MA 402: Project 3

Matthew Murray, Johnathan Rhyne

Fall 2019

Instructions:

- Detailed instructions regarding submission are available on the class website[1].
- The zip file should contain five files project3.pdf, project3.tex, classnotes.sty, swift.mat, and deblur.mat.
- More instructions:
  - MATLAB users: use `loadmat` (type `who` to display what variables are in your workspace.
  - Python users: use `scipy.io.loadmat`. This will return a dictionary with all the necessary variables.

1 Pen-and-paper exercises

The problems from this section total 20 points.

1) (10 points) Consider the matrix $\boldsymbol{A}$ with the SVD

$$\boldsymbol{A} = \begin{bmatrix} 4 & 0 \\ -5 & -3 \\ 2 & 6 \end{bmatrix} = \boldsymbol{U} \begin{bmatrix} 6\sqrt{2} & 0 \\ 0 & 3\sqrt{2} \\ 0 & 0 \end{bmatrix} \boldsymbol{V}^\top,$$

where

$$\boldsymbol{U} = \frac{1}{3}\begin{bmatrix} 1 & -2 & 2 \\ -2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix} \qquad \boldsymbol{V} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}.$$

(a) (0 points) Verify for yourself that it is indeed the SVD of $\boldsymbol{A}$, and that $\boldsymbol{U}, \boldsymbol{V}$ are orthogonal.

(b) (1 point) What is the rank of this matrix?

(c) (2 points) From the full SVD of $\boldsymbol{A}$, write down the reduced SVD of $\boldsymbol{A}$.

(d) (3 points) Compute the best rank-1 approximation of A.

(e) (2 points) Compute the 2-norm and the Frobenius norms of $\boldsymbol{A}$.

(f) (2 points) Using the SVD of $\boldsymbol{A}$, write down the SVD of $\boldsymbol{A}^\top$ and $\boldsymbol{A}^\top\boldsymbol{A}$.

Solution:

(a) The SVD of $\boldsymbol{A}$ is valid.

[1] `https://github.ncsu.edu/asaibab/ma402_fall_2019/blob/master/projects.md`

(b) The rank of a matrix is the maximum number of linearly independent columns or rows of the matrix. Because we have the SVD of $\boldsymbol{A}$, we can find the rank by counting the number on nonzero singular values of $\boldsymbol{A}$.

$$\therefore rank(\boldsymbol{A}) = 2$$

(c) In general, the reduced SVD of a Matrix A is...

$$\boldsymbol{A} = \sum_{j=1}^{r} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^T$$

where r is the rank of $\boldsymbol{A}$ and $\boldsymbol{u}_j$ and $\boldsymbol{v}_j$ are the left and right singular vectors of $\boldsymbol{A}$, respectively. Therefore, the reduced SVD of A is...

$$\boldsymbol{A} = \frac{1}{3}\begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix} 6\sqrt{2}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \end{bmatrix} + \frac{1}{3}\begin{bmatrix} -2 \\ 1 \\ -2 \end{bmatrix} 3\sqrt{2}\frac{1}{\sqrt{2}}\begin{bmatrix} -1 & 1 \end{bmatrix}$$

(d) The best rank-k approximation of A is...

$$\boldsymbol{A}_k = \sum_{j=1}^{k} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^T$$

where $k \leq r$. Here set $k = 1$.

$$\boldsymbol{A}_1 = \frac{1}{3}\begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix} 6\sqrt{2}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \end{bmatrix}$$

(e)

$$\|\boldsymbol{A}\|_F = \sqrt{4^2 + 0^2 + (-5)^2 + (-3)^2 + 2^2 + 6^2} = \sqrt{90}$$
$$\|\boldsymbol{A}\|_2 = \sigma_{\max}(\boldsymbol{A}) = 6\sqrt{2}$$

(f) The SVD is...

$$\boldsymbol{A} = \boldsymbol{U}\begin{bmatrix} \boldsymbol{\Sigma} \\ 0 \end{bmatrix}\boldsymbol{V}^\top$$

where

$$\begin{bmatrix} \boldsymbol{\Sigma} \\ 0 \end{bmatrix} = \begin{bmatrix} 6\sqrt{2} & 0 \\ 0 & 3\sqrt{2} \\ 0 & 0 \end{bmatrix} \qquad \boldsymbol{U} = \frac{1}{3}\begin{bmatrix} 1 & -2 & 2 \\ -2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix} \qquad \boldsymbol{V} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}.$$

$$\therefore \boldsymbol{A}^\top = (\boldsymbol{V}^\top)^\top \begin{bmatrix} \boldsymbol{\Sigma} & 0 \end{bmatrix} \boldsymbol{U}^\top = \boldsymbol{V}\begin{bmatrix} \boldsymbol{\Sigma} & 0 \end{bmatrix} \boldsymbol{U}^\top = \boldsymbol{V}\begin{bmatrix} 6\sqrt{2} & 0 & 0 \\ 0 & 3\sqrt{2} & 0 \end{bmatrix}\boldsymbol{U}^\top$$

$$\therefore \boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{V}\begin{bmatrix} \boldsymbol{\Sigma} & 0 \end{bmatrix} \boldsymbol{U}^\top\boldsymbol{U}\begin{bmatrix} \boldsymbol{\Sigma} \\ 0 \end{bmatrix}\boldsymbol{V}^\top = \boldsymbol{V}\begin{bmatrix} \boldsymbol{\Sigma}^2 \end{bmatrix}\boldsymbol{V}^\top = \boldsymbol{V}\begin{bmatrix} 72 & 0 \\ 0 & 18 \end{bmatrix}\boldsymbol{V}^\top$$

