ASSIGNMENT

ON

LINEAR ALGEBRA

ES1101: COMPUTATIONAL DATA ANALYSIS

FACULTY GUIDE:

Dr. Richa Sharma
Dr. Kavita Choudhary

PREPARED BY:

Hardika Kumar (2020BTechCSE033)
Keshaw Soni (2020BTechCSE039)
Jay Agarwal (2020BTechCSE038)
Raj Burad (2020BTechEEE002)

Institute of Engineering and technology (IET)
JK LAKSHMIPAT UNIVERSITY, JAIPUR

December 2020

Introduction

Define the rank of teams/groups/individuals based on their performance in a tournament/competition/contest those participated or based on their comparison based on some assumed parameters.

Objective

To find rank of teams using dominant eigenvalues and eigenvector with power method.

Methodology

Step 1: To collect data and form a matrix A of order (n x n). We will be using power method to get dominant eigenvalue & eigenvector and hence deduce ranks of teams.

Step2: Assuming an initial guess matrix X_0 of order (n x 1)

Step3: Now follow:

$X_1=AX_0$

$X_2=AX_1$

$\vdots$

unless X of two consecutive iterations do not become equal

Step4: The eigenvalue and eigenvector received are the most dominant values. Now this dominant eigenvector can be used to deduce ranks of the teams.

---

1. Solve the following problem using manual calculation.

Suppose that there are four teams in a league match. At the end of season, the results are as follows

Team 1 beat teams 2 and 3, but lost to team 4.

Team 2 beat team 3, but lost to teams 1 and 4.

Team 3 beat team 4, but lost to teams 1 and 2.

Team 4 beat teams 1 and 2, but lost to team 3.

(a) Form the corresponding matrix A that reflects these results, where

$$a_{ij} = \begin{cases} 1, \text{ if team i beats team j} \\ 0, \textit{otherwise} \end{cases}$$

(b) How small can the dominant eigenvalue for A be? How large? Explain.
(c) Find out eigen values. Which of these is most dominant eigen value?
(d) Find out eigen vector corresponding to most dominant eigen value using Power method and find how the teams can be ranked using eigen vectors

Hardika Kumar

Name: Hardika Kumar
Roll. No. : 2020BTech CSE033

TASK 1

(a)

	Team 1	Team 2	Team 3	Team 4
Team 1	0	1	1	0
Team 2	0	0	1	0
Team 3	0	0	0	1
Team 4	1	1	0	0

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

(b) The largest possible value for dominant eigenvalue is the sum of the row which is largest.
$\therefore$ Largest possible dominant eigenvalue $= 2$

Similarly smallest possible value is the sum of the row which is smallest
$\therefore$ Smallest possible dominant eigenvalue $= 1$

(6) Characteristic Eq^n of A

$|A-\lambda I| = 0$

$$\begin{vmatrix} 0-\lambda & 1 & 1 & 0 \\ 0 & 0-\lambda & 1 & 0 \\ 0 & 0 & 0-\lambda & 1 \\ 1 & 1 & 0 & 0-\lambda \end{vmatrix} = 0$$

$\Rightarrow \quad -\lambda[-\lambda(\lambda^2) - (-1)] - [1] + [\lambda(-1)] = 0$

$\Rightarrow \quad \lambda^4 - 2\lambda - 1 = 0$

Calculating eigenvalue i.e., roots of the characteristic eq. (λ^4-2λ-1=0)

```
import numpy as np
np.linalg.eigvals(A)
```

```
array([ 1.39533699+0.j        , -0.46035519+1.13931768j,
       -0.46035519-1.13931768j, -0.47462662+0.j        ])
```

DATE :

Eigenvalues (λ) = 1.395
-0.474
-0.460 - 1.140 i
-0.460 + 1.140 i

$\Rightarrow$ Most dominant eigenvalue = 1.395

(d)

To define rank we need dominant eigenvector

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \qquad X_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$X_1 = AX_0 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 2 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ 0.5 \\ 0.5 \\ 1 \end{bmatrix}$$

$$X_2 = AX_1 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0.5 \\ 0.5 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.5 \\ 1 \\ 1.5 \end{bmatrix} = 1.5 \begin{bmatrix} 0.67 \\ 0.33 \\ 0.67 \\ 1 \end{bmatrix}$$

