CS550: Massive Data Mining and Learning
Homework 3

Due 11:59pm Monday, Apr 18, 2022

Submission Instructions

Assignment Submission Include a signed agreement to the Honor Code with this assignment. Assignments are due at 11:59pm. Students should submit their homework via Canvas. 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. Finally, put your PDF answer file and all the code files in a folder named as your Name and NetID (i.e., Firstname-Lastname-NetID.pdf), compress the folder as a zip file (e.g., Firstname-Lastname-NetID.zip), and submit the zip file via Canvas.

Late Policy: The homework is due on 4/18 (Monday) at 11:59pm. We will release the solutions of the homework on Canvas on 4/22 (Friday) 11:59pm. If your homework is submitted to Canvas before 4/18 11:59pm, there will no late penalty. If you submit to Canvas after 4/18 11:59pm and before 4/22 11:59pm, your score will be penalized by 0.9^k, where k is the number of days of late submission. For example, if you submitted on 4/21, and your original score is 80, then your final score will be $80 \times 0.9^3 = 58.32$ for $22 - 18 = 3$ days of late submission. If you submit to Canvas after 4/22 11:59pm, then you will earn no score for the homework.

Honor Code Students may discuss homework problems with peers. 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 with whom they discussed the homework. Directly using code or solutions obtained from others or 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.

(Signed) SK _____

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

Answer to Question 1(a)

Adjacency matrix of graph G, 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 = $\begin{bmatrix} 4 & 3 & 3 & 3 & 2 & 2 & 4 & 1 \end{bmatrix}$

Community label vector, S = $\begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \end{bmatrix}$

Number of nodes, m = 11

Modularity is defined as,

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

Substituting these values, we will get

$$\boxed{\mathbf{Q \ = 0.39256}}$$

Now partitioning the graph by removing edge (A, G) in Graph G

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 = $\begin{bmatrix} 3 & 3 & 3 & 3 & 2 & 2 & 3 & 1 \end{bmatrix}$

Community label vector, S = $\begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \end{bmatrix}$

Number of nodes, m = 10

Substituting these values in Modularity equation, we will get

$$\boxed{\mathbf{Q \ = 0.48}}$$

Answer to Question 1(b)

Adding edge (E, H) and recalculating modularity of the partition,

Adjacency matrix A = $\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$

Degree Distribution, k = $\begin{bmatrix} 4 & 3 & 3 & 3 & 3 & 2 & 4 & 2 \end{bmatrix}$

Community label vector will be same, S = $\begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \end{bmatrix}$

Number of nodes, m = 12

Substituting these values in Modularity equation, we will get

$$\boxed{\mathbf{Q \ = 0.41319}}$$

The modularity Q go up as compared to 1(a) by adding edge (E, H). As we can see the nodes E and H belong to the same community, s_i, s_j value will be same and product will be 1. Therefore, adding an edge in original graph which is inside one of the community will increase intra-community connectivity and results in better community structure. Therefore, modularity of the network increases as compared to 1a on adding edge between E and H.

Answer to Question 1(c)

Adding edge (F, A) and recalculating modularity of the partition,

Adjacency matrix 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, k = $\begin{bmatrix} 5 & 3 & 3 & 3 & 2 & 3 & 4 & 1 \end{bmatrix}$

Community label vector will be same, S = $\begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \end{bmatrix}$

Number of nodes, m = 12

Substituting these values in Modularity equation, we will get

$$\boxed{\mathbf{Q \ = 0.31944}}$$

The modularity Q go down as compared to 1(a) by adding edge (F, 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. As we can see the nodes F and A belong to the diferent community, s_i, s_j value will be different and product will be -1. Therefore, modularity of the network decreases as compared to 1a on adding edge between F and A.