2) (10 points) Let $A \in \mathbb{R}^{m\times n}$. Recall: by definition, the Frobenius norm of $\boldsymbol{A}$ is $\|\boldsymbol{A}\|_F = \left(\sum_i \sum_j |a_{ij}|^2\right)^{1/2}$. In this problem, we will derive the formula

$$\|\boldsymbol{A}\|_F = \left(\sigma_1^2 + \cdots + \sigma_r^2\right)^{1/2}.$$

(a) The trace of a square matrix is the sum of its diagonals entries. Show (an alternative representation for the Frobenius norm):

$$\|\boldsymbol{A}\|_F = \left(\mathsf{trace}\,(\boldsymbol{A}^\top\boldsymbol{A})\right)^{1/2}.$$

(b) Let $\boldsymbol{C}, \boldsymbol{D}$ be $n \times n$ square matrices. Show: $\mathsf{trace}\,(\boldsymbol{CD}) = \mathsf{trace}\,(\boldsymbol{DC})$.
Remark: This is known as the cyclic property of trace, which is true despite the fact that in general $\boldsymbol{CD} \neq \boldsymbol{DC}$. A consequence of the cyclic property is: if $\boldsymbol{E}$ has the same size as $\boldsymbol{C}, \boldsymbol{D}$, it implies

$$\mathsf{trace}\,(\boldsymbol{CDE}) = \mathsf{trace}\,(\boldsymbol{DEC}) = \mathsf{trace}\,(\boldsymbol{ECD}).$$

(c) Using parts (a and b) complete the proof to show $\|\boldsymbol{A}\|_F = \left(\sigma_1^2 + \cdots + \sigma_r^2\right)^{1/2}$.

(d) Show: $\|\boldsymbol{A}\|_2 \leq \|\boldsymbol{A}\|_F \leq \sqrt{r}\|\boldsymbol{A}\|_2$.

Solution:

(a) We know that, by definition, the i^{th} row of $\boldsymbol{A}$ is the i^{th} column of $\boldsymbol{A}^\top$ and that i^{th} column of $\boldsymbol{A}$ is the i^{th} row of $\boldsymbol{A}^\top$

Now, we let a_i be the i^{th} row of $\boldsymbol{A}$ and $a_i^\top$ be the i^{th} column of $\boldsymbol{A}^\top$
We also know that if $\boldsymbol{A}$ is an mxn matrix then $\boldsymbol{A}^\top\boldsymbol{A}$ is nxn

We can write $\boldsymbol{A}^\top\boldsymbol{A}$ as $\sum_{i=1}^n a_i^\top a_i$

If we look at a general $a_i^\top a_i$ we notice the following:

$$\begin{bmatrix} a_{i1} \\ a_{i2} \\ \ldots \\ a_{in} \end{bmatrix} \qquad \begin{bmatrix} a_{i1} & a_{i2} & \ldots & a_{in} \end{bmatrix}$$

We are allowed to represent this outer product using the same elements since we are looking at $\boldsymbol{A}^\top\boldsymbol{A}$

Since we only care about the diagonal of the product, we will only consider the diagonal of each outer product, which will take the form of the following:

$$\begin{bmatrix} a_{i1}^2 & & & \\ & a_{i2}^2 & & \\ & & \ldots & \\ & & & a_{in}^2 \end{bmatrix}$$

From the above matrix, we can see that we need to sum up n number of rows of $\boldsymbol{A}$ and columns of $\boldsymbol{A}^\top$.
Using this information, we notice that the j^{th} diagonal element of $\boldsymbol{A}^\top\boldsymbol{A}$ can be written as:

$$\sum_{i=1}^{n} a_{ij}^2 \tag{1}$$

Since $\mathsf{trace}\,(\boldsymbol{A}^\top\boldsymbol{A})$ is the sum of its diagonal elements, we must sum 1 for every j less than or equal to n, we get the following summation:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}|^2 = \|\boldsymbol{A}\|_F^2 \tag{2}$$

From 2 We know that $\mathsf{trace}\left(\boldsymbol{A}^\top \boldsymbol{A}\right) = \|\boldsymbol{A}\|_F^2 \implies \sqrt{\mathsf{trace}\left(\boldsymbol{A}^\top \boldsymbol{A}\right)} = \|\boldsymbol{A}\|_F$

(b) Given two general nxn matrices, $\boldsymbol{C}$ and $\boldsymbol{D}$, we can represent $\boldsymbol{DC}$ as: $\sum_{j=1}^{n} d_j \hat{c}_j$
Where d_j is the j^{th} column of $\boldsymbol{D}$ and $\hat{c}_j$ is the j^{th} row of $\boldsymbol{C}$

Looking at the j^{th} term of this summation, we get the following:

$$\begin{bmatrix} d_{1j} \\ d_{2j} \\ \dots \\ d_{nj} \end{bmatrix} \qquad \begin{bmatrix} c_{j1} & c_{j2} & \dots & c_{jn} \end{bmatrix}$$

And when we multiply this outer product, we get a matrix with the following diagonal elements:

$$\begin{bmatrix} d_{1j}c_{j1} & & & \\ & d_{2j}c_{j2} & & \\ & & \dots & \\ & & & d_{nj}c_{jn} \end{bmatrix}$$

Which means we can write the $\mathsf{trace}\left(d_j \hat{c}_j\right)$ as $\sum_{i=1}^{n} d_{ij}c_{ji}$
To find the trace of $\boldsymbol{DC}$ we need only sum up each diagonal for every j less than or equal to n, or:

$$\mathsf{trace}\left(\boldsymbol{DC}\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}c_{ji}$$

By swapping $\boldsymbol{C and D}$ we also get the following:

$$\mathsf{trace}\left(\boldsymbol{CD}\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}d_{ji}$$

We will prove the following equation holds for every $n \in \mathbb{R}$

$$\mathsf{trace}\left(\boldsymbol{CD}\right) = \mathsf{trace}\left(\boldsymbol{DC}\right) \tag{3}$$

To show that (3), we prove by induction:

Proof. Show (3) holds for n = 1:

$$\mathsf{trace}\left(\boldsymbol{CD}\right) = \sum_{i=1}^{1}\sum_{j=1}^{1} c_{ij}d_{ji} = c_{11}d_{11} = d_{11}c_{11} = \sum_{i=1}^{1}\sum_{j=1}^{1} d_{ij}c_{ji} = \mathsf{trace}\left(\boldsymbol{DC}\right)$$

