ORF 363/COS 323	Problem set 2	Fall 2024, Princeton University
Instructor: A.A. Ahmadi		
TAs: Aydin, Budway, Hua, Tziampazis		
Due on Thursday, September 26, 2024, at 1:30pm EST, on Gradescope		

For all problems that involve coding, please include your code. Our instructions are given with MATLAB in mind (recommended language), but you are free to use a different language.

Problem 1: Image compression and honoring Sydney McLaughlin[1]

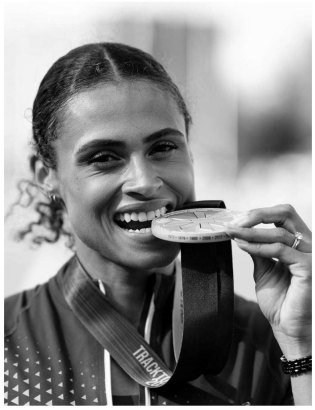

Figure 1: The image to be compressed.

If we were to rank the topics in computational linear algebra in terms of how frequently they arise in applications, *singular value decomposition* (SVD) would probably take one of the very top spots. In this problem, we introduce you to this concept and present one of its applications in image processing. The SVD also has applications in many other domains, notably in statistics.

Let A be a real $m \times n$ matrix of rank r. (Recall that the rank of A is the number of linearly independent columns of A.) The singular value decomposition of A is a decomposition of the form

$$A = U\Sigma V^T,$$

where U, Σ, and V are respectively $m \times m$, $m \times n$, and $n \times n$; U and V are orthogonal matrices (i.e., satisfy $U^TU = I$ and $V^TV = I$), and Σ is a matrix with r positive scalars $\sigma_1, \ldots, \sigma_r$ on

[1]Sydney Michelle McLaughlin-Levrone is the first track athlete to break six world records in the same event (in her case, 400 meters hurdles). Most recently, she set a world record time of 50.37 seconds at the 2024 Summer Olympics on August 8, 2024.

the diagonal of its upper left $r \times r$ block and zeros everywhere else. The scalars $\sigma_1, \ldots, \sigma_r$ are called the *singular values* of A. They are given as

$$\sigma_i = \sqrt{i\text{-th eigenvalue of } A^T A},$$

and by convention they appear in descending order:

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r.$$

The columns of U and V are respectively called the left and right *singular vectors* of A and can be obtained by taking an orthonormal set of eigenvectors[2] for the matrices AA^T and $A^T A$. In Matlab, the command `svd` handles these eigenvector computations for you and outputs the three matrices U, Σ, V.
First, you need to answer some very basic questions about the SVD.

1. Show that eigenvalues of $A^T A$ are always nonnegative. (Hence singular values are well-defined as real, nonnegative scalars.)

2. Show that if u_i and u_j are eigenvectors of $A^T A$ associated to distinct eigenvalues λ_i, λ_j, then u_i and u_j are orthogonal.

What does this have to do with image compression? Let A be a real $m \times n$ matrix with an SVD given by $A = U\Sigma V^T$ defined as above. For a positive integer $k \leq \min\{m, n\}$, we let $A_{(k)}$ denote an $m \times n$ matrix which is an "approximation" of the matrix A obtained from its top k singular values and singular vectors. Formally, we have

$$A_{(k)} := U_{(k)} \Sigma_{(k)} V_{(k)}^T,$$

where $U_{(k)}$ has the first k columns of U, $V_{(k)}$ has the first k columns of V, and $\Sigma_{(k)}$ is the upper left $k \times k$ block of Σ. A well-known result in optimization states that $A_{(k)}$ is a minimizer of the optimization problem

$$\min_{B \in \mathbb{R}^{m \times n}, rank(B) \leq k} ||A - B||_F.$$

Here, $||.||_F$ denotes the so-called Frobenius norm of a matrix defined as $||C||_F = \sqrt{\sum_{i,j} C_{i,j}^2}$. In words, the result states that among all $m \times n$ matrices of rank at most k, the matrix $A_{(k)}$ obtained from truncating the SVD is the one that best approximates A. The benefit

[2]By an orthonormal set of vectors we mean a collection of vectors that are pairwise orthogonal and each have 2-norm equal to one.

in approximating a matrix with low-rank matrices is that low-rank matrices admit a much more succinct representation. Let's see how this applies to our image compression problem.

