THE UNIVERSITY OF CHICAGO

Master of Science in Analytics

MSCA 37016 – Advanced Linear Algebra for Machine Learning

Instructions:

- Mark the question number and your final answer clearly (use a textbox.)
- Remember to show and explain your work (*If you can't explain it, you don't understand it.*)
- Please submit your solution through Canvas.

(15 points) Question 1

Given

$$\boldsymbol{A} = \begin{bmatrix} 4 & 0 & -2 \\ 2 & 5 & 4 \\ 0 & 0 & 5 \end{bmatrix}$$

1. *4*% - Calculate the characteristic equation of *A*
2. *4*% - Calculate the eigenvalues of *A*.
3. *4*% - Verify your answer:
    a. Using python
    b. By calculating $\sum_{i=1}^{2} \lambda_i$; $\prod_{i=1}^{2} \lambda_i$
4. *(*3%*)* Is matrix ***A*** Positive Definite? Why?

The University of Chicago, MSc Analytics Program
455 N Cityfront Plaza, Suite 950
Chicago, IL 60611
analytics@uchicago.edu

1. $Ax = \lambda x \qquad (A - \lambda I)x = 0$

$|A - \lambda I| = 0$

$$\begin{vmatrix} 4-\lambda & 0 & -2 \\ 2 & 5-\lambda & 4 \\ 0 & 0 & 5-\lambda \end{vmatrix} = (4-\lambda)(5-\lambda)^2 = 0$$

$(4-\lambda)(25 - 10\lambda + \lambda^2) = 100 - 40\lambda + 4\lambda^2 - 25\lambda + 10\lambda^2 - \lambda^3$

$= \boxed{-\lambda^3 + 14\lambda^2 - 65\lambda + 100 = 0}$

2. $(4-\lambda)(5-\lambda)(5-\lambda) = 0$

$\boxed{\lambda_1 = 4 \quad \lambda_2 = 5 \text{ (repeated eigenvalue)}}$

3. (b) The repeated eigenvalue is counted twice.

$\sum_i^2 \lambda_i = 4+5+5 = 14 = tr(A) = 4+5+5 = 14$

$\prod_i^2 \lambda_i = (4)(5)(5) = 100 = |A| = 4(25) = 100$

(c) **NO.** Positive definite matrix is a symmetric matrix whose $\lambda_i > 0$. Matrix A is not symmetric.

In [1]:
```
from numpy import linalg as la
import numpy as np
```

In [2]:
```
A = np.array([[4,0,-2],[2,5,4],[0,0,5]])
EgiVal, EigVect = la.eig(A)
```

In [3]:
```
print(la.eig(A))
```

```
(array([5., 4., 5.]), array([[ 0.        ,  0.4472136 , -0.89442719],
       [ 1.        , -0.89442719,  0.        ],
       [ 0.        ,  0.        ,  0.4472136 ]]))
```

In [4]:
```
coeff = [-1,14,-65,100]
print(np.roots(coeff))
```

```
[5.00000025 4.99999975 4.        ]
```

In [ ]:
```

```