DATE : / /

$$X_3 = AX_2 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.67 \\ 0.33 \\ 0.67 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.67 \\ 1 \\ 1 \end{bmatrix} = 1 \begin{bmatrix} 1 \\ 0.67 \\ 1 \\ 1 \end{bmatrix}$$

$$X_4 = AX_3 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0.67 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.67 \\ 1 \\ 1 \\ 1.67 \end{bmatrix} = 1.67 \begin{bmatrix} 1 \\ 0.6 \\ 0.6 \\ 1 \end{bmatrix}$$

$$X_5 = AX_4 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0.6 \\ 0.6 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.2 \\ 0.6 \\ 1 \\ 1.6 \end{bmatrix} = 1.6 \begin{bmatrix} 0.75 \\ 0.38 \\ 0.63 \\ 1 \end{bmatrix}$$

Using python to reach final iteration

```
Iteration 48 :
[[0.882 ]
 [0.5141]
 [0.7175]
 [1.    ]]       [1.3938]

Iteration 49 :
[[0.8822]
 [0.5139]
 [0.7163]
 [1.    ]]       [1.3961]

Iteration 50 :
[[0.8812]
 [0.5131]
 [0.7163]
 [1.    ]]       [1.3961]
```

Dominant Eigenvector = $\begin{bmatrix} 0.882 \\ 0.514 \\ 0.717 \\ 1 \end{bmatrix}$

Now, higher the value in the eigenvector higher will be its rank

Rank 1 $\longrightarrow$ Team 4 (1)
Rank 2 $\longrightarrow$ Team 1 (0.882)
Rank 3 $\longrightarrow$ Team 3 (0.717)
Rank 4 $\longrightarrow$ Team 2 (0.514)

Keshaw Soni

⇒ Assignment - linear Algebra :-

Q-1

Name - Keshaw Soni
Roll No. - 2020 BTech CSE039
Batch - B

(a)

$$A = \begin{array}{c|cccc} & \text{Team-1} & \text{Team-2} & \text{Team-3} & \text{Team-4} \\ \hline \text{Team-1} & 0 & 1 & 1 & 0 \\ \text{Team-2} & 0 & 0 & 1 & 0 \\ \text{Team-3} & 0 & 0 & 0 & 1 \\ \text{Team4} & 1 & 1 & 0 & 0 \end{array}$$

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

→ Characteristic Eqn. of A ⇒ $|A - \lambda I| = 0$

$$\Rightarrow \begin{vmatrix} 0-\lambda & 1 & 1 & 0 \\ 0 & 0-\lambda & 1 & 0 \\ 0 & 0 & 0-\lambda & 1 \\ 1 & 1 & 0 & 0-\lambda \end{vmatrix} = 0$$

$$-\lambda\left(-\lambda\left((\lambda^2) - (-1)\right)\right) - (1) + \left(\lambda(-1)\right) = 0$$

$$\Rightarrow \lambda^4 - 2\lambda - 1 = 0$$

Calculating eigenvalue i.e., roots of the characteristic eq. ($\lambda^4-2\lambda-1=0$)

```
import numpy as np
np.linalg.eigvals(A)
```

```
array([ 1.39533699+0.j        , -0.46035519+1.13931768j,
       -0.46035519-1.13931768j, -0.47462662+0.j        ])
```

(b) For A, the largest EigenValue is $= 2$.

and the smallest EigenValue is $= 1$.

(c) The EigenValue using characteristic Eqn of

A is $\Rightarrow$ 1.3953, -0.4746, $-0.4604 + 1.13931i$

and $-0.4604 - 1.1393i$

(d) <u>EigenValue using Power Method</u>:-

$$X_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$\rightarrow$ <u>1st - Iteration</u>

$$X_1 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 \\ 1 \\ 1 \\ 2 \end{bmatrix} \Rightarrow 2 \begin{bmatrix} 1 \\ 0.5 \\ 0.5 \\ 1 \end{bmatrix}$$

<u>2nd iteration</u>

$$X_2 \Rightarrow \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 1 \\ 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 \\ 1 \\ 2 \\ 3 \end{bmatrix} \Rightarrow 3 \begin{bmatrix} 0.67 \\ 0.33 \\ 0.67 \\ 1 \end{bmatrix}$$

<u>3rd iteration</u>

$$X_3 \Rightarrow \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 2 \\ 3 \end{bmatrix} \Rightarrow \begin{bmatrix} 3 \\ 2 \\ 3 \\ 3 \end{bmatrix} \Rightarrow 3 \begin{bmatrix} 1 \\ 0.67 \\ 1 \\ 1 \end{bmatrix}$$

4^{th} - Iteration

$$X_4 \Rightarrow \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 3 \\ 3 \end{bmatrix} \Rightarrow \begin{bmatrix} 5 \\ 3 \\ 3 \\ 5 \end{bmatrix} \Rightarrow 5 \begin{bmatrix} 1 \\ 0.60 \\ 0.60 \\ 1 \end{bmatrix}$$