Now we show that (3) holds for an arbitrary $n+1$ given that n holds:

$$\sum_{i=1}^{n+1}\sum_{j=1}^{n+1} c_{ij}d_{ji} = \sum_{i=1}^{n+1}\sum_{j=1}^{n} c_{ij}d_{ji} + \sum_{i=1}^{n+1} c_{ij}d_{ji} = \sum_{i=1}^{n+1}\sum_{j=1}^{n} c_{ij}d_{ji} + \sum_{i=1}^{n} c_{ij} + c_{n+1\ n+1}d_{n+1\ n+1}$$

We must now look at the first term and get:

$$\sum_{i=1}^{n+1}\sum_{j=1}^{n} c_{ij}d_{ji} = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}d_{ji} + \sum_{j=1}^{n} c_{ij}d_{ji}$$

Combining both the above equations we get the following:

$$\sum_{i=1}^{n+1}\sum_{j=1}^{n+1} c_{ij}d_{ji} = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}d_{ji} + \sum_{j=1}^{n} c_{ij}d_{ji} + \sum_{i=1}^{n} c_{ij}d_{ji} + c_{n+1\ n+1}d_{n+1\ n+1}$$

By both our induction hypothesis and the commutativity of numbers in a field (both Real and Complex numbers) we rewrite above as:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ji}c_{ij} + \sum_{j=1}^{n} d_{ji}c_{ij} + \sum_{i=1}^{n} d_{ji}c_{ij} + d_{n+1\ n+1}c_{n+1\ n+1}$$

Reversing our methods from above, we get the following:

$$\sum_{i=1}^{n+1}\sum_{j=1}^{n+1} d_{ji}c_{ij}$$

Since this still cycles through every term of $\boldsymbol{D}$ and $\boldsymbol{C}$, we can flip the i and j to get:

$$\sum_{i=1}^{n+1}\sum_{j=1}^{n+1} d_{ij}c_{ji} = \mathsf{trace}\left(\boldsymbol{DC}\right)$$

$\therefore$ we have shown that $\mathsf{trace}\left(\boldsymbol{CD}\right) = \mathsf{trace}\left(\boldsymbol{DC}\right)$ for $\boldsymbol{C}, \boldsymbol{D} \in \mathbb{R}^{nxn}$

$\square$

(c) Since $\boldsymbol{A}^\top\boldsymbol{A}$ is nxn regardless of if $\boldsymbol{A}$ has more rows than columns, more columns than rows, or the same number of each, and that we have proved in class that $\boldsymbol{A}^\top\boldsymbol{A}$ always takes the form of:

$$\boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^\top \tag{4}$$

We need only to look at (4).

As stated from the consequence of part b, we know the following

$$\|\boldsymbol{A}\|_F^2 = \mathsf{trace}\left(\boldsymbol{A}^\top\boldsymbol{A}\right) = \mathsf{trace}\left(\boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^\top\right) = \mathsf{trace}\left(\boldsymbol{V}\boldsymbol{V}^\top\boldsymbol{\Sigma}^2\right) = \mathsf{trace}\left(\boldsymbol{\Sigma}^2\right)$$

We also know that, by construction, $\sigma_n \geq \sigma_{n+1}$ and that $\exists r \leq n \ni \sigma_{r+1} = \cdots = \sigma_n = 0$
Therefore, we get the following:

$$\mathsf{trace}\left(\Sigma^2\right) = \sum_{i=1}^{n}\sigma_i^2 = \sum_{i=1}^{r}\sigma_i^2 + \sum_{i=r+1}^{n}\sigma_i^2 = \sum_{i=1}^{r}\sigma_i^2 + 0 = \sum_{i=1}^{r}\sigma_i^2$$

$$\therefore \|\boldsymbol{A}\|_F^2 = \mathsf{trace}\left(\Sigma^2\right) = \sum_{i=1}^{r}\sigma_i^2 \implies \|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{r}\sigma_i^2}$$

(d) By definition, $\|\boldsymbol{A}\|_2 = \sigma_1$

$$\|\boldsymbol{A}\|_2 = \sigma_1 = \sqrt{\sigma_1^2} \leq \sqrt{\sigma_1^2 + \cdots + \sigma_r^2} = \|\boldsymbol{A}\|_F \implies \|\boldsymbol{A}\|_2 \leq \|\boldsymbol{A}\|_F$$

For the second part of the inequality, we must first prove that $r\sigma_1^2 \geq \sum_{i=1}^{r}\sigma_i^2$

Proof. let r = 1 then we have:

$$\sigma_1^2 \geq \sigma_1^2$$

Which is trivially true
Now we look at $r = n \ni n \leq q - 1$ where A is a matrix of size qxq

$$n\sigma_1 \geq \sum_{i=1}^{n} \sigma_i^2 \implies n \geq \frac{\sigma_1^2}{\sigma_1^2} + \frac{\sigma_2^2}{\sigma_1^2} + \cdots + \frac{\sigma_n^2}{\sigma_1^2} \implies n \geq 1 + \frac{\sigma_2^2}{\sigma_1^2} + \cdots + \frac{\sigma_n^2}{\sigma_1^2}$$

By adding 1 to both sides, we get the following:

$$n + 1 \geq 1 + \frac{\sigma_2^2}{\sigma_1^2} + \cdots + \frac{\sigma_n^2}{\sigma_1^2} + 1$$

We must also note that by construction that $\sigma_1^2 \geq \sigma_n^2 \geq \sigma_{n+1}^2$
Which gives us $1 \geq \frac{\sigma_n^2}{\sigma_1^2} \geq \frac{\sigma_{n+1}^2}{\sigma_1^2}$
Therefore, we can rewrite $n + 1$ as:

$$n + 1 \geq 1 + \frac{\sigma_2^2}{\sigma_1^2} + \cdots + \frac{\sigma_n^2}{\sigma_1^2} + \frac{\sigma_{n+1}^2}{\sigma_1^2} \implies (n+1)\sigma_1^2 \geq \sum_{i=1}^{n+1} \sigma_i^2$$

