classmate

Date
Page

SC HW-3
Vibhor Agarwal
2020349

Q1 **Given:** $A \in R^{n \times n}$, A is triangular.
To Prove: It's eigenvalues are its diagonal elements.

Now A being triangular matrix. will always be non singular.
We know for triangular Matrices $\det(A)$ = product of diagonal entries.

ie $\begin{bmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$, for all matrices of such kind & any dimensions $n \times n$, $n \in N$. have always have

$[\text{determinant}]_{A_{n \times n}} = \prod a_{ii}$, $1 \leq i \leq n$.

Now, Let λ be an eigenvalue.
ie, $A - \lambda I$ will also be triangular. as λI is diagonal matrix so non diagonal entries remain untouched.

So, $|A - \lambda I| = 0$ for eigenvalue.

ie. $\prod_{i=1}^{n} |a_{ii} - \lambda| = 0$

But this means $\lambda = a_{ii}$ for atleast one $i \in [1, n]$

So this means that eigenvalues in case of triangular matrix is its diagonal entries.

Q3 Given:- $u, v \in \mathbb{R}^n$, $u^T v = 1$
$A = uv^T$

To Find: Eigen values of A.

Now we know $Rank(AB) = \min(Rank A), Rank(B))$
since u, v^T are row & column vectors
$Rank(u) = Rank(v^T) = 1$

ie $Rank(A) = Rank(uv^T) = 1$

Hence A is not full rank.
But this means we will have ~~Rank~~ 0 as eigenvalues of A. Here A is always diagonalisable if $uv^T \neq 0$.

Now from previous Q_2 result,
$Rank(uv^T)$ = No. of non zero eigen values.

$\Rightarrow$ No. of non zero eigenvalues = 1.

Hence zero eigen values = $n-1$. [As A must have n eigenvalues]

So multiplicity of ~~0~~ $0 = n-1$.

classmate
Date
Page

Now for other nonzero eigenvalue, we propose $\lambda = v^T u$ is the req. eigenvalue.
We now verify this, by taking u as eigenvector

$$Au = uv^T u$$

& ~~$\lambda u = v$~~

$$u\lambda = uv^T u$$

So, $Au = u\lambda = uv^T u$.
This indeed verifies $v^T u$ is eigenvalue of $A = uv^T$

So the eigenvalues of A are $0, v^T u$.

Now for power iterations.

$$u_k = \lambda_1^k \left[c_1 v_1 + c_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k v_2 + \cdots + c_n \left(\frac{\lambda_n}{\lambda_1}\right)^k v_n \right]$$

where (λ_1, v_1) are (eigenvalue, eigenvector) pair.
& u_k denotes estimated eigen vector in k^{th} iteration.
$c_1, c_2 - c_n \in R$.

Now $\lambda_1 = 1$ & $\lambda_2, \lambda_3 - - \lambda_n = 0$. so,

$$u_k = 1^k [c_1 v_1 + 0 + \cdots 0]$$

$u_k = c_1 v_1$ which is independent of k.
So it will converge in just a single initial iteration & won't converge any further.

Q4 **To show:** Eigenvalues of real symmetric +ve definite are real & >0.

So, let's assume $Ax = \lambda x$ & $x \neq 0$. Then, $A = A^T$.

$$\lambda \bar{x}^T x = \bar{x}^T (\lambda x)$$

[$\bar{x}$ denotes conjugate & λ is a scalar so can be changed position]

$$\Rightarrow \quad = \bar{x}^T (Ax)$$

$$= (\bar{x}^T A) x$$

$$= (A^T \bar{x})^T x$$

$$= (A\bar{x})^T x$$

$$= (\bar{A}\bar{x})^T x$$

[As A is real, $A = \bar{A}$]

$$= (\bar{\lambda}\bar{x})^T x$$

$$\lambda \bar{x}^T x = \bar{\lambda} \bar{x}^T x \quad \Rightarrow \bar{x}^T(x)[\lambda - \bar{\lambda}] = 0$$

Now since $\bar{x}^T x \neq 0$ as $x \neq 0$ so, $\lambda = \bar{\lambda}$.

But this means λ is also real.

Now we need to show $\lambda > 0$

Let A be a symmetric real +ve definite matrix

ie, $x^T A x > 0 \quad \forall x \in \mathbb{R}^n$. [def^n of +ve definite]

Now let x be ~~an~~ a corresponding eigenvector to λ eigenvalue. & $x \neq 0$.

ie

$$Ax = \lambda x$$

$$\Rightarrow \quad x^T A x = x^T \lambda x$$

$$\Rightarrow \quad \lambda = \frac{x^T A x}{\|x\|^2}$$

Now $x^T A x > 0$ & $\|x\|^2 > 0$

So, $\lambda > 0$

Hence $\lambda > 0$ and is real which completes our proof.

