Problem Set: Lesson 3.3

CDS 292. Fall, 2023

Instructions

In this document you will find a set of problems intended to be answered **by hand**, scanned and submitted to Blackboard. Make sure your document is well organized and contains readable responses to the questions. **Show your work! Full credit requires this. An answer without procedure is likely to elicit zero points.** There is one addendum to this handwritten rule: any required plot can be either scanned or sent as an additional file to your submission. If you choose to submit separate files with your plots, make sure they are labeled and can be easily identified.

Note: `Python` code questions will occur in problem sets. The answers have to be handwritten and must respect all `Python` syntax rules including proper indentation of code.

Total Points: 13

---

1. Apply the formula $k_i = \sum_{j=1}^{n} a_{i,j}$ to determine the degree of the node specified in the following matrices (make sure you show your work)

   (a) *[1 point]* Node $i = 3$ in the following matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

   (b) *[1 point]* Node $i = 6$ in the following matrix.

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} & a_{1,6} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} & a_{2,6} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} & a_{3,6} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & a_{4,5} & a_{4,6} \\ a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} & a_{5,5} & a_{5,6} \\ a_{6,1} & a_{6,2} & a_{6,3} & a_{6,4} & a_{6,5} & a_{6,6} \end{pmatrix}, \quad \text{where } a_{i,j} = \begin{cases} 1, & \mid i - j \mid = 1 \\ 0, & \text{otherwise} \end{cases}$$

   Not for credit: do you know what this network looks like?

2. Each line might be a list of degrees for a network. Write down for each whether they can indeed be a network (or not) and briefly say why.

   (a) *[1 point]* $k_1 = 3, k_2 = 4, k_3 = 4, k_4 = 3, k_5 = 3$

   (b) *[1 point]* $k_1 = 1, k_2 = 1, k_3 = 2, k_4 = 2$

3. *[1 point]* Assume that an undirected network with nodes labelled $1, 2, 3, 4$ has adjacency matrix $\mathbf{A}$ shown below in which node 1 is in row/column 1, node 2 in row/column 2, etc. (the missing items are symbolized by $\sqcup$).

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ \sqcup & \sqcup & \sqcup & 1 \\ \sqcup & \sqcup & \sqcup & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Indicate for which nodes one can calculate the degree and specify the value of those degrees.

4. Network G has node set $V(G) = \{\alpha, \beta, \gamma, \delta, \epsilon\}$

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

   (a) *[$^1\!/_2$ point]* Write `Python` code to create matrix $\mathbf{A}$, save it as `A`.

   (b) *[$^1\!/_2$ point]* Draw the network (using `Python`) and attach the plot.

   (c) *[2 points]* Only by using `A`, calculate the degree of each node and print the results.

5. *[3 points]* The HR department of a large company sends out a survey to all its employees. Each person i is asked to give a list of the teammates (also workers in the same company) with whom they have completed any work-related projects while they have been in the company. The number of co-workers person i names is labeled k_i. HR adds all the named teammates over all the workers, i.e. $\sum_i k_i$ and the result is 800. HR also calculates the average number of teammates per worker, and find that the value is 4. Find the number of workers in the company, and the overall number of pairs of workers that have been part of work-related projects. What is the "density" of work collaborations in this company?

6. Network D has $n = 50$ and $\rho = 0.08$.

   (a) *[1 point]* Obtain the number of links for network D (do this by direct calculation, not using `Python`)

   (b) *[1 point]* Create `Python` code to create network D. You can choose the label for the nodes and the link set. Calculate the density for network D using the `Python` function `density()`. You should obtain 0.08

Problem Set 3.3

① $k_i = \sum_{j=1}^{n} a_{ij}$ degree of node is equal to sum of all its link indicators.

