EMTH 211 Linear Algebra
Assignment

Danny Choo
student ID - 28842156
kxc11@uclive.ac.nz

Grace Dain Lee
student ID - 51455525
dil15@uclive.ac.nz

19th September 2016

Q1.a

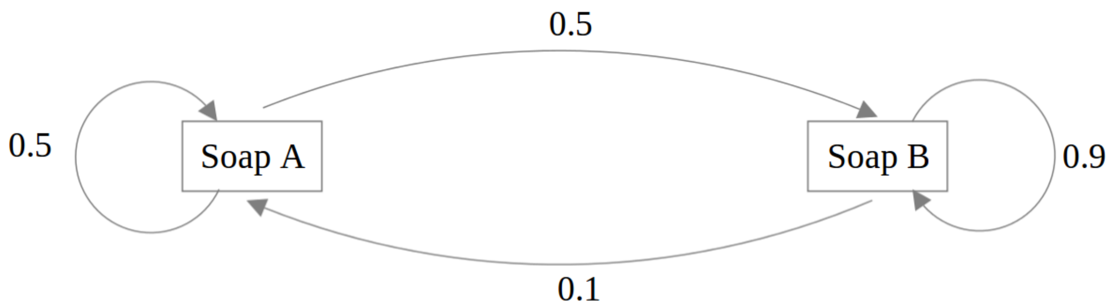


Figure 1: A directed graph of the senario.

Assume that, Soap A and B have 100 customers each at first

$$A_k\prime = 0.5A_k + 0.1B_k$$
$$B_k\prime = 0.5A_k + 0.9B_k$$

where A_k = customers of soap A for this month and B_k = customers of soap B for this month, and $A_k\prime$ = customers of soap A for next month and $B_k\prime$ = customers of soap B for next month.

Q1.b

$$\begin{bmatrix} A_k\prime \\ B_k\prime \end{bmatrix} = \begin{bmatrix} 0.5 & 0.1 \\ 0.5 & 0.9 \end{bmatrix} \begin{bmatrix} A_k \\ B_k \end{bmatrix} \text{(from 1.1)}$$

where

$$\begin{bmatrix} 0.5 & 0.1 \\ 0.5 & 0.9 \end{bmatrix}$$

is the transition matrix.

Q1.c

In one month's time

$$\Rightarrow \begin{bmatrix} A_k\prime \\ B_k\prime \end{bmatrix} = \begin{bmatrix} 0.5 & 0.1 \\ 0.5 & 0.9 \end{bmatrix} \begin{bmatrix} 100 \\ 100 \end{bmatrix}$$
$$= \begin{bmatrix} 50+10 \\ 50+90 \end{bmatrix} = \begin{bmatrix} 60 \\ 140 \end{bmatrix}$$

Q1.d

In two month's time

$$\Rightarrow \begin{bmatrix} A_k\prime\prime \\ B_k\prime\prime \end{bmatrix} = \begin{bmatrix} 0.5 & 0.1 \\ 0.5 & 0.9 \end{bmatrix} \begin{bmatrix} A_k\prime \\ B_k\prime \end{bmatrix}$$
$$= \begin{bmatrix} 0.5 & 0.1 \\ 0.5 & 0.9 \end{bmatrix} \begin{bmatrix} 60 \\ 140 \end{bmatrix}$$
$$= \begin{bmatrix} 30+14 \\ 30+126 \end{bmatrix} = \begin{bmatrix} 44 \\ 156 \end{bmatrix}$$

where $A_k\prime\prime$ is the number of customers in two months time for soap A, and $B_k\prime\prime$ is the number of customers in two months time for soap B.

Q1.e

In one year's time (12 months), the equation is:

$$\begin{bmatrix} A_k\prime\prime\prime \\ B_k\prime\prime\prime \end{bmatrix} = \begin{bmatrix} 0.5 & 0.1 \\ 0.5 & 0.9 \end{bmatrix}^{12} \begin{bmatrix} 100 \\ 100 \end{bmatrix}$$
$$\vec{v}_1 = A^{12}\vec{v}_0.$$

This will be solved by finding eigenvectors and eigenvalues of the problem.