5^{th} iteration

$$X_5 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \\ 3 \\ 5 \end{bmatrix} \Rightarrow \begin{bmatrix} 6 \\ 3 \\ 5 \\ 8 \end{bmatrix} \Rightarrow 8 \begin{bmatrix} 0.75 \\ 0.375 \\ 0.625 \\ 1 \end{bmatrix}$$

Using python to reach final iteration

```
Iteration 48 :
[[0.882 ]
 [0.5141]
 [0.7175]
 [1.    ]]       [1.3938]

Iteration 49 :
[[0.8822]
 [0.5139]
 [0.7163]
 [1.    ]]       [1.3961]

Iteration 50 :
[[0.8812]
 [0.5131]
 [0.7163]
 [1.    ]]       [1.3961]
```

(d) The Eigen Vector Corresponding to Most dominant Eigen Value is $\Rightarrow$

$$\begin{bmatrix} 0.88 \\ 0.51 \\ 0.71 \\ 1 \end{bmatrix}$$

$\Rightarrow$ Rank of 1 $\Rightarrow$ 1^{st}

$\Rightarrow$ Rank of 0.88 $\Rightarrow$ 2^{nd}

$\Rightarrow$ Rank of 0.71 $\Rightarrow$ 3^{rd} & Rank of 0.51 $\Rightarrow$ 4^{th}

Jay Agarwal

PAGE NO.
DATE / /

Assignment -1.
Linear Algebra.

Name- Jay Agarwal
Roll No. - 2020 BTech CSE 038
Batch - B.

Q-1.
(a) Forming a matrix by using the given data; is

$$A = \begin{array}{c|cccc} & team1 & team2 & team3 & team4 \\ \hline team1 & 0 & 1 & 1 & 0 \\ team2 & 0 & 0 & 1 & 0 \\ team3 & 0 & 0 & 0 & 1 \\ team4 & 1 & 1 & 0 & 0 \end{array}$$

(b) for A, the largest eigen value is = 2
and smallest will be = 1.

(c) Eigen value using characteristics Equation method :-
for this

$$\det(A-\lambda I)=0 \quad \text{or}$$
$$|A-\lambda I|=0$$

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -\lambda & 1 & 1 & 0 \\ 0 & -\lambda & 1 & 0 \\ 0 & 0 & -\lambda & 1 \\ 1 & 1 & 0 & -\lambda \end{bmatrix}$$

on solving this :-

$$-\lambda(-\lambda(\lambda^2-(-1))) - (1) + (\lambda(-1)) = 0$$
$$\Rightarrow \lambda^4 - \lambda - \lambda - 1 = 0$$
$$\Rightarrow \lambda^4 - 2\lambda - 1 = 0.$$

On solving this by using python:

Calculating eigenvalue i.e., roots of the characteristic eq. (λ^4-2λ-1=0)

```
import numpy as np
np.linalg.eigvals(A)
```

```
array([ 1.39533699+0.j        , -0.46035519+1.13931768j,
       -0.46035519-1.13931768j, -0.47462662+0.j        ])
```

Values of A = 1.3953,
-0.4746,
-0.4603 + 1.1393 i
- 0.4603 - 1.1393 i

(d) Using power method, dominant eigen value of matrix A:

for this, we assume a matrix X.

$$X_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

1^{st} iteration: $AX_0 = X_1$

$$X_1 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 2 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ 0.5 \\ 0.5 \\ 1 \end{bmatrix}$$

Similarly,

$$X_2 = AX_1 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0.5 \\ 0.5 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.5 \\ 1 \\ 1.5 \end{bmatrix} = 1.5 \begin{bmatrix} 0.67 \\ 0.33 \\ 0.67 \\ 1 \end{bmatrix}$$

$$X_3 = AX_2 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.67 \\ 0.33 \\ 0.67 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.67 \\ 1 \\ 1 \end{bmatrix} = 1 \begin{bmatrix} 1 \\ 0.67 \\ 1 \\ 1 \end{bmatrix}$$

$$X_4 = AX_3 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0.67 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.67 \\ 1 \\ 1 \\ 1.67 \end{bmatrix} = 1.67 \begin{bmatrix} 1 \\ 0.60 \\ 0.60 \\ 1 \end{bmatrix}$$

$$X_5 = AX_4 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0.6 \\ 0.6 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.2 \\ 0.6 \\ 1 \\ 1.6 \end{bmatrix} = 1.6 \begin{bmatrix} 0.75 \\ 0.375 \\ 0.625 \\ 1 \end{bmatrix}$$