$k_1 = a_{11} + a_{12} + a_{13} + \ldots + a_{1n}$

Determine degree of node specified in following matrixe's

ⓐ Determine degree of node 3 in the following matrix, using formula

$$A = \begin{array}{c|c} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

$n = 5$, 5×5 matrix

$$k_3 = \sum_{j=1}^{5} a_{3j}, \quad k_3 = \overset{1}{a_{31}} + \overset{1}{a_{32}} + \overset{0}{a_{33}} + \overset{0}{a_{34}} + \overset{0}{a_{35}} = 2$$

$\boxed{k_3 = 2}$

ⓑ

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{pmatrix}, \quad a_{ij} = \begin{cases} 1, \text{if } |i-j| = 1 \\ 0, \text{otherwise} \end{cases}$$

$$A = \begin{array}{c|c} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} \varnothing & 1 & \varnothing & \varnothing & \varnothing & \varnothing \\ 1 & \varnothing & 1 & \varnothing & \varnothing & \varnothing \\ \varnothing & 1 & \varnothing & 1 & \varnothing & \varnothing \\ \varnothing & \varnothing & 1 & \varnothing & 1 & \varnothing \\ \varnothing & \varnothing & \varnothing & 1 & \varnothing & 1 \\ \varnothing & \varnothing & \varnothing & \varnothing & 1 & \varnothing \end{pmatrix} \end{array}$$

→

Problem Set 3.3

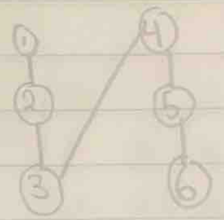
(b.) Find the degree of node 6.

$k_6 = \sum_{j=1}^{6} a_{ij}$, degree equal to sum of all it's link indicators.

$k_6 = a_{61} + a_{62} + a_{63} + a_{64} + a_{65} + a_{66}$

$k_6 = \emptyset + \emptyset + \emptyset + \emptyset + 1 + \emptyset = 1$

$\boxed{k_6 = 1}$



(2) Each example may be a list of degrees for network. Write down for each example whether they can be a network, or if they can't.

(a.) $k_1 = 3$, $k_2 = 4$, $k_3 = 4$, $k_4 = 3$, $k_5 = 3$

$n = 5$

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

k_5 can't be 3

max links in network $= (20)/2 = 1\emptyset$

We can use a matrix to determine if the list of possible degrees can create a network. Since self-links must be $\emptyset$, we can't create a matrix nor network with the degrees given.

Problem Set 3.3

(6) $k_1 = 1$, $k_2 = 1$, $k_3 = 2$, $k_4 = 2$

$n = 4$, 4×4 matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & & \end{pmatrix}$$

Not possible, because row must equal column in matrix

(3) Undirected network, with nodes 1-4 12/2 = 6 max links

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ \sqcup & \sqcup & \sqcup & 1 \\ \sqcup & \sqcup & \sqcup & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Diagonal must be zero. Already have 6 links matching rows/columns

We can calculate the degrees of the following nodes: 1, 4, in the above matrix.

Realistically, though, we can calculate the degree of all nodes since row must be equal to column. Thus, we can calculate the degrees of nodes 1, 2, 3, and 4.

$k_1 = 2$	$k_3 = 1$
$k_2 = 2$	$k_4 = 1$

Diagonal must be off. 6 max links, thus $a_{2,3}$ and a_{32} are zero

Problem Set 3.3

(4) Network G has a node set; $V(G) = \{\alpha, \beta, \gamma, \delta, \varepsilon\}$ in that order. The networks matrix can be represented:

$$
A = \begin{array}{c} \\ \alpha \\ \beta \\ \gamma \\ \delta \\ \varepsilon \end{array}
\begin{array}{c} \begin{array}{ccccc} \alpha & \beta & \gamma & \delta & \varepsilon \end{array} \\
\begin{pmatrix}
0 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 0
\end{pmatrix} \end{array}
$$

a.) Need python code to create matrix A, save as A.