To find the eignevalues:

$$det(A - \lambda I) = 0$$
$$det \begin{bmatrix} 0.5-\lambda & 0.1 \\ 0.5 & 0.9-\lambda \end{bmatrix} = 0$$
$$\Rightarrow (0.5-\lambda)(0.9-\lambda) - 0.05 = 0$$
$$\Rightarrow \lambda^2 - 1.4\lambda + 0.4 = 0$$
$$\Rightarrow 5\lambda^2 - 7\lambda + 2 = 0$$
$$\Rightarrow (5\lambda - 2)(\lambda - 1) = 0$$

Hence, $\lambda_1 = 0.4$ and $\lambda_2 = 1$

To find the eigenvectors at $\lambda_1 = 0.4$,

$$(A - \lambda_1 I)\vec{x}_1 = \vec{0}$$

$$\left[\begin{array}{cc|c} 0.1 & 0.1 & 0 \\ 0.5 & 0.5 & 0 \end{array}\right] R_2 -> R_2 - 5R_1 \quad \sim \quad \left[\begin{array}{cc|c} 0.1 & 0.1 & 0 \\ 0 & 0 & 0 \end{array}\right]$$

So, $x_1 = -x_2$. Thus, $\vec{x}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

To find the eigenvectors At $\lambda_2 = 1$,

$$(A - \lambda_2 I)\vec{x}_2 = \vec{0}$$

$$\left[\begin{array}{cc|c} -0.5 & 0.1 & 0 \\ 0.5 & -0.1 & 0 \end{array}\right] R_2 -> R_2 + R_1 \qquad \sim \qquad \left[\begin{array}{cc|c} -0.5 & 0.1 & 0 \\ 0 & 0 & 0 \end{array}\right]$$

So, $x_2 = 5x_1$. Thus, $x_2 = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$

We know that $\vec{v}_1 = A^{12}\vec{v}_0$, and $A = PDP^{-1}$. Hence, finding P, D, P^{-1} :

$$D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} = \begin{bmatrix} 0.4 & 0 \\ 0 & 1 \end{bmatrix}, P = \begin{bmatrix} 1 & 1 \\ -1 & 5 \end{bmatrix}, P^{-1} = \begin{bmatrix} \frac{5}{6} & -\frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} \end{bmatrix}$$

So,

$$\vec{v}_1 = A^{12}\vec{v}_0$$

$$\begin{bmatrix} A_k''' \\ B_k''' \end{bmatrix} = \begin{bmatrix} 0.5 & 0.1 \\ 0.5 & 0.9 \end{bmatrix}^{12} \begin{bmatrix} 100 \\ 100 \end{bmatrix}$$

Thus,

$$\Rightarrow \vec{v}_1 = PD^{12}P^{-1}\vec{v}_0 = \begin{bmatrix} 1 & 1 \\ -1 & 5 \end{bmatrix} \begin{bmatrix} 0.4^{12} & 0 \\ 0 & 1^{12} \end{bmatrix} \begin{bmatrix} \frac{5}{6} & -\frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} 100 \\ 100 \end{bmatrix}$$

The calculation was implemted via matlab. Matlab shows

```
>> Z = P*(D^12)*P_Inv*v

Z =

   33.3345
  166.6655

>> Z

Z =

   33.3345
  166.6655

>> Z(1) + Z(2)

ans =

   200
```

which is correct.

