Math 521 Homework 2

Experimenting with SVD and Random Matrices

Angel Silvar

Math 521
California State University - Long Beach
March 20, 2022

1 Theory

1.1 Problem 1: Sub-spaces and Decomposition

This part of the homework is to demonstrate non-uniqueness of any given vector $x \in \mathbb{R}^n$ such that any linear combination of vectors $w_i \in W$ i = 1,2 returns x. Here $W_1 + W_2 = W$. Where W_i is a vector space. In other words,

$$x = w_1 + w_2 = w_1^{'} + w_2^{'}$$

and $w_i \neq w_i^{'}$ and $W_i \subset W$ and $w_i, w_i^{'} \subset W_i$ i = 1,2.

Let $W_i \in \mathbb{R}_i = 1, 2$. Let $w_1 = \begin{pmatrix} x \\ 0 \\ z_1 \end{pmatrix}$ and $w_2 = \begin{pmatrix} 0 \\ y \\ z_2 \end{pmatrix}$ This hows that the first vector lives in the

1.2 Problem 2: Gram-Schmidt and Orthogonal Projections

Let V $\in \mathbb{R}^3$ and $\vec{u}^1 = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$, $\vec{u}^1 = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$ and let x = $\begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}$. Define V= span$\{u_1, u_2\}$ With these parameters, the Gram-Schmidt algorithm will be used to find a projection matrix (or a projector) $\mathbb{P}$ such that $\mathbb{P}^2 = \mathbb{P}$

For our inputs for the Gram-Schmidt, we first normalize our u vectors.

$\hat{u}^1 = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$,$\hat{u}^2 = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$

Then, let $W \subset V$ where W := $\begin{pmatrix} \hat{u}^1 \\ \hat{u}^2 \end{pmatrix}$. Now, initiate Gram-Schmidt Process:

$\hat{v}^1 = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$

$\hat{v}^2 = \vec{v}_1$ - $proj_{\mathbf{v_1}} \mathbf{u_2} = \frac{\mathbf{u_2} \cdot \mathbf{v_1}}{\|\mathbf{v_1}\|^2} \mathbf{v_1} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} -1 \frac{}{\sqrt{5}} \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -4/5 \\ 2/5 \\ 1 \end{pmatrix}$

Where $\vec{v}_1$ and $\vec{v_2}$ are our orthogonal vectors. Now we normalize $\vec{v}_2$.

$\hat{v} = \frac{1}{3\sqrt{5}} \begin{pmatrix} -4/5 \\ 2/5 \\ 1 \end{pmatrix}$

To form $\mathbb{P}^1$: $\hat{v}_1 \hat{v}_1^T = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 & 2 & 0 \end{pmatrix} \frac{1}{\sqrt{5}} \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} = \frac{1}{5} \begin{pmatrix} 1 & 2 & 0 \\ 2 & 4 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

To form $\mathbb{P}^2$: $\hat{v}_2 \hat{v}_2^T = \frac{1}{3\sqrt{5}} \begin{pmatrix} -4/5 & 2/5 & 1 \end{pmatrix} \frac{1}{3\sqrt{5}} \begin{pmatrix} -4/5 \\ 2/5 \\ 1 \end{pmatrix}$

$= \frac{1}{45} \begin{pmatrix} 16/25 & -8/25 & -4/5 \\ -8/25 & 4/25 & 2/5 \\ -4/5 & 2/5 & 1 \end{pmatrix}$

Then $\mathbb{P} = \mathbb{P}_1 + \mathbb{P}_2 = \frac{1}{5} \begin{pmatrix} 1 & 2 & 0 \\ 2 & 4 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \frac{1}{45} \begin{pmatrix} 16/25 & -8/25 & -4/5 \\ -8/25 & 4/25 & 2/5 \\ -4/5 & 2/5 & 1 \end{pmatrix} = \begin{pmatrix} 25/45 & 10/45 & -4/9 \\ 10/45 & 40/45 & 2/9 \\ -4/9 & 2/9 & 5/9 \end{pmatrix}$