And for further calculation, we using python and after using this we get;

Eigen vector corresponding to the most dominant eigen value,

$$\text{is} = \begin{bmatrix} 0.88 \\ 0.51 \\ 0.71 \\ 1 \end{bmatrix}$$

Now, Ranking of team using Eigen vector (dominant):
for this largest value in Eigen vector deserves more (rank) or highest rating and so on.

hence, vector = $\begin{bmatrix} 0.88 \\ 0.51 \\ 0.71 \\ 1 \end{bmatrix}$

and 1^{st} Rank = 1
2^{nd} Rank = 0.88
3^{rd} Rank = 0.71
4^{th} Rank = 0.51

Using python to reach final iteration

```
Iteration 48 :
[[0.882 ]
 [0.5141]
 [0.7175]
 [1.    ]]       [1.3938]

Iteration 49 :
[[0.8822]
 [0.5139]
 [0.7163]
 [1.    ]]       [1.3961]

Iteration 50 :
[[0.8812]
 [0.5131]
 [0.7163]
 [1.    ]]       [1.3961]
```

Raj Burad

Raj Burad
2020 BTech CSE 002

Date
Page

① a)

	Team 1	Team 2	Team 3	Team 4
Team 1	0	1	1	0
Team 2	0	0	1	0
Team 3	0	0	0	1
Team 4	1	1	0	0

MATRIX FORMATION:

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

b) Largest possible dominant eigenvalue = 2
Smallest possible dominant eigenvalue = 1

c) Using characteristic eq for A : $|A - \lambda I| = 0$

$$\begin{vmatrix} -\lambda & 1 & 1 & 0 \\ 0 & -\lambda & 1 & 0 \\ 0 & 0 & -\lambda & 1 \\ 1 & 1 & 0 & -\lambda \end{vmatrix} = 0$$

$$-\lambda[-\lambda(\lambda^2) - (-1)] - (1) + [\lambda(-1)] = 0$$

$$\lambda^4 - 2\lambda - 1 = 0$$

$$\lambda = 1.3953 + 0j, \ -0.4603 + 1.1393i, \ -0.4603 - 1.1393i, \ -0.4746 + 0 \cdot i$$

Calculating eigenvalue i.e., roots of the characteristic eq. (λ^4-2λ-1=0)

```python
import numpy as np
np.linalg.eigvals(A)
```

```
array([ 1.39533699+0.j        , -0.46035519+1.13931768j,
       -0.46035519-1.13931768j, -0.47462662+0.j        ])
```

So eigen values are
1.3953, −0.4603 + 1.1393 i
−0.4603 − 1.1393 i, −0.4746

d) Using Power Method

1st Iteration :

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 2 \end{bmatrix} = 2\begin{bmatrix} 1 \\ 0.5 \\ 0.5 \\ 1 \end{bmatrix}$$

2nd Iteration :

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0.5 \\ 0.5 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.5 \\ 1 \\ 1.5 \end{bmatrix} = 1.5\begin{bmatrix} 0.67 \\ 0.33 \\ 0.67 \\ 1 \end{bmatrix}$$

3rd Iteration :

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0.67 \\ 0.33 \\ 0.67 \\ 1 \end{bmatrix} \cancel{=} \begin{bmatrix} \cancel{0.67} \\ \cancel{0.33} \\ \cancel{0.67} \\ \cancel{1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0.667 \\ 1 \\ 1 \end{bmatrix}$$

4th Iteration

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0.667 \\ 1 \\ 1 \end{bmatrix} = 1.66\begin{bmatrix} 1 \\ 0.6 \\ 0.6 \\ 1 \end{bmatrix}$$

5th Iteration

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0.6 \\ 0.6 \\ 1 \end{bmatrix} = \underset{\text{dominant}}{1.6}\begin{bmatrix} 0.75 \\ 0.375 \\ 0.625 \\ 1 \end{bmatrix}$$