In the long run:

$$
\begin{aligned}
\vec{v}_k &= A^k \vec{v}_0 \\
&= PD^kP^{-1}\vec{v}_0 \\
&= \begin{bmatrix} 1 & 1 \\ -1 & 5 \end{bmatrix} \begin{bmatrix} 0.4^k & 0 \\ 0 & 1^k \end{bmatrix} \begin{bmatrix} \frac{5}{6} & -\frac{1}{6} \\ \frac{1}{6} & -\frac{1}{6} \end{bmatrix} \begin{bmatrix} 100 \\ 100 \end{bmatrix}
\end{aligned}
$$

As $k \to \infty$,

$$
D^{\infty} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}
$$

$$
\begin{aligned}
\vec{v}_k &= \begin{bmatrix} 1 & 1 \\ -1 & 5 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{5}{6} & -\frac{1}{6} \\ \frac{1}{6} & -\frac{1}{6} \end{bmatrix} \begin{bmatrix} 100 \\ 100 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 1 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} \frac{5}{6} & -\frac{1}{6} \\ \frac{1}{6} & -\frac{1}{6} \end{bmatrix} \begin{bmatrix} 100 \\ 100 \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{6} & \frac{1}{6} \\ \frac{5}{6} & \frac{5}{6} \end{bmatrix} \begin{bmatrix} 100 \\ 100 \end{bmatrix} \\
&= \begin{bmatrix} \frac{200}{6} \\ \frac{1000}{6} \end{bmatrix}
\end{aligned}
$$

Since,

$$
\Rightarrow \frac{200}{6} + \frac{1000}{6} = \frac{1200}{6} = 200
$$

The solution is correct.

Q1.f

As worked out in Q1e, $Ax = \lambda x$, then

$$
\begin{aligned}
&\Rightarrow Ax - \lambda x \\
&\Rightarrow (A - \lambda I) = 0
\end{aligned}
$$

when $\lambda = 1$,

$$
\left[\begin{array}{cc|c} -0.5 & 0.1 & 0 \\ 0.5 & -0.1 & 0 \end{array}\right] R_2 -> R_2 + R_1 \quad \sim \quad \left[\begin{array}{cc|c} -0.5 & 0.1 & 0 \\ 0 & 0 & 0 \end{array}\right] \quad x_1 = -x_2
$$

As done in Q 1.e, the eigenvector of $\lambda = 1$ is $x = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$.

Q2.a

(i)

The function is written as below

```
function [final_eig_val, final_eig_vect] = power_method(A, tol)
%
% This function performs the Power Method until two successive iterations
% have a difference of less than the tolerance
%
% Function declaration: [final_eig_val, final_eig_vect] = power_method(A, tol)
%
% Inputs:
% A = n x n matrix
% tol = tolerance value
%
% Outputs:
% final_eig_val = eigen-value
% final_eig_vect = corresponding eigen-vector
%
% Authors: Danny, Grace


%% Obtain starting matrix (y)
row_col = size(A); % Returns 1 x 2 matrix with dimensions of A
n = row_col(1);
y = ones(n,1) % Create n x 1 starting matrix

y_scaled = norm(y,2);
y = y./y_scaled;
%% Find 0th eigen-value
% Hence, 0th eigen-value (eig_val) and eigen-vector (eig_vect) obtained


first_eigen_value = dot((A * y), y) / dot(y, y);

y = A*y;
y_scaled = norm(y,2);
y = y./y_scaled;

second_eigen_value = dot((A * y), y) / dot(y, y);

%% Initialise diff to 0 (to be used to compare with tol)
diff = abs(first_eigen_value - second_eigen_value);

%% Iterate
while diff >= tol

    first_eigen_value = second_eigen_value;
    y = A*y;
    y_scaled = norm(y,2);
    y = y./y_scaled;
    second_eigen_value = dot((A * y), y) / dot(y, y);
    % Find difference
    diff = abs(first_eigen_value - second_eigen_value);

end

final_eig_val = second_eigen_value;
final_eig_vect = y;

end
```

The approximated eigenvalue and the corresponding eigenvalue was tested with the test file below:

```
clear all
clc

A = [-3.9 0.1 0.5 0.6; 0.1 7.2 0.1 -0.5; 0.5 0.1 1.1 0.3;0.6 -0.5 0.3 -10];

tol = 0.001;

[final_eig_val, final_eig_vect] = power_method(A, tol)
```

After running the above program, the output is shown as:

```
y =

     1
     1
     1
     1


final_eig_val =

  -10.0782


final_eig_vect =

    0.0953
   -0.0222
    0.0228
   -0.9949
```

as expected.