Therefore we have shown that $r\sigma_1^2 \geq \sum_{i=1}^{r} \sigma_i^2 \; \forall r \in \mathbb{N}$

We also know that $\sqrt{r\sigma_1^2} \geq \sqrt{\sum_{i=1}^{r} \sigma_i^2}$

We have now shown that $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{r} \sigma_i^2} \leq \sqrt{r\sigma_1^2} = \sqrt{r}\sqrt{\sigma_1^2} = \sqrt{r}\sigma_1 = \sqrt{r}\|\boldsymbol{A}\|_2$

Finally, we have proven that $\|\boldsymbol{A}\|_2 \leq \|\boldsymbol{A}\|_F \leq \sqrt{r}\|\boldsymbol{A}\|_2$

□

2 Numerical exercises

The problems from this section total 30 points.

3) (15 points) *Compressing* and *Denoising* images.

(a) (0 points) Load the file 'swift.mat'. You will find the variables `A` and `An` which are both matrices of size 512×1024.

(b) (2 points) In a single figure with 2 subplots, plot the clean as well as the noisy matrices as images. Denote the corresponding matrices as $\boldsymbol{A}$ and $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{E}$, where $\boldsymbol{E}$ represents the noise that is added to the original image. Unfortunately, in real applications we do not know exactly how much noise is added.

(c) (2 points) Plot the first 100 singular values of $\boldsymbol{A}$ and $\tilde{\boldsymbol{A}}$. (*Hint:* Use the `semilogy` plotting function).

(d) (3 points) In a single figure with 9 different subplots, plot $\boldsymbol{A}_k$ (the best rank-k approximation to $\boldsymbol{A}$) as images for $k = 10, 20, \ldots, 90$ (use these same values of k for the rest of this problem).

(e) (2 points) As two subplots of the same figure, plot (left panel) the storage cost of the truncated SVD as a function of k, (right panel) relative error of $\boldsymbol{A}_k$ (in the Frobenius norm) as a function of k. Comment on these two subplots. (Assume that each floating point number requires 1 unit of storage.)

(f) (3 points) Our proposed algorithm to denoise the image is to use a truncated SVD of the matrix corresponding to the noisy image, i.e., computing $\tilde{\boldsymbol{A}}_k$. In a single figure with 9 different subplots, plot $\tilde{\boldsymbol{A}}_k$ (the best rank-k approximation to $\boldsymbol{A}$) as images for $k = 10, 20, \ldots, 90$. Make sure to label each subplot.

(g) (2 points) Plot the relative error of the denoised image $\tilde{\boldsymbol{A}}_k$ (in the Frobenius norm) as a function of the truncation index k. For (approximately) what value of k is the minimum attained?

Instructions: In total, you have to submit 6 separate plots. Make sure to label each plot/subplot, and label the axes of the singular value and the error plots. MATLAB users may want to use commands like

```
figure, image(A), colorbar, colormap(gray(256))
```

Python users may want to try `imshow`.

Solution:

