CS550: Massive Data Mining and Learning
Homework 3

Due 11:59pm Saturday, Apr 20, 2019
Only one late period is allowed for this homework (11:59pm Sunday Apr 21)

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.

*(Signed)*Akhilesh Harish Mahajan_____

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

Answer to Question 1(a)

From inspection of graph as given in question 1a, we have:

	A	B	C	D	E	F	G	H
s_i	1	1	1	1	-1	-1	-1	-1
k_i	3	3	3	3	2	2	3	1

We know that
$Q = \frac{1}{4m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) s_i s_j$
Here, $m = 10$, $A_{ij} = 1$ if there is an edge between i and j, $A_{ij} = 0$ if there is no edge between i and j.
Using these substitutions in our modularity equation, we have:
$Q = 0.48$
We have neglected the calculation math for convenience and conciseness as desired in the question.
$\therefore \underline{Final\ Answer : Q = 0.48}$

Answer to Question 1(b)

From inspection of graph as given in question 1b, we have:

	A	B	C	D	E	F	G	H
s_i	1	1	1	1	-1	-1	-1	-1
k_i	4	3	3	3	3	2	4	2

We know that
$Q = \frac{1}{4m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) s_i s_j$
Here, $m = 12$, $A_{ij} = 1$ if there is an edge between i and j, $A_{ij} = 0$ if there is no edge between i and j.
Using these substitutions in our modularity equation, we have:
$Q = 0.4131944$
We have neglected the calculation math for convenience and conciseness as desired in the question.
$\therefore \underline{Final\ Answer : Q = 0.4131944}$

Here, the modularity went down, as we have recovered the edge AG, which is between the 2 different groups.

Answer to Question 1(c)

From inspection of graph as given in question 1c, we have:

	A	B	C	D	E	F	G	H
s_i	1	1	1	1	-1	-1	-1	-1
k_i	5	3	3	3	2	3	4	1

We know that
$Q = \frac{1}{4m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) s_i s_j$
Here, $m = 12$, $A_{ij} = 1$ if there is an edge between i and j, $A_{ij} = 0$ if there is no edge between i and j.
Using these substitutions in our modularity equation, we have:
$Q = 0.31944$
We have neglected the calculation math for convenience and conciseness as desired in the question.
$\therefore \underline{Final\ Answer : Q = 0.31944}$

Here, the modularity went further down, as we now have 2 edges AG and AF between 2 different groups.

Answer to Question 2(a)

Matrix A is:

$$A = \begin{bmatrix} & A & B & C & D & E & F & G & H \\ A & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ B & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ C & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ D & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ F & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ G & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ H & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Matrix D is:

$$D = \begin{bmatrix} & A & B & C & D & E & F & G & H \\ A & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ B & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ D & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ F & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ G & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ H & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$\therefore$ Laplacian Matrix $L = D - A$:

$$L = \begin{bmatrix} & A & B & C & D & E & F & G & H \\ A & 4 & -1 & -1 & -1 & 0 & 0 & -1 & 0 \\ B & -1 & 3 & -1 & -1 & 0 & 0 & 0 & 0 \\ C & -1 & -1 & 3 & -1 & 0 & 0 & 0 & 0 \\ D & -1 & -1 & -1 & 3 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 2 & -1 & -1 & 0 \\ F & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ G & -1 & 0 & 0 & 0 & -1 & -1 & 4 & -1 \\ H & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

Answer to Question 2(b)

Eigenvalues:
array([7.50344604e-16, 3.54248689e-01,
1.00000000e+00, 3.00000000e+00, 4.00000000e+00,
4.00000000e+00, 4.00000000e+00])
For code, please see the attached file:question2.ipynb

Answer to Question 2(c)

The eigenvector corresponding to the second smallest eignevalue:
array([-0.24701774, -0.38252766, -0.38252766,
-0.38252766, 0.38252766,
0.38252766, 0.24701774, 0.38252766])

Using 0 as the boundary, we can partition the graph into 2 communities. The communities are: -1, -1, -1, -1, 1, 1, 1, 1 for A, B, C, D, E, F, G, H respectively. That is, the graph G is partitioned as (A, B, C, D) in one partition, and as (E, F, G, H) in another partition.

Answer to Question 3(a)

Assume $2 \leq i \leq 1000000$. In this case, if we make a set C_i of nodes that are divisible by i, then all the nodes in this set shall have an edge between them and all the remaining nodes. This is the definition of clique. Hence, set C_i is a clique.
If C_i has fewer than 2 elements, then it is a clique by definition.

Answer to Question 3(b)

C_i is a maximal clique, if i is a prime less than one million.
If $i \leq 1000000$, but not prime, then let j be a factor of i, such that $1 < j < i$. In this case, we can add j to C_i, and still let C_i be a clique, as j is a factor of i, and hence is a factor of all the nodes in C_i. Thus, C_i is not maximal, as we can add more nodes.
Conversely, let us prove that if C_i is maximal, then i is prime. Let j be a node that is not in C_i, but can be added to C_i. $\therefore$ It has to be either a factor of i, which is not possible, since i is prime, or it has to be some multiple of i, which is already in C_i. Hence, we can't find any such nodes to add to C_i when i is prime.
Hence proved.

Answer to Question 3(c)

Let C_2 be a clique. Assume that there is some node j, which can be added to C_2. This is possible only when j is a factor of i, which in this case, is not possible, since there is no i less than 2, or when j is a multiple of i. But if j is a multiple of i, then it is already in C_i. Hence, C_2 is a maximal clique.
Let us compare C_2 with C_3. Since 1million/2 = 0.5 million, our size of C_2 is 0.5 million. Also, since 1million/3 = 0.33 million, our size of C_3 is 0.33 million. We can make a similar argument for all primes greater then 2. Hence, $|C_2| > |C_i| for\ i = prime, i > 2$
Let us say that i=6. That is, we are considering the case of C_6. We shall have all the nodes in C_6 initially divisible by 6. Now, we can either add 2 or 3 to increase it's size. But, we have see that for all primes, C_2 has the maximum size. $\therefore$ We shall add all the nodes divisible by 2 to C_6. Hence, maximum size of C_6 is essentially the size of C_2. We can make similar arguments for all i that are divisible by some non-trivial factor.
Hence, $|C_2| \geq |C_i| for\ i > 2$