Date
Page

Q5 Given:- $A \in R^{n \times n}$ is symetric & +ve definite.
ie. eigenvalues are real & +ve.

$$e^A = \sum_{K=0}^{\infty} \frac{1}{K!} A^K$$

Since A is symmetric so A is also diogonalisable.

So $A = QDQ^{-1}$ where Q is a orthogonal matrix with n independent vectors. &

$$D = \begin{bmatrix} \lambda_1 & 0 & 0 & - & \cdot \\ 0 & \lambda_2 & 0 & - & \cdot \\ \vdots & & & & \\ 0 & 0 & - & - & \lambda_n \end{bmatrix}$$

Now

$$e^D = \begin{bmatrix} \sum_{K=0}^{\infty} \frac{\lambda_1^K}{K!} & 0 & 0 & - & - \\ 0 & \sum_{K=0}^{\infty} \lambda_2^K / K! & & & \\ 0 & - & - & - & \sum_{K=0}^{\infty} \frac{\lambda_n^K}{K!} \end{bmatrix}$$

$$e^D = \begin{bmatrix} e^{\lambda_1} & 0 & 0 & - & - \\ 0 & e^{\lambda_2} & 0 & - & \\ \vdots & & & & e^{\lambda_n} \\ 0 & \cdot & & & \end{bmatrix} \quad \left(\text{as } \sum_{K=0}^{\infty} \frac{\lambda^K}{K!} = e^K\right)$$

Date
Page

$$A = QDQ^{-1}$$
$$\Rightarrow Q^{-1}AQ = D$$

$$e^{D} = \sum_{k=0}^{\infty} \frac{(Q^{-1}AQ)^{k}}{k!}$$

$$\Rightarrow A^2 = QDQ^{-1}QDQ^{-1}$$
$$A^2 = QD^2Q^{-1}$$
$$A^3 = QD^2Q^{-1}QDQ^{-1}$$
$$A^3 = QD^3Q^{-1}$$
$$\vdots$$
$$A^K = QD^KQ^{-1}$$
$$\Rightarrow D^K = Q^{-1}A^KQ$$

$$\Rightarrow (Q^{-1}AQ)^K = Q^{-1}A^KQ$$

$$e^{D} = \sum_{K=0}^{\infty} \frac{Q^{-1}A^KQ}{K!}$$

$$e^{D} = Q^{-1} \sum_{K=0}^{\infty} \frac{A^K}{K!} Q.$$

$$\Rightarrow e^{D} = Q^{-1}e^{A}Q.$$

$$\Rightarrow e^{A} = Qe^{D}Q^{-1}$$

$$e^{A} = Q \begin{bmatrix} e^{\lambda_1} & 0 & - & - & 0 \\ 0 & e^{\lambda_2} & 0 & - & - \\ | & & & & \\ 0 & 0 & - & - & e^{\lambda_n} \end{bmatrix} Q^{-1}$$

& $Q = [v_1 \ v_2 \ \cdots \ v_n]$ where v_i is eigen vector to λ_i eigenvalue. of $\underline{A}$

So

$$e^A = [v_1 \ v_2 \ \cdots \ v_n] \begin{bmatrix} e^{\lambda_1} & 0 & - & - & 0 \\ 0 & e^{\lambda_2} & 0 & - & - \\ | & & & & \\ 0 & - & - & & e^{\lambda_n} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ | \\ v_n \end{bmatrix}$$

(as $Q^{-1} = Q^T$ for orthogonal)

Q2 Given:- $A \in \mathbb{R}^{n \times n}$, A is non defective.
To Prove: Rank(A) = No. of non zero eigenvalues of A.

Since A is diagonalisable. [A is non defective iff diagonalisable]
$A = Q \Lambda Q^{-1}$ where Q is invertible. as A is non defective so all eigenvectors are linearly independent
we claim:
Now Rank (QA) = Rank (A) when Q is invertible

Now $A \in \mathbb{R}^{n \times n}$ & $Q \in \mathbb{R}^{n \times n}$.

So, $QAx = 0$ iff $Ax = 0$ ie QA & A has the same kernel.

So rank(QA) = n - Ker(QA) [By Rank Nullity]
rank(A) = n - Ker(A) = n - Ker(QA)

So, rank(QA) = rank(A)

So.

$\text{rank}(AQ) = \text{rank}((AQ)^T)$ [Rank(A) = rank(A^T)]
$= \text{rank}(Q^T A^T)$
$= \text{rank}(A^T)$ [as Q^T is invertible & rank(QA) = rank(A)]
$= \text{rank}(A)$.

So rank(QA) = rank(AQ) = rank(A).

Now,

rank (QA) = rank (A)

rank (AQ) = rank (A)

rank $(Q^{-1}AQ)$ = rank (A) [Multiplying by Q^{-1} which is invertible as Q is invertible]

rank (A) = rank (D).