(a) Loaded Swift.m

(b) Clean and Noisy Matrices

$\boldsymbol{A}$ and $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{E}$

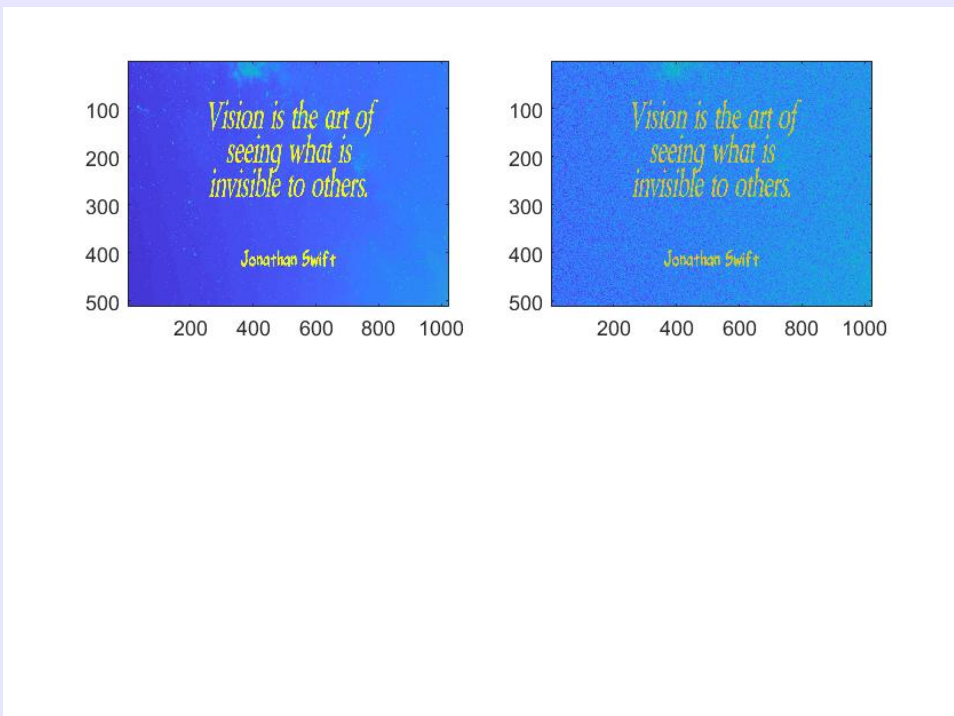

Figure 1: Image of Clean and Noisy Matrices

(c) First 100 Singular Values

First 100 Singular Values of $\boldsymbol{A}$ and $\tilde{\boldsymbol{A}}$

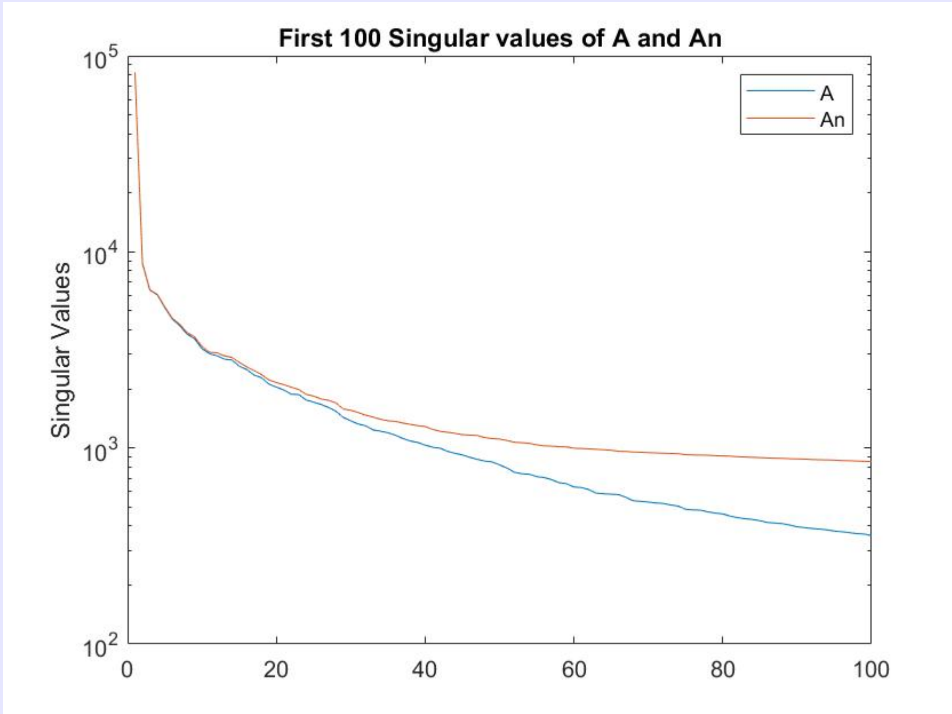

Figure 2: Plot of First 100 Singular Values

(d) Best Rank-k Approximation

Best Rank-k Approximation, $\boldsymbol{A}_k$, for $k = 10, 20, ..., 80, 90$

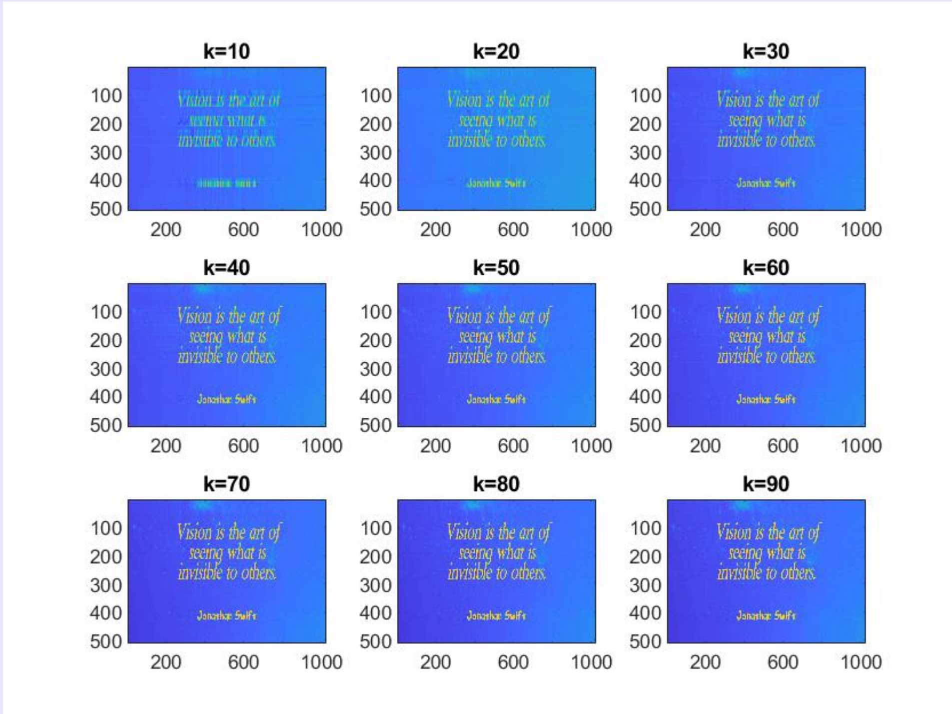

Figure 3: $\boldsymbol{A}_k$, for $k = 10, 20, ..., 80, 90$

(e) Storage Cost and Relative Error

Storage Cost and Relative Error

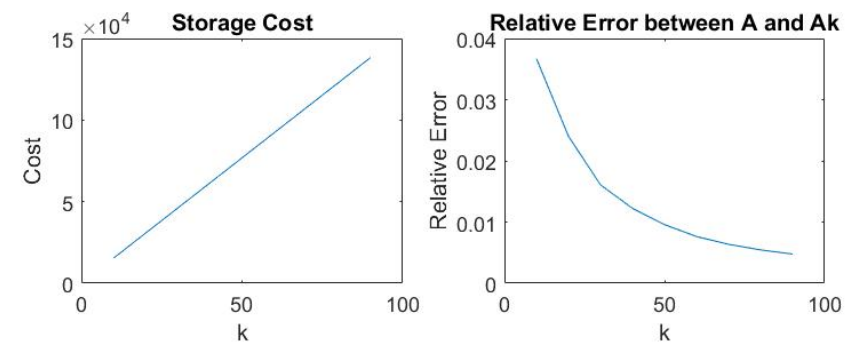

Figure 4: Storage Cost of $\boldsymbol{A}_k$ and Relative Error between $\boldsymbol{A}$ and $\boldsymbol{A}_k$

The storage cost is linearly proportional to k. More specifically the cost, $C(k) = (m+n+1)k$. The relative error should improve as more terms are added(the relative error decreases as k increases.)

