This homework is due on Monday, December 4th, 2023 at 11:59PM. Self-grades and HW Resubmissions are due the following Monday, December 11th, 2023 at 11:59PM.

1. Congrats!

We are almost at the end of the course now! Your hard work is paying off. No admin questions this week. (OPTIONAL) Draw a funny picture that will make staff crack up.
2. PCA Introduction

Let \( X \in \mathbb{R}^{m \times n} \) be defined as \( X := [\vec{x}_1 \; \cdots \; \vec{x}_n] \) where each \( \vec{x}_i \in \mathbb{R}^m \). Let \( X \) have an SVD \( X = U \Sigma V^\top \).

Now, let \( U_\ell := [\vec{u}_1 \; \cdots \; \vec{u}_\ell] \) where \( \vec{u}_i \) is the \( i \)th column of \( U \). In other words, \( U_\ell \) is the first \( \ell \) columns of \( U \). In this problem, we will go about showing that

\[
U_\ell \in \arg\min_{W \in \mathbb{R}^{m \times \ell}} \sum_{i=1}^n \|\vec{x}_i - WW^\top \vec{x}_i\|^2
\]

where \( W^\top W = I_\ell \) (i.e., it is a matrix with orthonormal columns). This is an important result for deriving PCA, as you will see in lecture.

(a) First, show that

\[
\|\vec{x}_i - WW^\top \vec{x}_i\|^2 = \|\vec{x}_i\|^2 - \|W^\top \vec{x}_i\|^2
\]

(HINT: Expand the left hand side of the equation above using transposes. That is, use the fact that \( \|\vec{v}\|^2 = \vec{v}^\top \vec{v} \).)

(b) Using the result from the previous part, we can simplify the original optimization problem to say

\[
\arg\min_{W \in \mathbb{R}^{m \times \ell}} \sum_{i=1}^n \|\vec{x}_i - WW^\top \vec{x}_i\|^2 = \arg\min_{W \in \mathbb{R}^{m \times \ell}} \sum_{i=1}^n \left( \|\vec{x}_i\|^2 - \|W^\top \vec{x}_i\|^2 \right)
\]

\[
\iff \quad \arg\min_{W \in \mathbb{R}^{m \times \ell}} \sum_{i=1}^n \left( -\|W^\top \vec{x}_i\|^2 \right)
\]

\[
\iff \quad \arg\max_{W \in \mathbb{R}^{m \times \ell}} \sum_{i=1}^n \|W^\top \vec{x}_i\|^2
\]

where we get the second line from noticing that we cannot change \( \vec{x}_i \), so we remove it from the optimization problem. