CS550: Massive Data Mining and Learning
Homework 3

Due 11:59pm Saturday, Apr 18, 2020
Only one late period is allowed for this homework (11:59pm Sunday Apr 19)

Submission Instructions

Assignment Submission Include a signed agreement to the Honor Code with this assignment. Assignments are due at 11:59pm. All students must submit their homework via Sakai. Students can typeset or scan their homework. Students also need to include their code in the final submission zip file. Put all the code for a single question into a single file.

Late Day Policy Each student will have a total of *two* free late days, and for each homework only one late day can be used. If a late day is used, the due date is 11:59pm on the next day.

Honor Code Students may discuss and work on homework problems in groups. This is encouraged. However, each student must write down their solutions independently to show they understand the solution well enough in order to reconstruct it by themselves. Students should clearly mention the names of all the other students who were part of their discussion group. Using code or solutions obtained from the web is considered an honor code violation. We check all the submissions for plagiarism. We take the honor code seriously and expect students to do the same.

Discussion Group (People with whom you discussed ideas used in your answers):

On-line or hardcopy documents used as part of your answers:

I acknowledge and accept the Honor Code.

*Keya Desai (KD)*_____

If you are not printing this document out, please type your initials above.

Answer to Question 1(a)

Modularity of a network divided into two communities is defined as:

$$Q = \frac{1}{4m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) s_i s_j \tag{1}$$

- Partitioning the graph along (A-G) without removing the edge in Graph G:
  Community label vector, **S = [1, 1, 1, 1, -1, -1, -1, -1]**

  Adjacency Matrix, **A =**
  $$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

  Degree distribution, **k = [4, 3, 3, 3, 2, 2, 4, 1]**
  Number of nodes, **m = 11**

  Substituting the values of A, k, m and s, in Eq. 1, we get $\boxed{Q = \mathbf{0.39256}}$

- Partitioning the graph by **removing edge (A-G)** in Graph G:
  Community label vector, **S = [1, 1, 1, 1, -1, -1, -1, -1]**

  Adjacency Matrix, **A =**
  $$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

  Degree distribution, **k = [3, 3, 3, 3, 2, 2, 3, 1]**
  Number of nodes, **m = 10**

  Substituting the values of A, k, m and s, in Eq. 1, we get $\boxed{Q = \mathbf{0.48}}$

Answer to Question 1(b)

- Adding edge (E-H) to the original network G and recalculating modularity of the partition in 1(a):

S remains same since the partition of the graph is still (A-G). Recalculating A, k, m:

Adjacency Matrix, $\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$

Degree distribution: $\mathbf{k} = [\ \mathbf{4, 3, 3, 3, 3, 2, 4, 2}]$
Number of nodes, $\mathbf{m = 12}$

Substituting the values of A, k, m and s, in Eq. 1, we get $\boxed{Q = \mathbf{0.41319}}$

- Nodes E and H belong to the same community. Adding an edge inside the same community increases the intra-community connectivity and results in better community structure. s_i, s_j values of E and H is same resulting in a product of 1. This leads to addition in the calculation of Q. Hence, the modularity of the network **increases** as compared to 1(a) on adding an edge between E and H.

Answer to Question 1(c)

- Adding edge (A-F) in the original network G and recalculating modularity of the partition in 1(a):
  S remains same since the partition of the graph is still (A-G). Recalculating A, k, m:

  Adjacency Matrix, $\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

  Degree distribution: $\mathbf{k} = [\ \mathbf{5, 3, 3, 3, 2, 3, 4, 1}]$
  Number of nodes, $\mathbf{m = 12}$

  Substituting the values of A, k, m and s, in Eq. 1, we get $\boxed{Q = \mathbf{0.31944}}$

- The modularity (Q) **decreases** as compared to 1(a). Nodes A and F belong to different communities. The aim in partitioning the network is to minimise the inter cluster edges. Adding an edge which crosses the two communities increases the inter-community connectivity and hence decreases the modularity of the network. s_i, s_j values of A and F

are different resulting in a product of -1. This leads to subtraction in the calculation of Q. Hence, the modularity of the network decreases as compared to 1(a) on adding an edge between A and F.

Answer to Question 2(a)

Adjacency Matrix (A), Degree Matrix (D) and Laplacian matrix(L) for the graph are:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$L = D - A = \begin{bmatrix} 4 & -1 & -1 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & 0 & 0 & -1 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

Answer to Question 2(b)