$$
\begin{array}{c} \\ \alpha \\ \beta \\ \gamma \\ \delta \\ \varepsilon \end{array}
\begin{array}{c} \begin{array}{ccccc} \alpha & \beta & \gamma & \delta & \varepsilon \end{array} \\
\begin{pmatrix}
a_{\alpha\alpha} = 0 & a_{\alpha\beta} = 1 & a_{\alpha\gamma} = 1 & a_{\alpha\delta} = 1 & a_{\alpha\varepsilon} = 1 \\
a_{\beta\alpha} = 1 & a_{\beta\beta} = 0 & a_{\beta\gamma} = 1 & a_{\beta\delta} = 0 & a_{\beta\varepsilon} = 0 \\
a_{\gamma\alpha} = 1 & a_{\gamma\beta} = 1 & a_{\gamma\gamma} = 0 & a_{\gamma\delta} = 1 & a_{\gamma\varepsilon} = 0 \\
a_{\delta\alpha} = 1 & a_{\delta\beta} = 0 & a_{\delta\gamma} = 1 & a_{\delta\delta} = 0 & a_{\delta\varepsilon} = 1 \\
a_{\varepsilon\alpha} = 1 & a_{\varepsilon\beta} = 0 & a_{\varepsilon\gamma} = 0 & a_{\varepsilon\delta} = 1 & a_{\varepsilon\varepsilon} = 0
\end{pmatrix} \end{array}
$$

```
import networkx as nx
G = nx.Graph()
Symbols = [α, β, γ, δ, ε] # Need unicode values for each letter - Note

for e in symbols: # Populate network
    G.add_node(e)

G.add_edge(α, β)  # alpha links
G.add_edge(α, γ)
G.add_edge(α, δ)
G.add_edge(α, ε)
```

→

Problem Set 3.3

(4) # code... continued

```
G.add_edge(β, γ) # Beta links

G.add_edge(γ, δ) # gamma links

G.add_edge(δ, ε) # delta links

A = nx.adjacency_matrix(G) # Convert network to matrix
A = A.toarray() # Convert matrix to uncompressed form
nx.draw_networkx(G)
```

b.) # See attached file

c.) Using A only, calculate degree of each node, print results
degree calculated by adding link indicators in the row.
Go row by row, loop. then go column by column, inner-loop.

$$A \quad \begin{array}{c c} & \begin{array}{ccccc} \alpha & \beta & \gamma & \delta & \epsilon \end{array} \\ \begin{array}{c} \alpha \\ \beta \\ \gamma \\ \delta \\ \epsilon \end{array} & \begin{pmatrix} - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{pmatrix} \end{array}$$

```
for e in range(len(symbols)):
    k = sum(A[e])
    print(f"Degree of node {symbols[e]} = {degree}")
```

Problem Set 3.3

(5) HR department → Survey → employees

i_1 → teammates, completed projects with

i_2 → teammates, completed projects with

i_3 → teammates, completed projects with

...

Number of teammates is employees degree, # of links

$i_1 \to i_3, i_4, i_5 \to k = 3, k_{i_1} = 3$

HR → average # teammates per worker → finds $\langle k \rangle = 4$

Sum # named teammates = 800

total teammates listed = 800

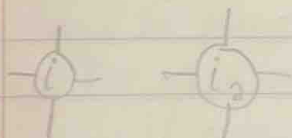
Average number of teammates listed per worker = 4

On average, each employee worked with 4 other employees on a project

How many workers in the company? what is n?

teamates also work in the company.

800 teammates are named, doesn't mean # of employees



Company network, 800 links in this network

$m = 800$

$800/4 = 200$ workers in company

$\langle k \rangle = 2m/n$, $4 = 1600/n$, 400 pairs

$p = r/tt = 800/19{,}900 \approx 0.04$

$200 \cdot 199/2 = m_{complete}$

Problem Set 3.3

(6) Network, called "D" - m = 98   98/1,225 , 196/2,450

n = 50   p = 0.08 / 8%

nodes   density, chance any 2 nodes are connected

a.) Wants to know "m", how many links in network.

50(49)/2 = 1225 max links · 0.08 = 98 links in network D

b.) Python code, create network D.

```
import networkx as nx
D = nx.Graph() # Create empty network
for e in range(1,51):
    D.add_node(e)    #adds 50 nodes to network

# Density tells us the CHANCE of any 2 nodes being connected.
# Need to add 98 links to network
num_edges = 0
for e in range(1,51):  # Iterate through nodes
    for e2 in range(e+1,51): # Inner-loop to serve as second node in network
        if num_edges <-98:  # Check edges <98
            D.add_edge(e,e2)
            num_edges = num_edges+1
        else:
            break        # exit loop if more than 98 links created

nx.density(D)
```