Answer to Question 2(a)

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

&

Laplacian matrix 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)

Code is written in Q2.ipynb and results are as follows:

The eigen values and corresponding eigen vectors are as follows:

```
Eigenvalue and corresponding eigenvectors:-

eigenvalue  1 :-  2.1403818880962127e-16  -  [-0.35355339 -0.35355339 -0.35355339 -0.35355339 -0.35355339 -0.35355339
 -0.35355339 -0.35355339]

eigenvalue  2 :-  0.3542486889354087  -  [-0.24701774 -0.38252766 -0.38252766 -0.38252766  0.38252766  0.38252766
  0.24701774  0.38252766]

eigenvalue  3 :-  1.0000000000000049  -  [ 0.00000000e+00 -3.18493382e-17  8.59836280e-17 -5.79022479e-17
 -4.08248290e-01 -4.08248290e-01  3.76795815e-18  8.16496581e-01]

eigenvalue  4 :-  3.0000000000000036  -  [ 0.00000000e+00 -2.70599246e-17 -1.12776339e-16  1.50488323e-16
  7.07106781e-01 -7.07106781e-01 -1.06520593e-17  1.12242936e-16]

eigenvalue  5 :-  3.999999999999996  -  [ 0.60717154 -0.27939608 -0.1005666  -0.22720886 -0.20239051 -0.20239051
  0.60717154 -0.20239051]

eigenvalue  6 :-  4.000000000000001  -  [ 0.00000000e+00  5.62206567e-01  2.31676233e-01 -7.93882800e-01
  2.16840434e-16 -2.82759927e-16 -1.11022302e-16 -1.71737624e-16]

eigenvalue  7 :-  4.000000000000002  -  [-0.07964119 -0.56053094  0.80283611 -0.16266398  0.02654706  0.02654706
 -0.07964119  0.02654706]

eigenvalue  8 :-  5.645751311064582  -  [ 0.66255735 -0.14261576 -0.14261576 -0.14261576  0.14261576  0.14261576
 -0.66255735  0.14261576]
```

Answer to Question 2(c)

Code is written in Q2.ipynb and results are as follows:

```
Second smallest eigen value =  0.3542486889354087
Corresponding Eigen vector =
 [-0.24701774 -0.38252766 -0.38252766 -0.38252766  0.38252766  0.38252766
  0.24701774  0.38252766]
```

Partitioning the graph into 2 communities using 0 boundry, we get

Community 1: positive points

NodeIDs	Node	Eigenvector
5	E	0.3825276
6	F	0.3825276
7	G	0.24701774
8	H	0.3825276

Community 2: negative points

NodeIDs	Node	Eigenvector
1	A	-0.24701774
2	B	-0.38252766
3	C	-0.38252766
4	D	-0.38252766

Answer to Question 3(a)

If i is any integer greater than 1, then the set C_i of nodes of G that are divisible by i is a clique. This is because any two nodes in C_i will have at least one common factor, i.e. i. Therefore, any two nodes in set C_i will have an edge between them. That is the reason why we can say that C_i is a clique.

Answer to Question 3(b)

- A clique C is maximal when every node not in C is missing an edge to at least one member of C.

- If i $>$ 1000000, C_i is an empty clique.

- Case 1 : If i is not a prime number: Consider an integer j which is a factor of i such that $1 < j < i$. As 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. Therefore, none of the node is missing an edge and C_i will not be maximal.

- Case 2 : If i is a prime number: If i is prime, then there will be no node which is not in C_i and has an edge with i itself. Let j be such node. j cannot be a factor of i because i is a prime number. Since there is an edge between i and j, j must be a multiple of i and hence should already be in C_i. Therefore, C_i is maximal.
Hence, C_i will be a maximal clique for every prime number i $<$ 1000000

Answer to Question 3(c)

We already know that C_i will be a maximal clique for every prime number. From all the prime members, i=2 has maximum multiples in the set and therefore, C_2 has maximum elements in the set as compared to other maximal cliques. Hence, C_2 is the largest maximal clique possible and therefore is a unique maximal clique.