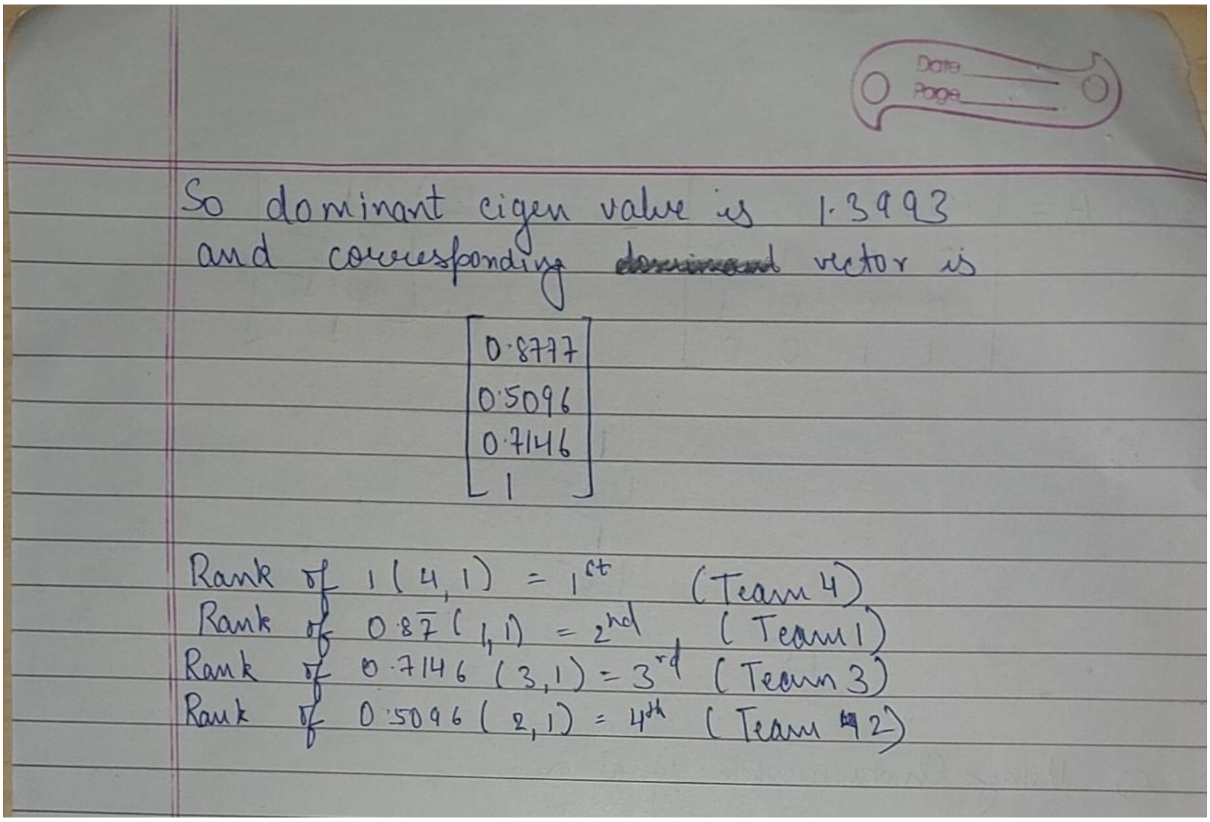
corresponding eigen vector

Using python to reach final iteration

```
Iteration 48 :
[[0.882 ]
 [0.5141]
 [0.7175]
 [1.    ]]        [1.3938]

Iteration 49 :
[[0.8822]
 [0.5139]
 [0.7163]
 [1.    ]]        [1.3961]

Iteration 50 :
[[0.8812]
 [0.5131]
 [0.7163]
 [1.    ]]        [1.3961]
```



So dominant eigen value is 1.3993
and corresponding ~~dominant~~ vector is

$$\begin{bmatrix} 0.8777 \\ 0.5096 \\ 0.7146 \\ 1 \end{bmatrix}$$

Rank of 1 (4,1) = 1st (Team 4)
Rank of 0.877 (1,1) = 2nd (Team 1)
Rank of 0.7146 (3,1) = 3rd (Team 3)
Rank of 0.5096 (2,1) = 4th (Team 2)

2. Category Selected: Entertainment

Data Collection

Top Indian Hindi Web Series and their Ratings

1. Scam 1992 ☆9.4
2. The Family Man ☆8.6
3. Special OPS ☆8.5
4. Paatal Lok ☆7.8
5. Panchayat ☆8.7
6. Sacred Games ☆8.7
7. Mirzapur ☆8.4
8. Pitchers ☆9.1
9. Asur ☆8.4
10. Kota Factory ☆9.0

(The ratings have been given based on the reviews of the public and the critics since all of these series were released on the digital platform.)