Then, P $= \begin{pmatrix} 25/45 & 10/45 & -4/9 \\ 10/45 & 40/45 & 2/9 \\ -4/9 & 2/9 & 5/9 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}$

Note = $\mathbb{P}^2 = \mathbb{P}$

2 Computing

This part of the assignment is to grab an image and use the given data set, then apply certain algorithms and determine the outputs. The algorithm that will be used will be

S $= \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$ where the S_x and S_y are both scaling factors for the x and y direction. Here the algorithm was created in such a way that the translation and transformation of the data set was all done at once. (Instead of the algorithm processing the image 1 operation at a time). Consider that X is our matrix with homogeneous coordinates, P is a point within the data set , and S is our transformation matrix.

In other words,

$$S(X - P) + P = SX - SP + P$$

Here SX is our transformation. SP - P is our translation.

The picture used is of the solar system.

Figure 1 – Default Picture of Solar System

2.1 Problem 1: Scale the Data set

This part of the assignment is to see how

```

Img = imread( 'ssystem.jpg');
Img = rgb2gray(Img);
alpha = [1 ,2]';
P = [ 50 , 30 , 1]';

[newImg] = scale ( Img , alpha , P );
image(newImg)

function [newImg] = scale(Img,alpha,P)


S = diag([alpha', 1]);
P_new = S*P;
S(1:2,3) = P(1:2)-P_new(1:2);

Img = double(Img);
[m,n] = size(Img);

```

```
[X,Y] = meshgrid(1:n,1:m);

coords = [X(:)'; Y(:)'; ones(1, m*n)];

new_coords = S*coords;
sf = interp2(X,Y,Img, new_coords(1,:)', new_coords(2,:)');
newImg = reshape(sf, m,n);
newImg = uint8(newImg);

end
```

```

Img = imread( 'ssystem.jpg');
Img = rgb2gray(Img);
image(Img)
```

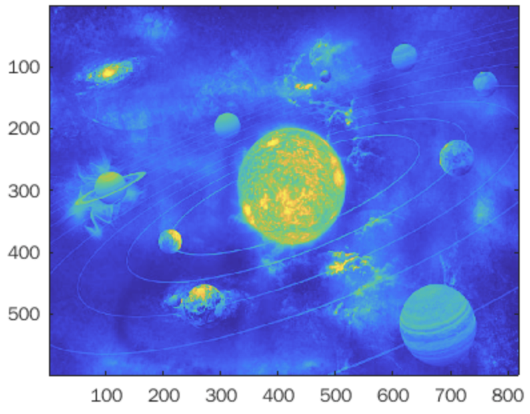

Figure 2 – rgb2gray function is applied.

```
alpha = [1 ,2]';
P = [ 50 , 30 , 1]';
[newImg] = scale ( Img , alpha , P );
image(newImg)
```

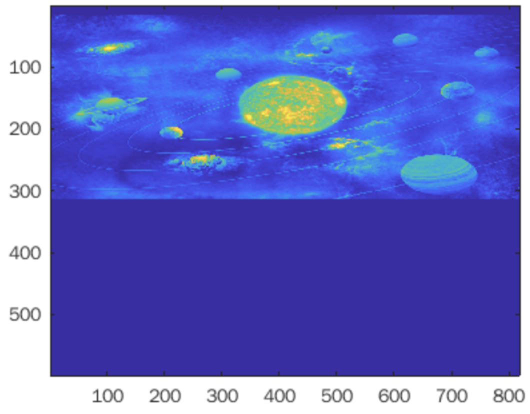

Figure 3 – Shifted upward picture of Solar System

```
alpha = [0.5 ,1]';
P = [ 50 , 30 , 1]';
[newImg] = scale ( Img , alpha , P );
image(newImg)
```