For the Laplacian matrix L given in 2(a), the eigen values and the corresponding eigen vectors are:

```
Eigen values:
[1.9962663954589044e-16, 0.35424868893540984, 0.9999999999999993, 3.0,
3.9999999999999956, 3.999999999999998, 4.000000000000003, 5.645751311064586]

Eigen vectors:
[[-3.53553391e-01 -2.47017739e-01  0.00000000e+00  0.00000000e+00
   6.06945398e-01 -7.04000701e-03 -8.10414811e-02  6.62557346e-01]
 [-3.53553391e-01 -3.82527662e-01  1.02125361e-16 -1.18577334e-15
  -2.85951736e-01 -5.69570946e-01 -5.49685251e-01 -1.42615758e-01]
 [-3.53553391e-01 -3.82527662e-01  8.23418241e-17 -8.78258152e-16
  -1.00691034e-01 -2.16290004e-01  8.07101780e-01 -1.42615758e-01]
 [-3.53553391e-01 -3.82527662e-01 -1.03628893e-16  2.01672234e-15
  -2.20302629e-01  7.92900957e-01 -1.76375048e-01 -1.42615758e-01]
 [-3.53553391e-01  3.82527662e-01 -4.08248290e-01 -7.07106781e-01
  -2.02315133e-01  2.34666900e-03  2.70138270e-02  1.42615758e-01]
 [-3.53553391e-01  3.82527662e-01 -4.08248290e-01  7.07106781e-01
  -2.02315133e-01  2.34666900e-03  2.70138270e-02  1.42615758e-01]
 [-3.53553391e-01  2.47017739e-01 -8.08382926e-17  4.73091557e-17
   6.06945398e-01 -7.04000701e-03 -8.10414811e-02 -6.62557346e-01]
 [-3.53553391e-01  3.82527662e-01  8.16496581e-01  3.89841883e-16
  -2.02315133e-01  2.34666900e-03  2.70138270e-02  1.42615758e-01]]
```

Eigen vector corresponding to each eigen value is given below:

- 1.9962663954589044e-16, $\begin{bmatrix} -0.35355339 \\ -0.35355339 \\ -0.35355339 \\ -0.35355339 \\ -0.35355339 \\ -0.35355339 \\ -0.35355339 \\ -0.35355339 \end{bmatrix}$
- 0.35424868893540984, $\begin{bmatrix} -0.24701774 \\ -0.38252766 \\ -0.38252766 \\ -0.38252766 \\ 0.38252766 \\ 0.38252766 \\ 0.24701774 \\ 0.38252766 \end{bmatrix}$
- 0.9999999999999993, $\begin{bmatrix} 0.00000000e+00 \\ -3.18493382e-17 \\ 8.59836280e-17 \\ -5.79022479e-17 \\ -4.08248290e-01 \\ -4.08248290e-01 \\ 3.76795815e-18 \\ 8.16496581e-01 \end{bmatrix}$
- 3.0, $\begin{bmatrix} 0.00000000e+00 \\ -2.70599246e-17 \\ -1.12776339e-16 \\ 1.50488323e-16 \\ 7.07106781e-01 \\ -7.07106781e-01 \\ -1.06520593e-17 \\ 1.12242936e-16 \end{bmatrix}$
- 3.9999999999999956, $\begin{bmatrix} 0.60717154 \\ -0.27939608 \\ -0.1005666 \\ -0.22720886 \\ -0.20239051 \\ -0.20239051 \\ 0.60717154 \\ -0.20239051 \end{bmatrix}$

- 3.999999999999998, $\begin{bmatrix} 0.00000000e+00 \\ 5.62206567e-01 \\ 2.31676233e-01 \\ -7.93882800e-01 \\ 2.16840434e-16 \\ -2.82759927e-16 \\ -1.11022302e-16 \\ -1.71737624e-16 \end{bmatrix}$
- 4.000000000000003, $\begin{bmatrix} -0.07964119 \\ -0.56053094 \\ 0.80283611 \\ -0.16266398 \\ 0.02654706 \\ 0.02654706 \\ -0.07964119 \\ 0.02654706 \end{bmatrix}$
- 5.645751311064586, $\begin{bmatrix} 0.66255735 \\ -0.14261576 \\ -0.14261576 \\ -0.14261576 \\ 0.14261576 \\ 0.14261576 \\ -0.66255735 \\ 0.14261576 \end{bmatrix}$

Answer to Question 2(c)

- Second smallest eigen value $\lambda_2 = 0.35424868893540984$
- Eigen vector corresponding to $\lambda_2 =$ [−0.24701774 − 0.38252766 − 0.38252766 − 0.38252766 0.38252766 0.38252766 0.24701774 0.38252766]
- Using 0 as boundary, the partition obtained of the graph is:

  Cluster 1: Negative points

Node IDs	**Node**	**Eigen vector**
1	A	-0.24701774
2	B	-0.38252766
3	C	-0.38252766
4	D	-0.38252766

Cluster 2: Positive points

Node IDs	**Node**	**Eigen vector**
5	E	0.38252766
6	F	0.38252766
7	G	0.24701774
8	H	0.38252766

Answer to Question 3(a)

For any integer i > 1, the set C_i of nodes of G that are divisible by i is a clique because node i will have an edge with every node j in the set since j is divisible by i. And every node j $\neq$ i, will have an edge with node k $\neq$ i, j since integer j and k have a common factor i.

Answer to Question 3(b)

C_i will be a maximal clique for every prime number i < 1000000.

- A clique C is maximal when every node not in C is missing an edge to atleast one member of C.
- If i >1000000, C_i is an empty clique.
- If i is not a prime number: Consider an integer j which is a factor of i such that 1<j<i. Since j is not divisible by i, it will not be in C_i. But node j will have an edge with every member of C_i because j is a common factor. Hence, C_i will not be maximal.
- If i is prime, then there will be no node which is not in C_i and has an edge with i itself. To prove this, let j be such a node. j can not be a factor of i because i is a prime number. But since there is an edge between i and j, j must be a multiple of i and therefore should already be in C_i. Hence C_i is maximal.

Answer to Question 3(c)

We proved in 3(b) that C_i can be maximal only for i = prime number. Out of all the prime numbers, i = 2 has the maximum multiples in the set and hence C_2 will have maximum elements. Therefor C_2 is the largest maximal clique possible.