But we know rank of a diagonal matrix is no. of non zero entries or eigenvalues.

So, rank(A) = No. of non zero eigenvalues of A.

Vibhor Agarwal

SC Assignment 3

2020349

Problem 6

The results that we obtain in this problem are as follows

```
Normalised Power Iteration :
The found largest magnitude eigenvalue is  11.0
The corresponding eigenvector is  [0.5  1.   0.75]
Inverse Power Iteration :
The found smallest magnitude eigenvalue is  2.0000000000000004
The corresponding eigenvector is  [-0.2 -0.4  1. ]
Numpy eigenvalues and eigenvectors :
The eigenvalues are  [11. -2. -3.]
The corresponding eigenvector columns are
 [[ 3.71390676e-01  1.82574186e-01 -5.26283806e-16]
 [ 7.42781353e-01  3.65148372e-01 -5.54700196e-01]
 [ 5.57086015e-01 -9.12870929e-01  8.32050294e-01]]
```

- We see that the actual eigenvalues found by numpy function is [11 -2 -3]
- The Normalised Power Iteration as expected can find the largest magnitude eigenvalue 11.
- The corresponding eigen vector found is [0.5 1 0.75] which is very close to the actual eigenvector found by numpy ie. [0.371 0.742 0.557]
- Inverse Power Iteration can also find the eigenvalue of the smallest magnitude ie 2 However its not able to find the correct sign because we are using vector infinite norm for getting the magnitude which is always positive but the found magnitude is correct.
- The eigenvector found by inverse iteration corresponding to eigenvalue -2 is [-0.2 -0.4 1] which is close to actual eigenvector of [0.182 0.36 -0.91]. Note that we can always multiply eigenvector by a scalar and so on multiplying it by -1 gives us [0.2 0.4 -1] which is fairly accurate as compared to eigenvector found by Numpy.

Problem 7

```
Shifted Inverse Iteration :
The found eigenvalue is  2.133074475348525
The corresponding eigenvector is  [-0.60692002  1.          0.34691451]
Numpy eigenvalues and eigenvectors :
The eigenvalues are  [0.57893339 2.13307448 7.28799214]
The corresponding eigenvector columns are
 [[-0.0431682  -0.49742503 -0.86643225]
 [-0.35073145  0.8195891  -0.45305757]
 [ 0.9354806   0.28432735 -0.20984279]]
```

- The actual eigenvalues found by numpy function is [7.28 2.13 0.57]
- We see that shifted inverse iteration can find the eigenvalue nearest to 2 ie. 2.133
- The eigenvector found by Shifted Inverse Iteration is [-0.606 1 0.3469] and that found by numpy corresponding to 2.133 is [-0.49 0.81 0.28] which is fairly close to that by shifted inverse iteration.

Problem 8

The results that we obtain are as follows

```
Rayleigh Quotient Iteration :
The found eigenvalue is : 11.000000000000018
The corresponding eigenvector is : [0.5  1.   0.75]
The convergence rate is :  0.2340098684432489
Numpy eigenvalues and eigenvectors :
The eigenvalues are  [11. -2. -3.]
The corresponding eigenvector columns are
 [[ 3.71390676e-01  1.82574186e-01 -5.26283806e-16]
 [ 7.42781353e-01  3.65148372e-01 -5.54700196e-01]
 [ 5.57086015e-01 -9.12870929e-01  8.32050294e-01]]
```

- The eigenvalue that we obtain using Rayleigh Quotient Iteration for matrix in Q6 is 11. We take a random matrix here. We observe from the result of Numpy Eigenvalue that 11 is indeed an eigenvalue of the matrix.
- The corresponding eigenvector found by Rayleigh Quotient Iteration is [0.5 1 0.75] which is close to the corresponding eigenvector of 11 found by Numpy as [0.37 0.74 0.557]
- The largest magnitude eigenvalue is found by Numpy is 11 and we calculate the convergence rate using 11 and the approximate convergence rate comes out to be 0.23. For subsequent iterations this convergence rate comes close to 2 and 6.26.

Problem 9

The results that we obtain are as follows

```
For matrix in Q6 :
Eigen values using QR Iteration :  [11. -3. -2.]
Eigen values using numpy :  [11. -2. -3.]
For matrix in Q7 :
Eigen values using QR Iteration :  [7.28799214 2.13307448 0.57893339]
Eigen values using numpy :  [7.28799214 2.13307448 0.57893339]
```

As we observe for Matrix is problem 6 the eigenvalues are [11,-2,-3] and that obtained by our QR Iteration are also [11,-3,-2] which is exactly the same.

As we observe for Matrix is problem 7 the eigenvalues are [7.28,2.13,0.57] and that obtained by our QR Iteration are also [7.28,2.133,0.578] which is the same and hence verifies our algorithm.