Download the file `mclaughlin.jpg` into your Matlab path. You can read this in by typing:

```
A=imread('mclaughlin.jpg');
A=im2double(A);
A=rgb2gray(A);
```

If you are using Python, you may wish to use the package pillow and numpy:

```
from PIL import Image
import numpy as np
image = Image.open('mclaughlin.jpg')
A = np.array(image, dtype=float) / 255.0
A = np.dot(A, [0.2989, 0.5870, 0.1140])
```

The result is a 1007×768 matrix A, with each entry representing a single pixel in the picture with a number between 0 and 1. To upload this picture on Instagram, you would need to upload $1007 \times 768 = 773376$ numbers (pixels).

3. For $k = 10, 50, 90, 130$, use Matlab to compute $A_{(k)}$ as defined above. Report the value of $||A - A_{(k)}||_F$ in each case. (Include your code for this part and the next.)

4. Use the commands `subplot` and `imshow` to produce on the same figure the original image, as well as your compressed images $A_{(k)}$ for $k = 10, 50, 90, 130$. Label your subplots. In addition, produce two separate plots demonstraing (i) $||A - A_{(k)}||_F$ versus k, and (ii) "total savings" versus k. Total savings is to be interpreted as the answer to the question: How many fewer numbers do you need in order to store $A_{(k)}$ than you did to store A. Explain why this number is equal to $mn - (n + m + 1)k$. How much are you saving for $k = 130$?

5. Use the Matlab function `imwrite` and upload the two images `imshow(A)` and `imshow(`$A_{(130)}$`)` side by side on Instagram with the caption `#PrincetonORF363`. Ask your friends if they can tell which image is the original and which is the compressed version. (We are joking. There's no grade assigned to this part of the problem!)

Problem 2: Find all the local minimizers, local maximizers, global minimizers, and global maximizers of the following function over $\mathbb{R}^2$ (or argue if some do not exist):

$$f(x_1, x_2) = \frac{1}{2}x_1^2 + 4x_1x_2 + \frac{1}{2}x_2^2 - x_2^3.$$

Plot this function using Matlab and the command `ezsurf` and rotate the surface around with your mouse. Does the number of local minima and maxima that you found agree with what you see on the plot?

Problem 3: Properties of positive semidefinite matrices

Prove or disprove the following statements. All matrices are $n \times n$, symmetric, and real.

(a) Suppose $A \succeq 0$. Then the largest entry in absolute value of A must be on the diagonal.[3]

(b) If $A \succeq 0$ and $\text{trace}(A) = 0$, then $A = 0$.

(c) If $A \succeq 0, B \succeq 0$, and $A + B = 0$, then $A = B = 0$.

(d) If $A \succeq 0, B \succeq 0$, and $AB = 0$, then $A = 0$ or $B = 0$.

Problem 4: Copositive matrices

An $n \times n$ real symmetric matrix Q is said to be *copositive* if $x^TQx \geq 0$ for all $x \in \mathbb{R}^n$ such that $x \geq 0$. (The inequality on x is elementwise.)

(a) Prove that the set of $n \times n$ copositive matrices is convex. Show that the set of $n \times n$ noncopositive matrices is nonconvex unless $n = 1$.

(b) Give an example of a matrix that is copositive but neither positive semidefinite nor elementwise nonnegative. (You have to prove all claims about the example that you produce.)

[3] In other words, the largest entry of $|A|$ must be on the diagonal, where $|A|$ is the matrix whose entries are the absolute values of those of A.