(f) Best Rank-k Approximation

Best Rank-k Approximation, $\tilde{\boldsymbol{A}}_k$, for $k = 10, 20, ..., 80, 90$

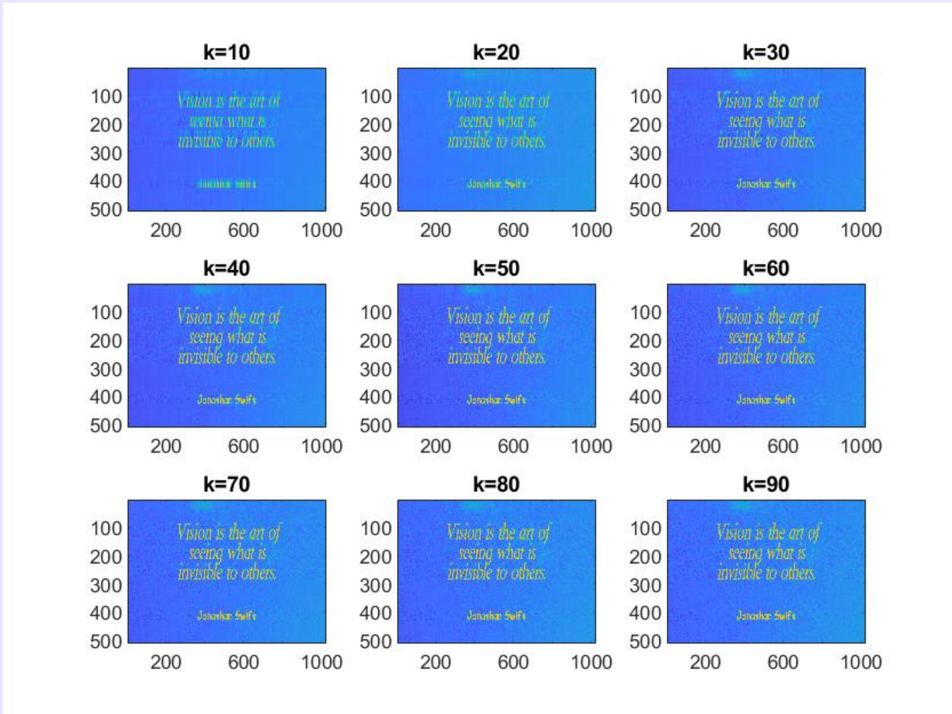

Figure 5: $\tilde{\boldsymbol{A}}_k$, for $k = 10, 20, ..., 80, 90$

(g) Relative Error

Relative Error

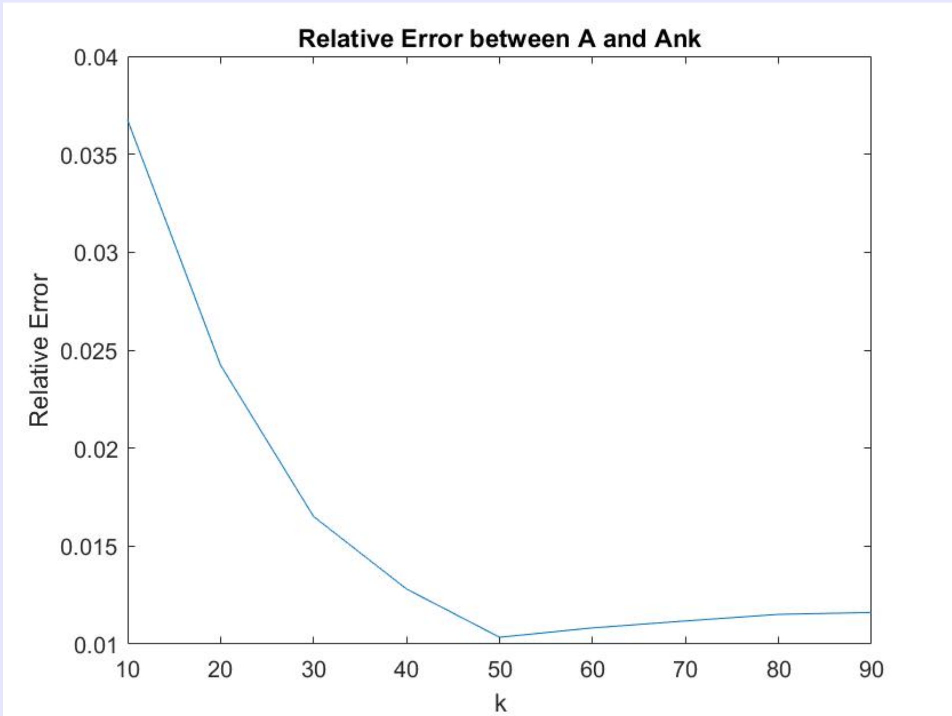

Figure 6: Relative Error between $\boldsymbol{A}$ and $\tilde{\boldsymbol{A}}_k$

The minimum occurs at approximately $k = 50$.

4) (15 points) *Deblurring* an image.

(a) (0 points) Load the file 'deblur.mat'. You will find the variables `A` (blurring operator, size 4096×4096) and `bn` (blurred and noisy image, size 4096×1), `xtrue` (true image, size 4096×1).

(b) (2 points) In a single figure with 2 subplots, plot the true image, and the blurry image with noise. Note that you will have to reshape the vectors into 64×64 images.

(c) (2 points) Recall the naive solution $\boldsymbol{x}_{\text{naive}} = \boldsymbol{A}^{-1}\boldsymbol{b}_{\text{noisy}}$. Plot this solution as an image. (MATLAB users should look up backslash \, and Python users should look up `numpy.linalg.solve`. Do not compute the inverse of the matrix!)