(ii)

Similarly, the Inverse power method function is written as below

```
function [eig_val_now,x, i]=inverse_method(A,tol)
%Power method for computing eigenvalues
%% Obtain starting matrix (y)
%row_col = size(A); % Returns 1 x 2 matrix with dimensions of A
%n = row_col(1);
%x = ones(n,1); % Create n x 1 starting matrix

%% Find 0th eigen-value
%eig_val_now = norm(A*y, Inf); % Infinity norm of (A*y)
%eig_vect_now = y;
% Hence, 0th eigen-value (eig_val) and eigen-vector (eig_vect) obtained

row_col = size(A);
eig_val_now = 0;
shift = 0;

N = row_col(1);
I = eye(N);
x = ones(N,1);
x(1) = 0;
x(N) = 0;
conv = 10000;
eig_val_before= 0;
i = 0;

 % compute shifted inverse matrix
B = inv(A - shift*I);

while (conv > tol)
   i = i + 1;
   y = B*x;
   x = y / norm(y,2);
   eig_val_now = x'*A*x;
   conv = abs(eig_val_now-eig_val_before) / abs(eig_val_now);
   eig_val_before = eig_val_now;
end
```

The approximated eigenvalue and the corresponding eigenvalue was tested with the test file below:

```

clear all
clc

A = [-3.9 0.1 0.5 0.6; 0.1 7.2 0.1 -0.5; 0.5 0.1 1.1 0.3;0.6 -0.5 0.3 -10];

tol = 0.001;
```

```
evalue = -3.9;
evalue2 = 1.1;
evalue3 = 7.2;

%[final_eig_val , final_eig_vect] = power_method_infinity_norm (A, tol)
%gershdisc(A)


A_shift =  A - evalue* eye(size(A));
[lambda2,x, i]=inverse_method(A_shift,tol);
lambda2 = lambda2 + evalue;

A_shift2 = A - evalue2*eye(size(A));
[lambda3,x, i]=inverse_method(A_shift2,tol);
lambda3 = lambda3 + evalue2;

A_shift3 = A - evalue3*eye(size(A));
[lambda4,x, i]=inverse_method(A_shift3,tol);
lambda4 = lambda4 + evalue3;
%return
lambda2
lambda3
lambda4
```

After running the above program, the output is shown as:

```

lambda2 =

   -3.8971


lambda3 =

    1.1596


lambda4 =

    7.2166
```

as expected.

Q2.b

(i)

$$A = \begin{bmatrix} -3.9 & 0.1 & 0.5 & 0.6 \\ 0.1 & 7.2 & 0.1 & -0.5 \\ 0.5 & 0.1 & 1.1 & 0.3 \\ 0.6 & -0.5 & 0.3 & -10 \end{bmatrix}$$

Using the power method function in Q2a, the dominant eigenvalue is -10.0782, and the corresponding eigenvector is

$$\begin{bmatrix} 0.0953 \\ -0.0222 \\ 0.0228 \\ -0.9949 \end{bmatrix}$$

(ii)

The Gerschgorin Disks Diagram is attached on a separate page.
Using the inverse power method matlab function written in Q2a, the approximations were found to be

```

lambda2 =

   -3.8971

