

# COM 530500 Network Science Homework #1

DUE: Thursday, October 14, 2021

*No late homework will be accepted.*

班級: 資應所

姓名: 鄭程哲

學號: 110065512

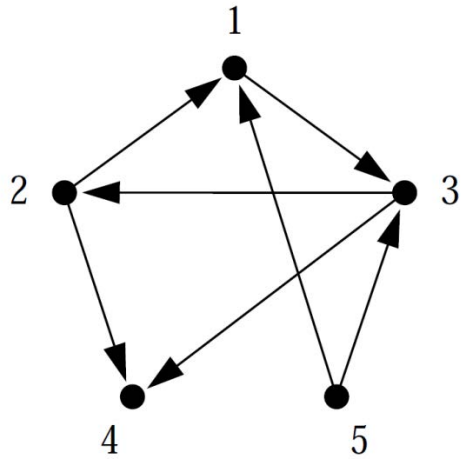


Figure 1: Network (a).

## Problem 1.(10%)

(a) (5%) Write down the adjacency matrix of network (a).

(b) (5%) Write down the cocitation matrix of network (a).

*Solution:*

$$(a) \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$(b) \text{ Cocitation matrix } C = AA^T = \begin{bmatrix} 2 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

◇

## Problem 2.(10%)

(a) (5%) Write down the incidence matrix of network (b).

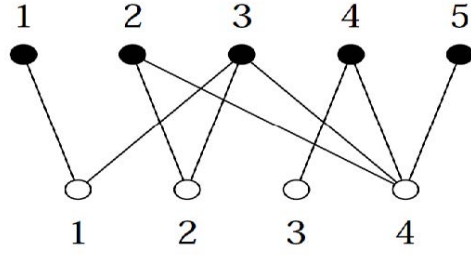


Figure 2: Network (b).

- (b) (5%) Write down the projection matrix for the projection of network (b) onto its black vertices.

*Solution:*

- (a) I regard white dots as different groups and black dots as vertexes.

$$\text{Incidence matrix} = B = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$(b) \text{ Projection matrix } P = B^T B = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 1 & 1 \\ 1 & 2 & 3 & 1 & 1 \\ 0 & 1 & 1 & 2 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

◇

**Problem 3.(10%)** Consider a bipartite network, with its two types of vertices. Suppose there are  $n_1$  vertices of type 1 and  $n_2$  vertices of type 2. Show that the mean degrees  $c_1$  and  $c_2$  of the two types are related by  $c_2 = \frac{n_1}{n_2}c_1$ .

*Solution:* If there are  $m$  edges between two types of vertices, The mean degree of type 1 vertices

$$c_1 = \frac{m}{n_1}, \text{ and } c_2 = \frac{m}{n_2}. \text{ Therefore, } \frac{c_1}{c_2} = \frac{\frac{m}{n_1}}{\frac{m}{n_2}} = \frac{n_2}{n_1}, \text{ and we get } c_2 = \frac{n_1}{n_2}c_1.$$

◇

**Problem 4.(20%)** Given

$$A = \begin{pmatrix} 0 & 2 & -1 \\ 2 & 3 & -2 \\ -1 & -2 & 0 \end{pmatrix},$$

- (a) (10%) Find all eigenvalues of matrix  $A$ .  
(b) (10%) Find an orthogonal matrix  $U$  that diagonalizes  $A$ .

*Solution:*

(a) Eigenvalues can be obtained by the equation:  $Ax = \lambda x$ , and  $(A - \lambda I)$  is a singular matrix. Therefore,  $\det(A - \lambda I) = 0$ .

$$\det(A - \lambda I) = \det \begin{pmatrix} 0 - \lambda & 2 & -1 \\ 2 & 3 - \lambda & -2 \\ -1 & -2 & 0 - \lambda \end{pmatrix} = 0$$

$$-\lambda^3 + 3\lambda^2 + 9\lambda + 5 = 0$$

$$\text{Eigenvalues}(\lambda) = -1, -1, 5$$

(b) Orthogonal matrix is composed by eigenvectors of  $A$ .

1. When  $\lambda = -1$ ,  $A - (-1)I = \begin{pmatrix} 1 & 2 & -1 \\ 2 & 4 & -2 \\ -1 & -2 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 2 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ . So a normalized eigenvector for  $\lambda_1 = -1$  is  $v_1 = (\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}})$ .
2. When  $\lambda = 5$ ,  $A - (5)I = \begin{pmatrix} -5 & 2 & -1 \\ 2 & -2 & -2 \\ -1 & -2 & -5 \end{pmatrix} \sim \begin{pmatrix} 1 & -1 & -1 \\ -5 & 2 & -1 \\ -1 & -2 & -5 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{pmatrix}$ . So a normalized eigenvector for  $\lambda_1 = -1$  is  $v_1 = (\frac{-1}{\sqrt{6}}, \frac{-2}{\sqrt{6}}, \frac{1}{\sqrt{6}})$ .

Finally, orthogonal matrix  $U = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{6}} \\ 0 & 0 & \frac{-2}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \end{pmatrix}$ . ◇

**Problem 5.(15%)** Read the tutorial from Ping-En Lu's GitHub repository to install Python3 and python-igraph (if you need).

Paste your screenshots of "Hello, World!" of both Python 3 (5%) and python-igraph (5%), and write a brief report (5%). (For example, you can write down some problems you encountered, and how you solved them.)

*Solution:* I didn't encounter any problem during the installation of both Python and Python-igraph. Additionally, I didn't meet the issue the tutorial has mentioned about installing Python-igraph on Windows. I installed it directly by typing pip command in command line.

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!")
Hello, World!
>>> exit()
```

Figure 3: Python screenshot.

◇

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from igraph import *
>>> g = Graph()
>>> print(g)
IGRAPH U--- 0 0 --
```

Figure 4: Python-iGraph screenshot.

**Problem 6.(35%)** Please download the **tvshow** dataset from Ping-En Lu’s GitHub repository, and find the following information from this dataset.

- Number of nodes. (5%)
- Number of edges. (5%)
- Mean degree. (5%)
- Maximum degree. (5%)
- Diameter. (5%)

You need to upload your **python source code** to iLMS, and **write a brief report (10%) including screenshots, README file, and descriptions of your code** below the solution area. There will be no points for this problem if you do not upload your python source code to iLMS.

*Solution:*

- Number of nodes: 3892
- Number of edges: 8631
- Mean degree: 4.4352517985611515
- Maximum degree: 67
- Diameter: 28

My code is divided into two major part. The first part is to create whole graph by the dataset. I used a loop to read each row and add vertices and edges in the graph. The second part is to output the reslut. Because iGraph is powerful and very convenience, I simply called built-in methods to get the information I need. ◇

```
IGRAPH U--- 3892 8631 --
Number of nodes: 3892
Number of edges: 8631
Mean degree: 4.4352517985611515
Maximum degree: 67
Diameter: 28
```

Figure 5: Result screenshot.