(d) (3 points) Compute the condition number $\kappa_2(\boldsymbol{A}) = \|\boldsymbol{A}\|_2\|\boldsymbol{A}^{-1}\|_2$ of the matrix A. Using perturbation analysis explain why you expect the naive solution to perform poorly (you are given that $\|\boldsymbol{e}\|_2/\|\boldsymbol{b}\|_2 = 0.05$).

(e) (3 points) Implement the truncated SVD formula

$$\boldsymbol{x}_k = \sum_{j=1}^{k} \boldsymbol{v}_j \frac{\boldsymbol{u}_j^\top \boldsymbol{b}_{\text{noisy}}}{\sigma_j},$$

for $k = 400, 800, \ldots, 3600$. In a single figure with 9 subplots, plot the reconstructed vectors $\boldsymbol{x}_k$ as images.

(f) (3 points) Plot the relative error in the reconstructed solution as a function of k. For (approximately) what value of k is the minimum attained?

(g) (2 points) In your words, explain the behavior of the error as a function of k.

Instructions: In total, you have to submit 4 separate plots. Make sure to label each plot/subplot, and label the axes of the error plots.

Solution:

(a) Loaded Deblur.m

(b) The True Image and the Blurry, Noisy Image

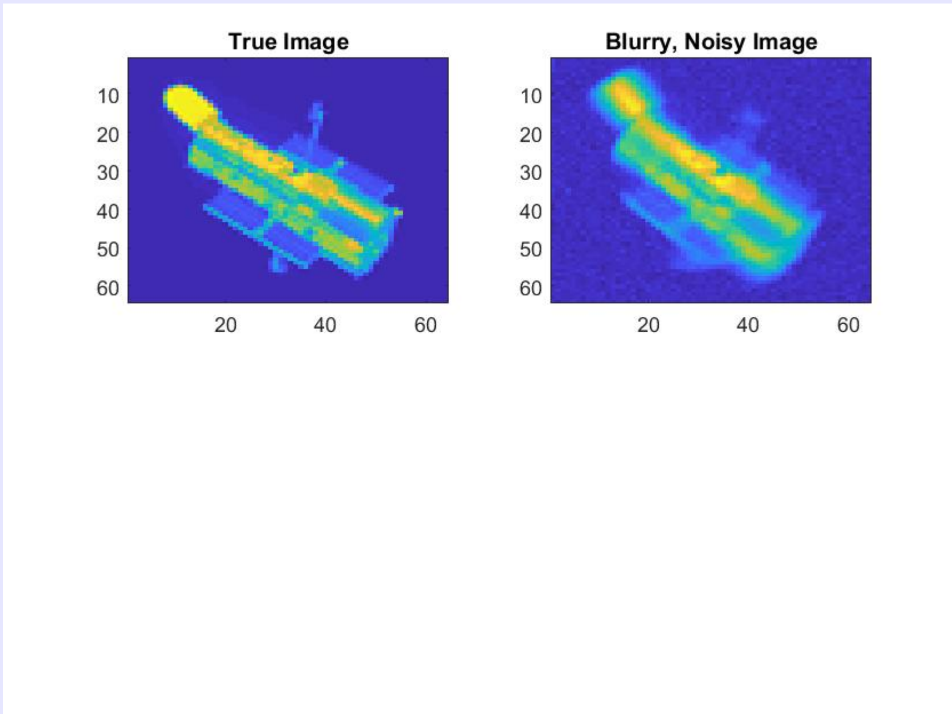

Figure 7: The True Image and the Blurry, Noisy Image

(c) The Naive Deblurring Solution

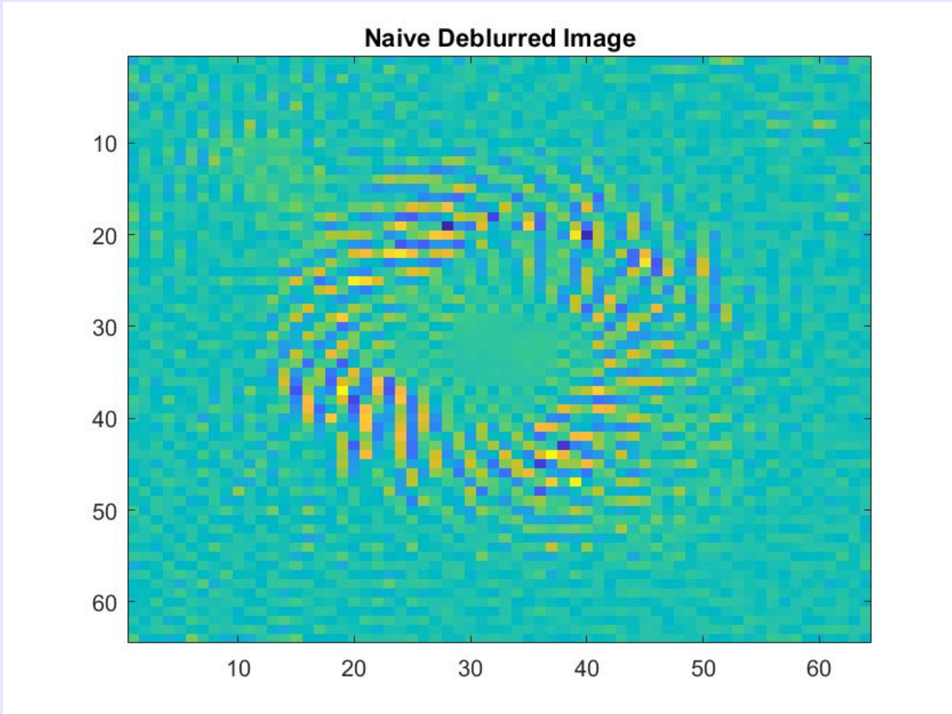

Figure 8: The Naive Deblurred Image

(d) The condition number $\kappa_2(\boldsymbol{A}) = 5.0399 * 10^3$.
The naive soultion fails because the noise is being amplified by the condition number. Then, the relative error in the naive solution satisfies:

$$\frac{\|\boldsymbol{x}_{\text{naive}} - \boldsymbol{x}_{\text{true}}\|_2}{\|\boldsymbol{x}_{\text{true}}\|} \leq \|\boldsymbol{A}\|_2 \|\boldsymbol{A}^{-1}\|_2 \frac{\|\boldsymbol{e}\|_2}{\|\boldsymbol{b}\|_2} = \kappa_2(\boldsymbol{A}) \frac{\|\boldsymbol{e}\|_2}{\|\boldsymbol{b}\|_2}$$

The condition number is much larger than 1. The relative error in the output is bounded by $5.0399 * 10^3 * 0.05 = 251.995$. The relative error in the output is large even for a small relative errror in the input. The blurring matrix is ill-conditioned.