```

```

lambda3 =

    1.1596


lambda4 =

    7.2166
```

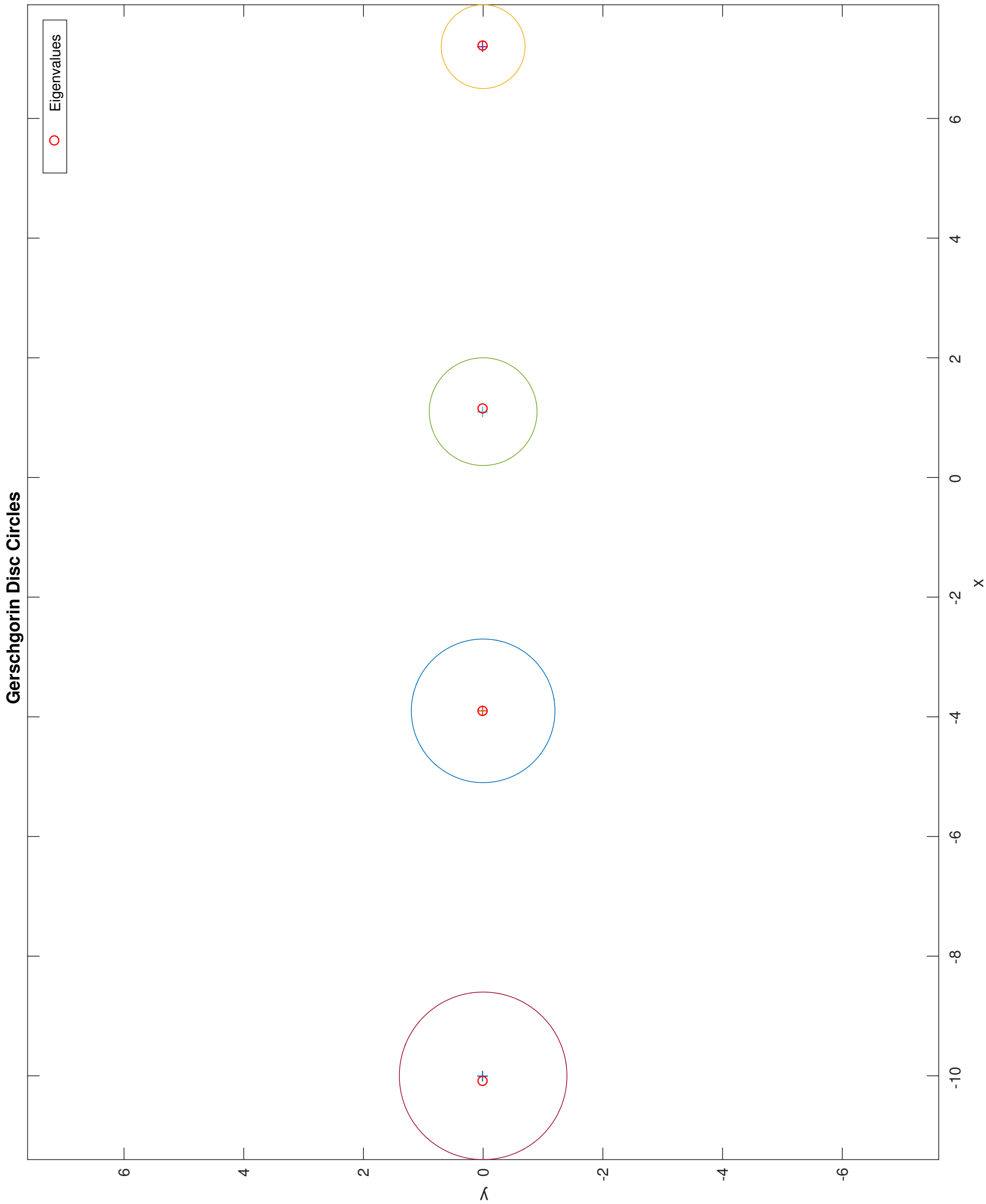
Q3

(a-d)

See attached.

(e)

The higher degree fits become less reliable than ones of lower degree because the least squares method works best for data that is relatively linear. As the polynomial order increases, it deviates further from the original function. Hence, the least squares method is not effective for higher degree polynomials.



Gerschgorin Disc Circles
Eigenvalues
x
y
6
4
2
0
-2
-4
-6
-8
-10

Comparing Least Squares Fit
Scatter of x and y
Order of 1
Least Sqaures
Order of 2
Order of 3
Order of 4
Order of 5
x
y