CS550: Massive Data Mining and Learning Spring 2022
Problem Set 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)___Vaishnavi Manthena_________

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

QUESTION 1

Note: I was a little confused about using AG edge or not so, did both versions in each part. The code to implement the calculations specified in the below problems is in question_1.py.

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

Answer to Question 1(a)

Modularity of {A, B, C, D} and {E, F, G, H} considering the complete graph of the figure: 0.392562

Calculation: Substituting the following concrete values into the above equation:

A =

	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

K Values (degrees):

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

Vertices of top component ({A, B, C, D}) have $s_i = 1$.
Vertices of bottom component ({E, F, G, H}) have $s_i = -1$.

m = # of edges in graph = 11

I will call all these above parameters as "INITIAL PARAMETERS."

Modularity of {A, B, C, D} and {E, F, G, H} considering the complete graph of the figure excluding AG: 0.48

Calculation: Substitute into equation for Q, initial parameters with the following changes:

- $A_{AG} = 0$
- $A_{GA} = 0$
- K values of A and G are 3 instead of 4
- m = 10

Answer to Question 1(b)

Modularity of {A, B, C, D} and {E, F, G, H} considering the complete graph of the figure along with the edge EH: 0.413194

Calculation: Substitute into equation for Q, initial parameters with the following changes:

- $A_{EH} = 1$, $A_{HE} = 1$
- K[E] = 3, K[H] = 2
- m = 12

Modularity of {A, B, C, D} and {E, F, G, H} considering the complete graph of the figure along with the edge EH and excluding AG: 0.495868

Calculation: Substitute into equation for Q, initial parameters with the following changes:

- $A_{AG} = 0$, $A_{GA} = 0$, $A_{EH} = 1$, $A_{HE} = 1$
- K[A] = 3, K[G] = 3, K[E] = 3, K[H] = 2
- m = 11

Conclusion: In both corresponding cases the modularity has increased (0.413194 > 0.392562 AND 0.495868 > 0.48). The modularity goes up because there is increased interconnectivity within the second partition. Modularity increases with increase in connectedness within partitions.

Answer to Question 1(c)

Modularity of {A, B, C, D} and {E, F, G, H} considering the complete graph of the figure with the edge AF: 0.31944

Calculation: Substitute into equation for Q, initial parameters with the following changes:

- $A_{AF} = 1$, $A_{FA} = 1$
- K[A] = 5, K[F] = 3
- m = 12

Modularity of {A, B, C, D} and {E, F, G, H} considering the complete graph of the figure with the edge AF and excluding AG: 0.39256

Calculation: Substitute into equation for Q, initial parameters with the following changes:

- $A_{AG} = 0$, $A_{GA} = 0$, $A_{AF} = 1$, $A_{FA} = 1$
- K[A] = 4, K[F] = 3, K[G]=3
- m = 11

Conclusion: In both corresponding cases the modularity has decreased (0.31944 < 0.392562 AND 0.39256 < 0.48). The modularity goes down because there is increased interconnectivity between the partitions. Modularity decreases with increase in edges going across partitions.

Answer to Question 2(a)

Adjacency Matrix:

	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

Degree Matrix:

	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

Laplacian Matrix:

	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

Answer to Question 2(b)

Code: Given in question_2.py

<u>Eigen values:</u>

[3.80005615e-16 3.54248689e-01 1.00000000e+00 3.00000000e+00
 4.00000000e+00 4.00000000e+00 4.00000000e+00 5.64575131e+00]

<u>Corresponding Eigen Vectors (each column):</u>

```
[-3.53553391e-01 -2.47017739e-01  0.00000000e+00  0.00000000e+00 6.11249792e-01  9.41577125e-03  3.58473880e-02 -6.62557346e-01]
[-3.53553391e-01 -3.82527662e-01  1.90866342e-17 -9.68276432e-17 -1.65195973e-01 -6.58691980e-01 -4.97160436e-01  1.42615758e-01]
[-3.53553391e-01 -3.82527662e-01 -7.39522061e-18  3.39825436e-16  -2.49799723e-01 -9.67513902e-02  7.97854999e-01  1.42615758e-01]
[-3.53553391e-01 -3.82527662e-01  2.88872092e-16 -1.06734799e-16 -1.96254096e-01  7.46027599e-01 -3.36541951e-01  1.42615758e-01]
[-3.53553391e-01  3.82527662e-01 -4.08248290e-01  7.07106781e-01 -2.03749931e-01 -3.13859042e-03 -1.19491293e-02 -1.42615758e-01]
[-3.53553391e-01  3.82527662e-01 -4.08248290e-01 -7.07106781e-01 -2.03749931e-01 -3.13859042e-03 -1.19491293e-02 -1.42615758e-01]
[-3.53553391e-01  2.47017739e-01 -1.72288891e-16 -1.36262993e-16 6.11249792e-01  9.41577125e-03  3.58473880e-02  6.62557346e-01]
[-3.53553391e-01  3.82527662e-01  8.16496581e-01  3.31146360e-17 -2.03749931e-01 -3.13859042e-03 -1.19491293e-02 -1.42615758e-01]
```

Column 'i' above is the eigen vector corresponding to ith eigen value in the list.