(e) The Truncated SVD

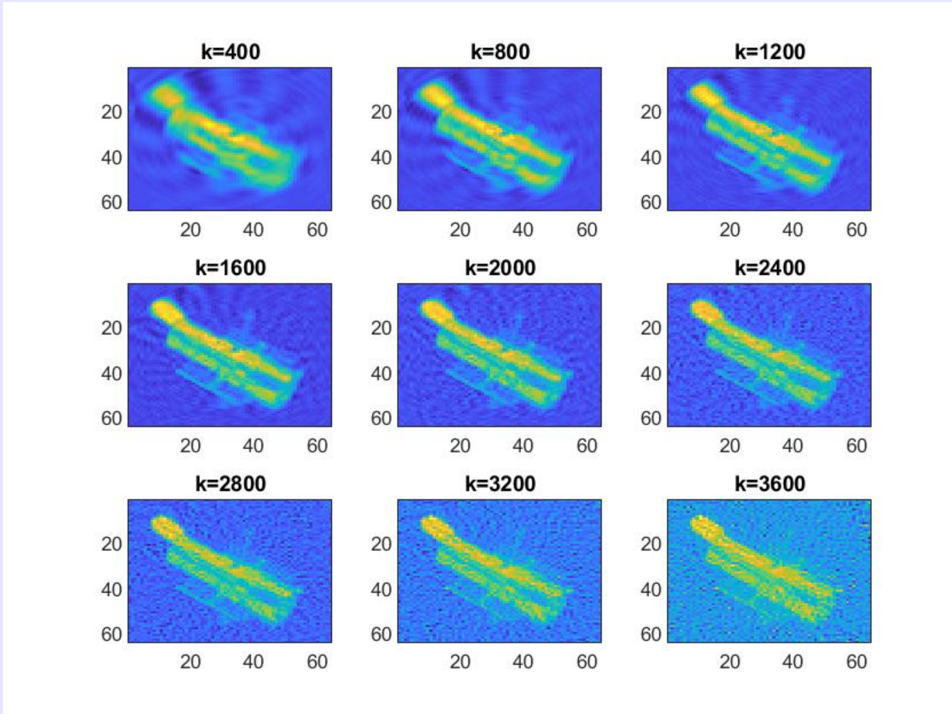

Figure 9: Truncated SVD

The minimum occurs at approximately $k = 50$.

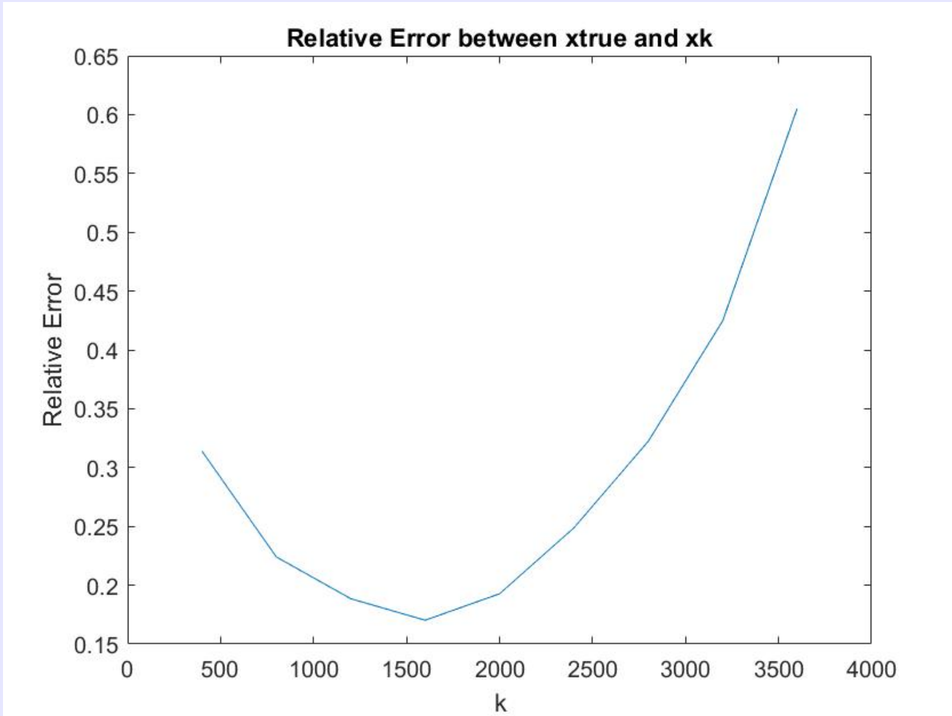

Figure 10: Truncated SVD

(f) The relative error is high for small values of k, reaches a minimum, and then becomes large again for large values of k. The noise becomes more apparent for small k values. For large k values, the terms of the truncated SVD are less important. Therefore, the images are less accurate.

Solution:

For Figure 1

```
%Clean and Noisy Images
subplot(2,2,1)
imagesc(A)
subplot(2,2,2)
imagesc(An)
```

For Figure 2

```
sigma = svd(A); %singular values of A
sigman = svd(An); %singular values of An
semilogy(sigma(1:100))
title('First 100 Singular values of A and An')
ylabel('Singular Values')
hold on
semilogy(sigman(1:100))
legend('A','An')
hold off
```