(Source: Top 30 Best Indian Hindi Web Series - IMDb)

Shows	SCAM 1992	THE FAMILY MAN	SPECIAL OPS	PAATAL LOK	PANCHAYAT	SACRED GAMES	MIRZAPUR	PITCHERS	ASUR	KOTA FACTORY
SCAM 1992	1	2	2	2	2	2	2	2	2	2
THE FAMILY MAN	0	1	2	2	0	0	2	0	2	0
SPECIAL OPS	0	0	1	2	0	0	2	0	2	0
PAATAL LOK	0	0	0	1	0	0	0	0	0	0
PANCHAYAT	0	2	2	2	1	1	2	0	2	0
SACRED GAMES	0	2	2	2	1	1	2	0	2	0
MIRZAPUR	0	0	0	2	0	0	1	0	1	0
PITCHERS	0	2	2	2	2	2	2	1	2	2
ASUR	0	0	0	2	0	0	1	0	1	0
KOTA FACTORY	0	2	2	2	2	2	2	0	2	1

Mathematical Model

A matrix is formed which reflects the following results

$$A = a_{ij} = \begin{cases} 2, & \text{if rating of } i > j \\ 1, & \text{if rating of } i = j \\ 0, & \text{otherwise} \end{cases}$$

A =

```
[[1 2 2 2 2 2 2 2 2 2]
 [0 1 2 2 0 0 2 0 2 0]
 [0 0 1 2 0 0 2 0 2 0]
 [0 0 0 1 0 0 0 0 0 0]
 [0 2 2 2 1 1 2 0 2 0]
 [0 2 2 2 1 1 2 0 2 0]
 [0 0 0 2 0 0 1 0 1 0]
 [0 2 2 2 2 2 2 1 2 2]
 [0 0 0 2 0 0 1 0 1 0]
 [0 2 2 2 2 2 2 0 2 1]]
```

X_0 =

```
[[1]
 [1]
 [1]
 [1]
 [1]
 [1]
 [1]
 [1]
 [1]
 [1]]
```

Deducing rank of the shows using power method

```
import numpy as np
X=[]
eigen=[0]
X.append(np.array([[1],[1],[1],[1],[1],[1],[1],[1],[1],[1]]))
print(A,'\n\n',X)
print("----------------------------------------")

i=0
while len(eigen)==1 or np.round(eigen[i],4)!=np.round(eigen[i-1],4):
    X.append(np.matmul(A,X[i]))
    eigen.append(max(X[i+1]))
    X[i+1]= np.round(X[i+1]/eigen[i+1],4)
    print("Iteration",i+1,":")
    print(X[i+1],"\t",eigen[i+1])
    print()
    i+=1
```

Output:

```
Iteration 1 :
[[1.    ]
 [0.4737]
 [0.3684]
 [0.0526]
 [0.6316]
 [0.6316]
 [0.2105]
 [0.8947]
 [0.2105]
 [0.7895]]        [19]

Iteration 2 :
[[1.    ]
 [0.2265]
 [0.1381]
 [0.0055]
 [0.4088]
 [0.4088]
 [0.0552]
 [0.8011]
 [0.0552]
 [0.6243]]        [9.5262]
```

```
Iteration 3 :
[[1.000e+00]
 [1.139e-01]
 [5.740e-02]
 [9.000e-04]
 [2.759e-01]
 [2.759e-01]
 [1.880e-02]
 [7.206e-01]
 [1.880e-02]
 [4.995e-01]]    [6.447]
  :
  :
  :
Iteration 22 :
[[1.000e+00]
 [2.600e-03]
 [1.000e-03]
 [0.000e+00]
 [4.060e-02]
 [4.060e-02]
 [3.000e-04]
 [3.763e-01]
 [3.000e-04]
 [1.416e-01]]    [2.2068]

Iteration 23 :
[[1.000e+00]
 [2.600e-03]
 [1.000e-03]
 [0.000e+00]
 [4.060e-02]
 [4.060e-02]
 [3.000e-04]
 [3.763e-01]
 [3.000e-04]
 [1.416e-01]]    [2.2066]

Iteration 24 :
[[1.000e+00]
 [2.600e-03]
 [1.000e-03]
 [0.000e+00]
 [4.060e-02]
 [4.060e-02]
 [3.000e-04]
 [3.763e-01]
 [3.000e-04]
 [1.416e-01]]    [2.2066]
```

Dominant Eigenvalue= [2.2066]

Corresponding Eigenvector =

```
[1.000e+00]
[2.600e-03]
[1.000e-03]
[0.000e+00]
[4.060e-02]
[4.060e-02]
[3.000e-04]
[3.763e-01]
[3.000e-04]
[1.416e-01]
```

Result and Discussion:

Rank of the shows can be deduced from the dominant eigenvector that we've calculated. Higher the value of the show in the eigenvector, higher will be its rank.