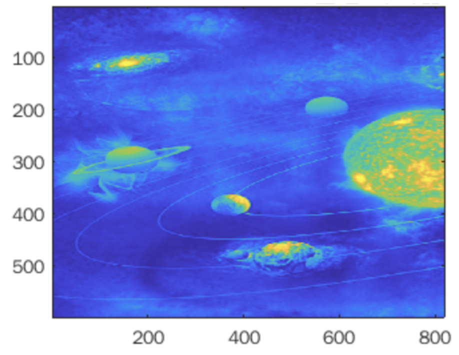

Figure 4 – Zoomed in picture of Solar System

```
alpha = [2 ,1]';
P = [ 50 , 30 , 1]';
[newImg] = scale ( Img , alpha , P );
image(newImg)
```

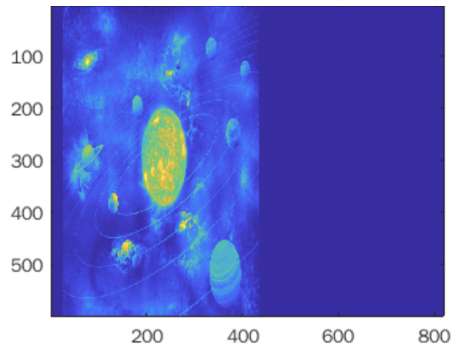

Figure 5 – Left shifted picture of Solar System

2.2 Problem 2: Translate a 2D image Horizontally and Vertically

Here the algorithm used T $= \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$ applied to the homogeneous coordinates made in problem 1.

```
tx = 350 ;
ty = 6;

Img = imread( 'ssystem.jpg');
Img = rgb2gray(Img);
Img = double(Img);
```

```

[newImg] = translate( Img, tx , ty );
image(newImg)

function [newImg] = translate(Img, tx, ty)
S = eye(3);
S(1:2,3) = [tx,ty]';

Img = double(Img);
[m,n] = size(Img);

[X,Y] = meshgrid(1:n,1:m);

coords = [X(:)'; Y(:)'; ones(1, m*n)];

new_coords = S*coords;

sf = interp2(X,Y,Img, new_coords(1,:)', new_coords(2,:)');

newImg = reshape(sf, m,n);

newImg = uint8(newImg);

end
```

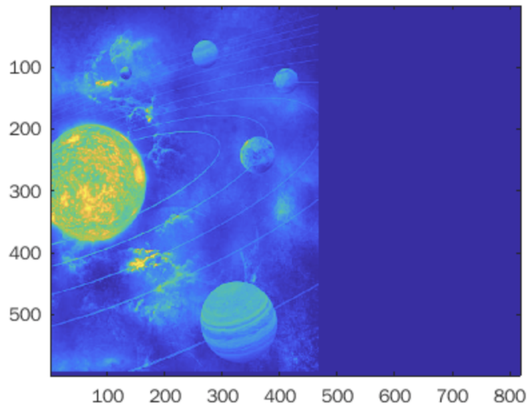

Figure 6 – Result is similar to Figure 5.

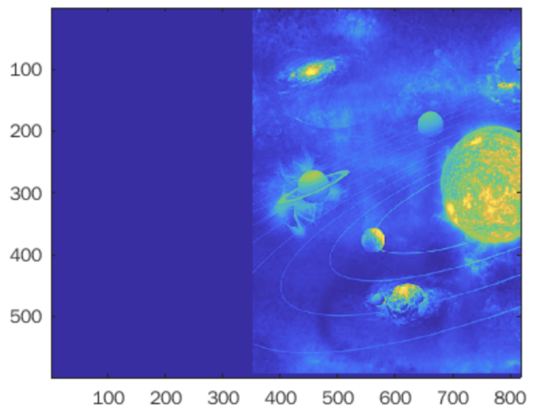

Figure 7 – Result is similar to Figure 5 but translated to the right.

