

## Sample Solutions for Assignment 2.

Reading: Lectures 3-5 in the text.

1. p. 24, Exercise 3.5.

The same is true for the Frobenius norm since

$$\begin{aligned}\|E\|_F^2 &= \sum_{i,j} |E_{ij}|^2 = \sum_{i=1}^m \sum_{j=1}^n |u_i \bar{v}_j|^2 = \sum_{i=1}^m \sum_{j=1}^n |u_i|^2 |\bar{v}_j|^2 = \\ &= \left( \sum_{i=1}^m |u_i|^2 \right) \left( \sum_{j=1}^n |\bar{v}_j|^2 \right) = \|u\|_F^2 \|v\|_F^2.\end{aligned}$$

2. p. 30, Exercise 4.1.

(a)

$$\begin{bmatrix} 3 & 0 \\ 0 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

(b)

$$\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

(c)

$$\begin{bmatrix} 0 & 2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = I_{3 \times 3} \begin{bmatrix} 2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

(d) The singular values are  $\sqrt{2}$  and 0.

$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} = I \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 \\ \sqrt{2}/2 & -\sqrt{2}/2 \end{bmatrix}.$$

(e) The singular values are 2 and 0.

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 \\ \sqrt{2}/2 & -\sqrt{2}/2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 \\ \sqrt{2}/2 & -\sqrt{2}/2 \end{bmatrix}$$

3. p. 30, Exercise 4.5.

Following the existence proof in Theorem 4.1, we see that it involves finding a vector  $v_1$  with norm 1 such that  $\|Av_1\| = \|A\| \equiv \sigma_1$ . If that vector is real, then  $u_1 = Av_1$  is real, and these can be completed to real orthonormal bases  $\{u_1, u_2, \dots, u_m\}$  and  $\{v_1, v_2, \dots, v_m\}$  for  $\mathbf{R}^m$ . If  $U_1$  is the matrix whose columns are  $u_1, \dots, u_m$  and  $V_1$  is the matrix whose columns are  $v_1, \dots, v_m$ , then, as in Theorem 4.1,

$$U_1^* A V_1 = \begin{bmatrix} \sigma_1 & 0 \\ 0 & B \end{bmatrix},$$

where  $B$  is an  $m - 1$  by  $n - 1$  matrix. If we assume that real matrices of size  $m - 1$  by  $n - 1$  have real SVD's and follow the remaining steps of the theorem, we see that they will produce a real SVD of  $A$ . [As a base case, any real  $k \times 1$  matrix has a real reduced SVD:

$$\begin{bmatrix} a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} a_1/\sigma_1 \\ \vdots \\ a_k/\sigma_1 \end{bmatrix} \cdot \sigma_1 \cdot 1,$$

where  $\sigma_1 = \sqrt{\sum_{i=1}^k a_i^2}$ . This can be completed to a real full SVD.]

Thus, we need only prove that if  $A$  is real, then there is a real nonzero vector  $w$  such that  $\|Aw\|/\|w\| = \max_{\substack{v \in \mathbf{C}^n \\ v \neq 0}} \|Av\|/\|v\|$ . Suppose that this maximum is attained by a complex vector  $x + iy$ . Then, just by definition of the 2-norm,

$$\|A(x + iy)\|^2 = \sum_{j=1}^m |(Ax)_j + i(Ay)_j|^2 = \sum_{j=1}^m (Ax)_j^2 + (Ay)_j^2 = \|Ax\|^2 + \|Ay\|^2. \quad (1)$$

Similarly,  $\|x + iy\|^2 = \|x\|^2 + \|y\|^2$ . Since  $\|A(x + iy)\| = \|A\| \cdot \|x + iy\|$ , we have from (1)

$$\|Ax\|^2 + \|Ay\|^2 = \|A\|^2(\|x\|^2 + \|y\|^2). \quad (2)$$

But we also know that  $\|Ax\|^2 \leq \|A\|^2 \cdot \|x\|^2$  and  $\|Ay\|^2 \leq \|A\|^2 \cdot \|y\|^2$ . The only way that we can have equality in (2), then, is if  $\|Ax\| = \|A\| \cdot \|x\|$  and  $\|Ay\| = \|A\| \cdot \|y\|$ . At least one of  $x$  and  $y$  must be nonzero, so we therefore have a nonzero *real* vector, say,  $x$  such that  $\|Ax\| = \|A\| \cdot \|x\|$ . The result now follows.

4. p. 37, Exercise 5.4.

We can write the matrix in the form

$$\begin{bmatrix} 0 & V\Sigma U^* \\ U\Sigma V^* & 0 \end{bmatrix}.$$

If we apply it to the matrix

$$\begin{bmatrix} V & -V \\ U & U \end{bmatrix}, \quad (3)$$

the result is

$$\begin{bmatrix} V\Sigma & V\Sigma \\ U\Sigma & -U\Sigma \end{bmatrix} = \begin{bmatrix} V & -V \\ U & U \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & -\Sigma \end{bmatrix}.$$

Thus, the eigenvalues of the larger matrix are plus and minus the singular values of  $A$  and the eigenvectors are the columns of the matrix in (3).