Dominant Eigenvector=

```
[2.600e-03]
[1.000e-03]
[0.000e+00]
[4.060e-02]
[4.060e-02]
[3.000e-04]
[3.763e-01]
[3.000e-04]
[1.416e-01]
```

Rank of the shows

Shows		Rank		Eigenvalue
SCAM 1992	==>	1st	==>	[1.000e+00]
PITCHERS	==>	2nd	==>	[3.767e-01]
KOTA FACTORY	==>	3rd	==>	[1.416e-01]
PANCHAYAT	==>	4th/5th	==>	[4.060e-02]
SACRED GAMES	==>	4th/5th	==>	[4.060e-02]
THE FAMILY MAN	==>	6th	==>	[2.600e-03]
SPECIAL OPS	==>	7th	==>	[1.000e-03]
MIRZAPUR	==>	8th/9th	==>	[3.000e-04]
ASUR	==>	8th/9th	==>	[3.000e-04]
PAATAL LOK	==>	10th	==>	[0.000e+00]

Conclusion:

From the above data we can conclude that by using power method we can calculate dominate eigenvalue and corresponding eigenvector.
Eigenvector can then be used to deduce rank of each team

Appendix

(Data Source: Top 30 Best Indian Hindi Web Series - IMDb)
(Python file: LA_Asignment_1 (1).ipynb - Colaboratory (google.com))

Q1

```
import numpy as np
A=np.array([[0,1,1,0],[0,0,1,0],[0,0,0,1],[1,1,0,0]])
X=[]
eigen=[0]
X.append(np.array([[1],[1],[1],[1]]))
print(A,'\n\n',X)
print("-----------------------------------------")

i=0
while len(eigen)==1 or np.round(eigen[i],2)!=np.round(eigen[i-1],2):
    X.append(np.matmul(A,X[i]))
    eigen.append(max(X[i+1]))
    X[i+1]= np.round(X[i+1]/eigen[i+1],4)
    print("Iteration",i+1,":")
    print(X[i+1],"\t",eigen[i+1])
    print()
    i+=1
```

Output:

```
[[0 1 1 0]
 [0 0 1 0]
 [0 0 0 1]
 [1 1 0 0]]

 [array([[1],
       [1],
       [1],
       [1]])]
-----------------------------------------
```

```
Iteration 1 :
[[1. ]
 [0.5]
 [0.5]
 [1. ]]          [2]

Iteration 2 :
[[0.6667]
 [0.3333]
 [0.6667]
 [1.    ]]       [1.5]

Iteration 3 :
[[1.    ]
 [0.6667]
 [1.    ]
 [1.    ]]       [1.]

Iteration 4 :
[[1. ]
 [0.6]
 [0.6]
 [1. ]]          [1.6667]

Iteration 5 :
[[0.75 ]
 [0.375]
 [0.625]
 [1.   ]]        [1.6]
 :
 :
Iteration 46 :
[[0.882 ]
 [0.5137]
 [0.7158]
 [1.    ]]      [1.397]

Iteration 47 :
[[0.8809]
 [0.5129]
 [0.7165]
 [1.    ]]      [1.3957]

Iteration 48 :
[[0.882 ]
 [0.5141]
 [0.7175]
 [1.    ]]      [1.3938]
```

```
Iteration 49 :
[[0.8822]
 [0.5139]
 [0.7163]
 [1.    ]]        [1.3961]

Iteration 50 :
[[0.8812]
 [0.5131]
 [0.7163]
 [1.    ]]        [1.3961]
```

Q2

In [5]:
```python
import pandas as pd
import numpy as np

en_show = pd.read_csv("ASN_1.csv")

df = pd.DataFrame(en_show)
df.index=range(1,11)
df
```

Output:

	Shows	SCAM1992	THE FAMILY MAN	SPECIAL OPS	PAATAL LOK	PANCHAYAT	SACRED GAMES	MIRZAPUR	PITCHERS	ASUR	KOTA FACTORY
1	SCAM 1992	1	2	2	2	2	2	2	2	2	2
2	THE FAMILY MAN	0	1	2	2	0	0	2	0	2	0
3	SPECIAL OPS	0	0	1	2	0	0	2	0	2	0
4	PAATAL LOK	0	0	0	1	0	0	0	0	0	0
5	PANCHAYAT	0	2	2	2	1	1	2	0	2	0
6	SACRED GAMES	0	2	2	2	1	1	2	0	2	0
7	MIRZAPUR	0	0	0	2	0	0	1	0	1	0
8	PITCHERS	0	2	2	2	2	2	2	1	2	2
9	ASUR	0	0	0	2	0	0	1	0	1	0
10	KOTA FACTORY	0	2	2	2	2	2	2	0	2	1

```
In [6]: A=df.iloc[:,1:].to_numpy()
        print("Matrix:")
        print(A)
```