Even after sorting, I got same result because eigen values are already in ascending order.

Answer to Question 2(c)

Eigen vector corresponding to second smallest eigen value (3.54248689e-01):

$$\begin{bmatrix} \mathbf{-2.47017739e-01} \\ \mathbf{-3.82527662e-01} \\ \mathbf{-3.82527662e-01} \\ \mathbf{-3.82527662e-01} \\ \mathbf{3.82527662e-01} \\ \mathbf{3.82527662e-01} \\ \mathbf{2.47017739e-01} \\ \mathbf{3.82527662e-01} \end{bmatrix}$$

Using 0 as the boundary, the first 4 vertices (A, B, C, D) that correspond to the negative components of the eigen vector corresponding to the second smallest eigen value form one partition and the last 4 vertices (E, F, G, H) that correspond to the positive components of the eigen vector corresponding to the second smallest eigen value form the other partition.

Partition 1: A, B, C, D
Partition 2: E, F, G, H

Answer to Question 3(a)

$i: any\ integer\ greater\ than\ 1.$
$C_i: set\ of\ nodes\ of\ G\ that\ are\ divisible\ by\ i.$

If we choose two arbitrary nodes of G. These nodes have a common factor 'i' that is not 1. So, there is an edge between these nodes. Therefore, any two nodes of C_i have an edge between each other. So, the edge density of this set of nodes is 1. So, C_i is a clique.

Answer to Question 3(b)

C_i is a maximal clique iff 'i' is a prime number between 2 and 1000000.

<u>Proof by contradiction of forward direction: If C_i is a maximal clique, then 'i' is a prime number between 2 and 1000000.</u>

Assume the contradiction statement: C_i is a maximal clique. 'i' is not a prime number between 2 and 1000000.

Case 1: 'i' is greater than 1000000.
In this case none of the nodes in the graph (2 to 10^6) are divisible by 'i.' Therefore, C_i is an empty set which is not maximal, since you can add a node to the empty set and get a clique. This is a contradiction.

Case 2: 'i' is a number between 2 and 10^6 that is not a prime.
Since 'i' is not prime, it has a factor (say 'm') apart from 1 and itself. Any number divisible by 'i' is also divisible by 'm,' but not vice versa. So, C_i is a proper subset of C_m. So, C_i, is not a maximal clique since the elements of $C_m - C_i$ could be added to it to form a larger clique. This is a contradiction.

<u>Proof of reverse direction: If 'i' is a prime number between 2 and 1000000, then C_i is a maximal clique.</u>

C_i consists of all multiples of the prime number 'i' and this includes 'i' itself.

Consider an arbitrary element 'm' that is not divisible by 'i.' 'm' and 'i' are relatively prime, because the only factor (apart from 1) of 'i' is 'i' itself and this is not a factor of 'm.' So, there is no edge between 'm' and 'i.' So, any arbitrary node 'm' not in the clique C_i is missing an edge with at least one node in C_i (for example 'i' as just explained). Therefore, C_i is a maximal clique.

Answer to Question 3(c)

To prove: C_2 is a unique maximal clique.

It is maximal because it satisfies the necessary and sufficient condition above (2 is a prime between 2 and 1000000).

C_2 consists of all the even nodes between 2 and 1000000. The size of C_2 is 500000.

Claim 1: There is no clique that is larger than 500000 nodes.
Every pair of consecutive integers are relatively prime. If we try to divide $[2, 10^6]$ into buckets of size 2 we get: [2, 3], [4, 5], [6, 7]….., [999998, 999999], $[10^6]$. A clique cannot consist of 2 elements belonging to the same bucket, because there are relatively prime and will not have an edge in between. So, the size of a clique cannot exceed the number of buckets (500000).

Claim 2: There is no other clique with size 500000.
We can consider C_2 as a clique formed by taking one element from each of the buckets above. We do this by choosing the even element from each bucket. However, there is no other way to choose one element from each bucket to form a distinct clique with 500000 elements. This is clear by considering the first two buckets as illustrated in table below:

Possible element combination	Can they be part of same clique?
[2, 4]	Yes. Common factor 2
[2, 5]	No. Relatively prime.
[3, 4]	No. Consecutive and hence relatively prime.
[3, 5]	No. Relatively prime.

So, if we try to choose one element from each of the first two buckets it has to be 2 and 4. Now, if we continue to select one value from each bucket we cannot select and odd number since it will be relatively prime with 2. Hence, the only way to select 500000 elements and form a clique is to select all even nodes.

From the claims 1 and 2, C_2 is a unique maximal clique.