5. **Image compression.** In Matlab, type `imagedemo`. You will see a picture of an Albrecht Durer print. Type `who` to see what variables it has used and type `type imagedemo` to see the actual Matlab code that you have run. You will see at the end that it executes the commands:

```
imagesc(X);
colormap(map);
axis off;
```

The 648 by 509 matrix  $X$  contains a grayscale number (from 1 to 128) for each pixel in a grid. This number determines how dark or light that pixel will be shaded when the command `imagesc(X)` is executed. This is fine if one can store a 648 by 509 matrix, but if there are many such images and they are, say, being sent from outer space, using this large a matrix to represent each one could be prohibitive!

Compute the SVD of  $X$ . Try executing the above commands with  $X$  replaced by some low rank approximations formed from the largest singular values and corresponding singular vectors, and decide about how many singular values/vectors are needed to make the picture recognizable. Turn in a few plots showing how the picture improves as you increase the rank of the approximation used. Label each plot with the rank of the approximation used. [You can put several plots on one page using the `subplot` command. Type `help subplot` to see exactly how it works. You can save your plots to a file by typing `print -depsc hw2plots.eps` where the filename `hw2plots` can be replaced by any name you like.]

Following are some sample plots, produced by the code below:

```
imagedemo                                % Generate original picture.
print -depsc original.eps                % Print it to a file.

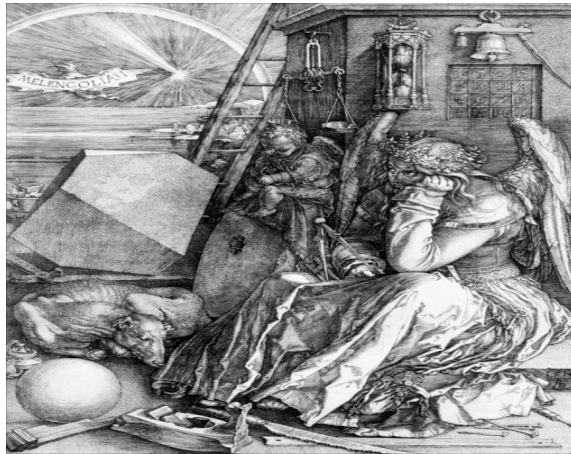
[U,Sigma,V] = svd(X);                    % Compute the SVD of X.
Vp = V';                                  % Store V'.
X1 = U(:,1)*Sigma(1,1)*Vp(1,:);           % Rank 1 approximation to X.
X10 = U(:,1:10)*Sigma(1:10,1:10)*Vp(1:10,:); % Rank 10 approximation to X.
X50 = U(:,1:50)*Sigma(1:50,1:50)*Vp(1:50,:); % Rank 50 approximation to X.
X100 = U(:,1:100)*Sigma(1:100,1:100)*Vp(1:100,:); % Rank 100 approximation to X.

figure(2)                                % Go to a new figure box.
subplot(2,2,1)                            % Put 4 plots on one page.
imagesc(X1); colormap(map); axis off      % Image from rank 1 approximation.
title('Rank 1')
```

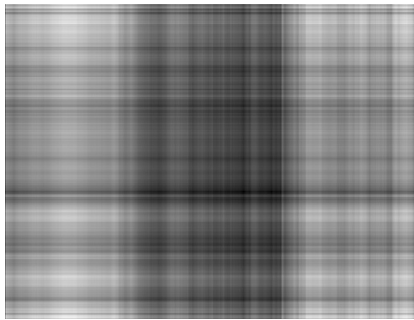
```

subplot(2,2,2)
imagesc(X10); colormap(map); axis off      % Image from rank 10 approximation.
title('Rank 10')
subplot(2,2,3)
imagesc(X50); colormap(map); axis off      % Image from rank 50 approximation.
title('Rank 50')
subplot(2,2,4)
imagesc(X100); colormap(map); axis off     % Image from rank 100 approximation.
title('Rank 100')
print -depsc approximations.eps

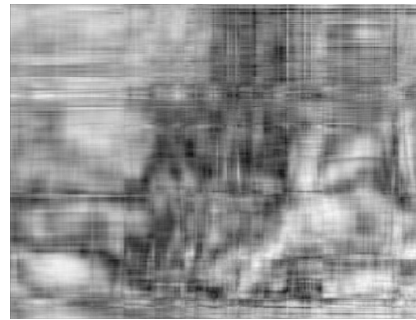
```



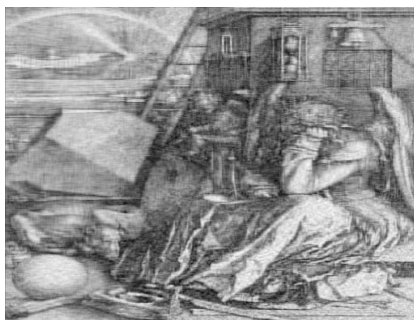
Rank 1



Rank 10



Rank 50



Rank 100