Output:

```
Matrix:
[[1 2 2 2 2 2 2 2 2 2]
 [0 1 2 2 0 0 2 0 2 0]
 [0 0 1 2 0 0 2 0 2 0]
 [0 0 0 1 0 0 0 0 0 0]
 [0 2 2 2 1 1 2 0 2 0]
 [0 2 2 2 1 1 2 0 2 0]
 [0 0 0 2 0 0 1 0 1 0]
 [0 2 2 2 2 2 2 1 2 2]
 [0 0 0 2 0 0 1 0 1 0]
 [0 2 2 2 2 2 2 0 2 1]]
```

```
In [4]: import numpy as np
        X=[]
        eigen=[0]
        X.append(np.array([[1],[1],[1],[1],[1],[1],[1],[1],[1],[1]]))
        print(A,'\n\n',X)
        print("-----------------------------------------")

        i=0
        while len(eigen)==1 or np.round(eigen[i],4)!=np.round(eigen[i-1],4):
            X.append(np.matmul(A,X[i]))
            eigen.append(max(X[i+1]))
            X[i+1]= np.round(X[i+1]/eigen[i+1],4)
            print("Iteration",i+1,":")
            print(X[i+1],"\t",eigen[i+1])
            print()
            i+=1
```

Output:

```
[[1 2 2 2 2 2 2 2 2 2]
 [0 1 2 2 0 0 2 0 2 0]
 [0 0 1 2 0 0 2 0 2 0]
 [0 0 0 1 0 0 0 0 0 0]
 [0 2 2 2 1 1 2 0 2 0]
 [0 2 2 2 1 1 2 0 2 0]
 [0 0 0 2 0 0 1 0 1 0]
 [0 2 2 2 2 2 2 1 2 2]
 [0 0 0 2 0 0 1 0 1 0]
 [0 2 2 2 2 2 2 0 2 1]]

 [array([[1],
        [1],
        [1],
        [1],
        [1],
        [1],
        [1],
        [1],
        [1],
        [1]])]
 ----------------------------------------

Iteration 1 :
[[1.    ]
 [0.4737]
 [0.3684]
 [0.0526]
 [0.6316]
 [0.6316]
 [0.2105]
 [0.8947]
 [0.2105]
 [0.7895]]       [19]

Iteration 2 :
[[1.    ]
 [0.2265]
 [0.1381]
 [0.0055]
 [0.4088]
 [0.4088]
 [0.0552]
 [0.8011]
 [0.0552]
 [0.6243]]       [9.5262]
```

```
Iteration 3 :
[[1.000e+00]
 [1.139e-01]
 [5.740e-02]
 [9.000e-04]
 [2.759e-01]
 [2.759e-01]
 [1.880e-02]
 [7.206e-01]
 [1.880e-02]
 [4.995e-01]]     [6.447]

Iteration 4 :
[[1.000e+00]
 [6.160e-02]
 [2.710e-02]
 [2.000e-04]
 [1.957e-01]
 [1.957e-01]
 [7.900e-03]
 [6.533e-01]
 [7.900e-03]
 [4.075e-01]]     [4.9634]
Iteration 5 :
[[1.     ]
 [0.0359]
 [0.0144]
 [0.     ]
 [0.146 ]
 [0.146 ]
 [0.0039]
 [0.5981]
 [0.0039]
 [0.3402]]       [4.1138]
```

⋮

```
Iteration 21 :
[[1.000e+00]
 [2.600e-03]
 [1.000e-03]
 [0.000e+00]
 [4.060e-02]
 [4.060e-02]
 [3.000e-04]
 [3.764e-01]
 [3.000e-04]
 [1.416e-01]]    [2.2076]

Iteration 22 :
[[1.000e+00]
 [2.600e-03]
 [1.000e-03]
 [0.000e+00]
 [4.060e-02]
 [4.060e-02]
 [3.000e-04]
 [3.763e-01]
 [3.000e-04]
 [1.416e-01]]    [2.2068]

Iteration 23 :
[[1.000e+00]
 [2.600e-03]
 [1.000e-03]
 [0.000e+00]
 [4.060e-02]
 [4.060e-02]
 [3.000e-04]
 [3.763e-01]
 [3.000e-04]
 [1.416e-01]]    [2.2066]

Iteration 24 :
[[1.000e+00]
 [2.600e-03]
 [1.000e-03]
 [0.000e+00]
 [4.060e-02]
 [4.060e-02]
 [3.000e-04]
 [3.763e-01]
 [3.000e-04]
 [1.416e-01]]    [2.2066]
```