2.3 Problem 3: Rotating Matrix

The part of this assignment is to rotate the data set around. In this case, the picture of the solar system will be rotated around with the algorithm: T $= \begin{pmatrix} \cos(\theta) & \text{-}\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$. Note that the angle of rotation is θ

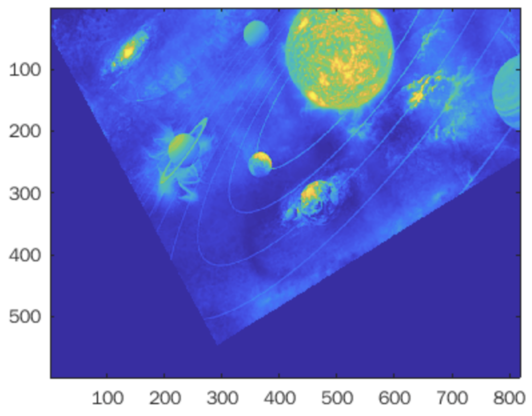

Figure 8 – Image was rotated.

```

Img = imread( 'ssystem.jpg');
Img = rgb2gray(Img);
Img = double(Img);


theta = 3.14 / 6 ;

P = [ 50 , 30 , 1]' ;


[newImg] = rotated( Img , theta , P );
image(newImg)


```

```
function [newImg] = rotated(Img,theta,P)

S = [cos(theta) -sin(theta) 0; sin(theta) cos(theta) 0; 0 0 1];
P_new = S*P;
S(1:2,3) = P(1:2)-P_new(1:2);

Img = double(Img);
[m,n] = size(Img);

[X,Y] = meshgrid(1:n,1:m);

coords = [X(:)'; Y(:)'; ones(1, m*n)];

new_coords = S*coords;

sf = interp2(X,Y,Img, new_coords(1,:)', new_coords(2,:)');

newImg = reshape(sf, m,n);

newImg = uint8(newImg);

end
```

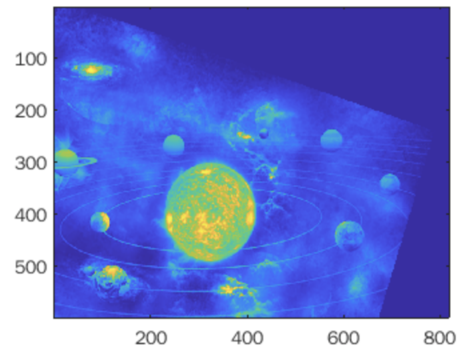

Figure 9 – Image was rotated above was mirrored.

2.4 Problem 4: Novelty Filter

Here the, the 3 matrices that represented 3 different black blocks was concatenated into a vector and an algorithm was used (in problem 2 of the theory section) in order to project the 4th block onto this created vector of concatenated 3 blocks. In short, the projected 4th block onto the orthogonalized basis and created a matrix projector. The result is X $= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$.

It must be noted that there is no possible way to create the the 2nd column found in the 3rd group of blocks. The orthogonal residual is found in the matrix above.

```

A = [1 1 1 1;
     1 0 0 1;
     1 0 0 1;
     1 0 0 1;
     1  1  1  1];

B = [1  1  1  1;
     0  0  0  1;
     0  0  0  1;
     0  0  0  1;
     1  1  1  1];

C = [1  1  1  1;
     0  0  0  0;
     0  0  0  0;
     0  0  0  0;
     1  1  1  1];

X = [1  1  1  1;
     0  1  0  1;
     0  1  0  1;
     0  1  0  1;
     1  1  1  1];

W = [A(:) B(:) C(:)];
M = orth(W);
P = M*M';
Wperp = (eye(length(P))-P)*X(:);
Wperpdisp = [Wperp(1:5) Wperp(6:10) Wperp(11:15) Wperp(16:20)]
```

