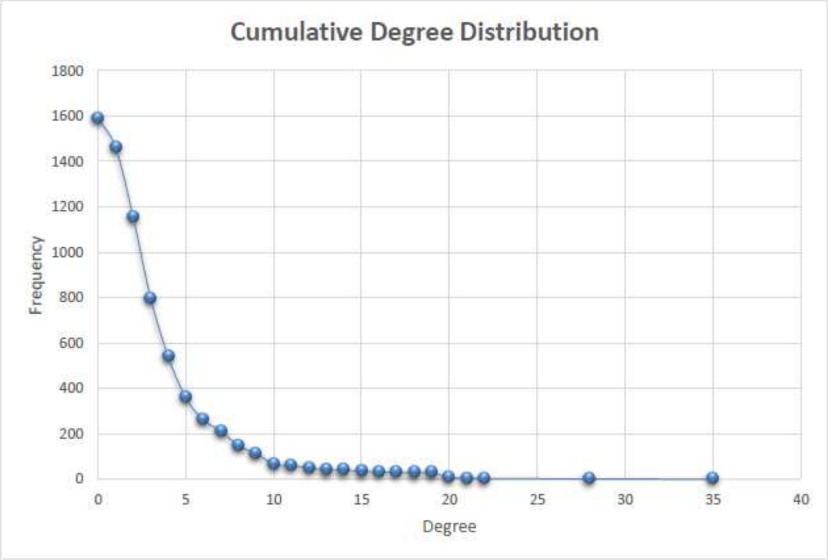
Degree Distribution Results:

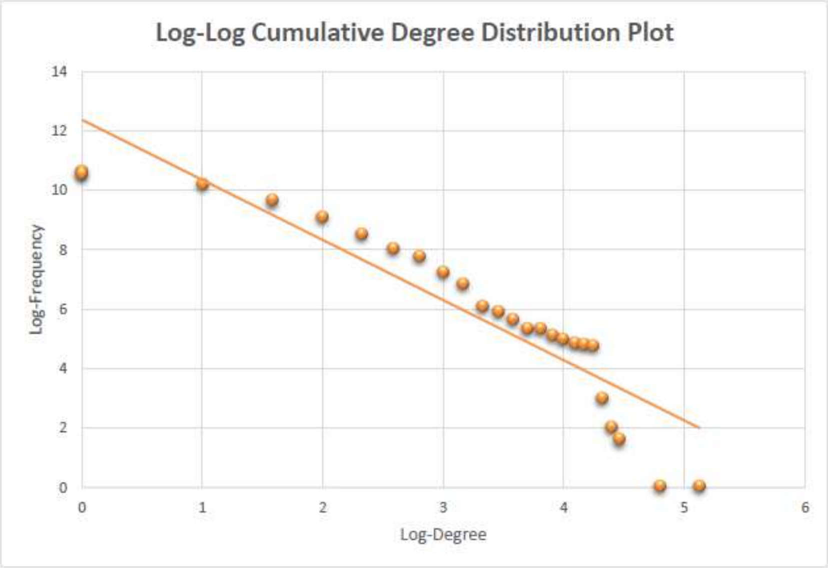


Log-log space

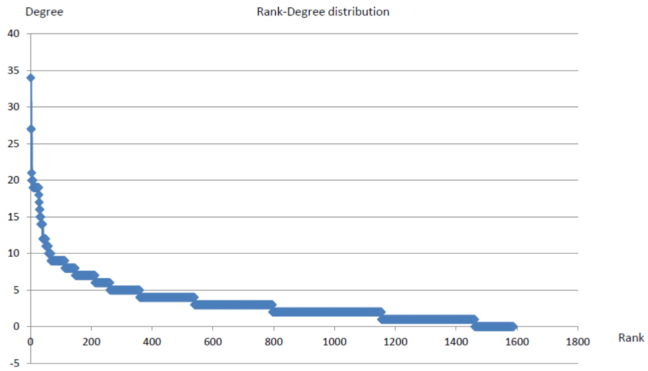


Cumulative degree distribution results

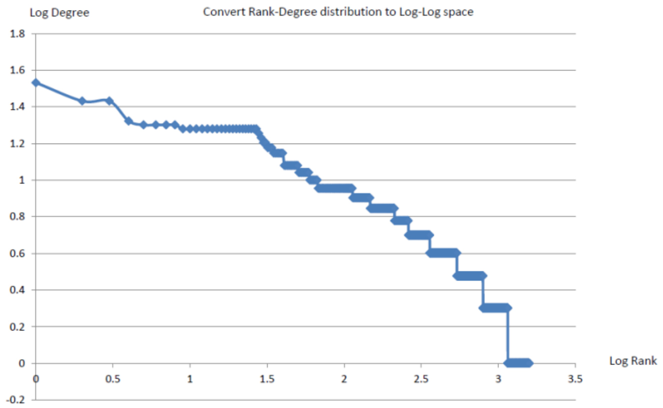




Rank degree distribution results:



Log-log space



Clustering Coefficient: 0.878

Diameter of the Network: 17

Tips for programming:

In order to calculate the average distance between a pair of nodes, and calculate the diameter of the network, you will need to implement algorithms which calculate shortest path between any two nodes. Example of algorithms include

Dijkstra's algorithm (http://en.wikipedia.org/wiki/Dijkstra's_algorithm)

Breath First Search algorithm (http://en.wikipedia.org/wiki/Breadth-first_search)

Alternatively, you may consider following programming tools/packages which are specifically designed for network and graph data. These packages have algorithms for finding shortest path and network diameter.

Gephi: The open Graph Viz Platform

https://gephi.org/

(Please note Gephi also has API functions to support user programming. You can check API functions for the following URL:

https://gephi.org/docs/api/

Java Platform: JUNG (Java Universal Network/Graph Framework)

http://jung.sourceforge.net/

http://www.datalab.uci.edu/papers/JUNG_tech_report.html

Python: NetworkX (High-productivity software for complex networks)

http://networkx.github.com/

.Net: NodeXL (Open source template for Microsoft tools)

http://nodexl.codeplex.com/