# Lab 2

#### Name: (your name)

Some remarks before you start:

- Import the Python packages below each time you start a session.
- Before submitting your report, remove all ouputs by clicking on **Edit→Clear all ouputs**. Then download your report by clicking on **File→Download→.ipynb**. This is the file you will submit on Canvas.

In this lab, you will practice:

- perform matrix operations (addition, subtraction, multiplication)
- find stationary vector of a matrix
- rank web pages using PageRank algorithm
- solve a problem in ecology using PageRank algorithm

Problems	Points
1,3	2.5
2, 4, 5, 6, 7	4
Readability of your report	3
Total: 7	Total: 28

# ✓ I. Import necessary Python packages

Execute the following code each time you start a session.

from sympy import\* # import everything from the standard Python symbolic-computing package (SymPy)
from scipy import\* # import everything from the standard Python scientific-computing package (SciPy)

## ✓ II. Matrix operations

It is more convenient to perform matrix operations with the SymPy package than with the Numpy package. As a tutorial example, let us define two matrices

 $A = \begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix} \text{ and } B = \begin{bmatrix} -2 & 0 \\ 2 & 1 \end{bmatrix}$ 

using the command Matrix (rather than array):

```
A = Matrix([[1,2],[3,2]])
B = Matrix([[-2,0],[2,1]])
```

There are a few different ways to render the matrix A + B. Try each of the following commands.

```
print(A+B)
```

pprint(A+B)

Besides numeric matrices, you can also work with symbolic matrices. Try the following:

C = Matrix([['a'],['b']])
C

#### Exercise 1

Run each of the following commands and explain what each of them does.

A - 2\*B A\*C A\*\*2 A\*\*2\*B

A\*\*2\*B\*\*2

#### **Exercise 2**

Let  $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ . Find all  $2 \times 2$  matrices B such that AB = BA. <u>Hint:</u> if you write  $B = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ , then the equation AB = BA is equivalent to a linear system of 4 equations with 4 unknowns a, b, c, d. You then write an associated matrix of this system and solve it using **rref** (review Lab 1 if you forget how to use it).

### III. Stationary vectors of a matrix

Consider matrix  $A = \begin{bmatrix} -1 & 0 \\ 2 & 3 \end{bmatrix}$ . A *stationary vector* of A is a nonzero vector v such that Av = v. To find v, you rewrite this equation as

 $(A-I_2)v=0$  where  $I_2$  is the 2 imes 2 identity matrix. Write  $v=egin{bmatrix}x\\y\end{bmatrix}$ .

The above equation becomes

 $\left( \begin{bmatrix} -1 & 0 \\ 2 & 3 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ 

which is equivalent to

$$\begin{bmatrix} -2 & 0 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

This equation is equivalent to a linear system of 2 equations and 2 unknowns x, y. Its associated matrix can be obtained simply by attaching the zero column on the right hand side to the matrix on the left hand side, as follows:

$$egin{bmatrix} -2 & 0 & 0 \ 2 & 2 & 0 \end{bmatrix}$$

A = Matrix([[-1,0],[2,3]])
B = A - eye(2) # B equals A minus the identity matrix
C = zeros(2,1) # C is a 2x1 matrix of all zeros
B = B.row\_join(C) # attach column 0 on the right of matrix B
B.rref(lambda x: abs(x) < 10\*\*-10)[0] # any number within 10^-10 of 0 to be regarded as 0</pre>

From here, you can see that x = y = 0. Therefore, matrix A has no stationary vectors.

#### **Exercise 3**

Find all stationary vectors of the matrix

	$\lceil 1/2 \rceil$	1/3	1/4]
A =	1/2	1/3	1/2
	0	1/3	1/4

Which of them is a probability vector?

#### **Exercise 4**

There is a simple numerical method to solve approximately for the probability stationary vector of an  $n \times n$  stochastic matrix A. The idea is as follows. Let  $v_0$  be an arbitrary probability vector of length n. For each positive integer k, let  $v_k = A^k v_0$ . You can see that

$$v_{k+1} = A^{k+1}v_0 = AA^kv_0 = Av_k$$

Perron-Frobenius theorem guarantees that the sequence of vectors  $v_0, v_1, v_2, v_3, \ldots$  converges to some vector v. In the limit  $k \to \infty$ , the above equation yields v = Av. Vector v is therefore the probability stationary vector of A.

With the matrix A given in Exercise 3,

- Choose any probability vector  $v_0$
- Compute  $v_k = A^k v_0$  for k = 5, 10, 20, 40
- What is the approximate probability stationary vector of A?
- Now choose a different probability vector  $v_0$  and repeat Step 2 and 3. Do you get a different probability stationary vector?

### ✓ IV. PageRank algorithm

#### Exercise 5

Consider the following internet



- Intuitively (without any calculation), how would you rank these web pages?
- Write the transition matrix (without damping) of the internet.
- Find the PageRank vector of the internet. *Recall:* it is the probability stationary vector of the transition matrix.
- Does the order of the pages based on the PageRank agree with the order that you guessed earlier?

#### Exercise 6

If the internet has disconnected clusters, the PageRank probability stationary vector may not be unique. In that case, it is ambiguous how to order the pages. Brin and Page's solution is to introduce a damping parameter  $d \in [0, 1]$  and adjust the transition matrix to

$$A=dA'+(1-d)rac{1}{n}J$$

where n is the number of pages, A' is the regular transition matrix (without damping), and J is an  $n \times n$  matrix in which all entries are equal to 1. To see the rationale behind the damping parameter, imagine that you freely navigate from one web page to another in the following manner: with probability d, you click on any link on the page that you are on (all outbound links still have the same chance of being clicked on). With probability 1 - d, you randomly jump to any page on the internet. This is a clever idea to "connect" all disconnected clusters.

With the damping parameter d=0.85, determine the PageRank vector of the following internet.



#### Exercise 7

Surprisingly, the PageRank algorithm used by Google to organize the web pages from a Google search result can be used to model food webs. Consider a mini food web in the below figure.



Although *Grass* is not an animal, it is still a node of the food web. The arrow  $A \rightarrow B$  indicates that A feeds B. For example, *Lion* feeds *Grass* by its manure. Unlike in the PageRank model, the "authority" of a node was not determined by incoming links to that node, but rather by outgoing links. The more other species that a given species supports with the nutrients passing though it, the more important it is to the ecosystem. Regarding the "quality" of links, unlike in the PageRank model where all outbound links at a given node are regarded as equal, animals typically prefer certain food/preys over others. Assume the following:

- Fox's preference is 1/3 for Cricket, 2/3 for Vole.
- *Grass*'s preference is 1/5 for *Fox*, 1/3 for *Lion*, 1/3 for *Warthog*, and 1/30 for each other species. Think of it this way: *Lion* and *Warthog* produce the most manure, so *Grass* prefers them more.
- Lion's preference is 1/6 for Fox, 5/6 for Warthog.
- Owl and Snake have no preference among their preys.
- Warthog's preference is 1/6 for Cricket, 5/6 for Grass.

Find the PageRank vector to determine the importance level of each species in the ecosystem.