2.5 Problem 5: Rank-k Approximations and Cumulative Engery

The SVD algorithm is applied to the image.

```
Img = imread( 'ssystem.jpg');
Img = rgb2gray(Img);
Img = double(Img);

[U,S,V] = svd( Img, "vector");

figure(2)
x = 1:length(S);
s = abs(diag(S));

plot(x,s)
title('Singular Values')
xlabel('Index')
ylabel('Singular Value')


E = sum(s.^2);
E_k = zeros( length(S)) ;
for i = 1:length(s)
    E_k(i) = sum(s.^2) / E;
end

rank = find(E_k >= 0.95 , 1, 'first');

k=10;
```

```

Img_10 = U(:,1:k)*S(1:k,1:k)*V(:,1:k)' ;
err_rel_10 = s(11)/s(1);

k = 50;

Img_50 = U(:,1:k)*S(1:k,1:k)*V(:,1:k)';
err_rel_50 = s(51)/s(1);

k = 100 ;

Img_100 = U(:,1:k)*S(1:k,1:k)*V(:,1:k)';
err_rel_100 = s(101)/s(1);


k = 200;

Img_200 = U(:,1:k)*S(1:k,1:k)*V(:,1:k)' ;
err_rel_200 = s(201)/s(1);


figure(2)
subplot(2)
subplot(2,2,1)
imagesc(Img_10, colormap(gray))
title('Rank of 10, ')


figure(2)
subplot(2,2,1)
imagesc(Img_10), colormap(gray)
title('Rank 10, rel. error = )
subplot(2,2,2)
imagesc(Img_50), colormap(gray)
title('Rank 50, rel. error = )
subplot(2,2,3)
imagesc(Img_100), colormap(gray)
title('Rank 100, rel. error = )
subplot(2,2,4)
imagesc(Img_200), colormap(gray)
title('Rank 200, rel. error = )
```

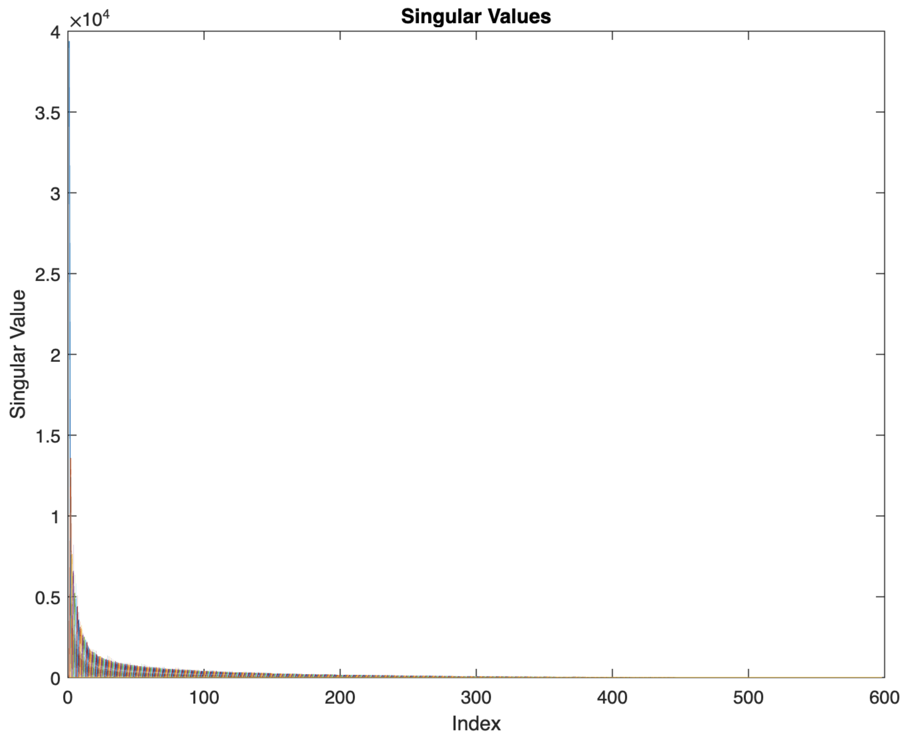

Figure 10 – Image was rotated above was mirrored.

Then following are the images that are approximated with Rank 10, 50 , 100, and 200.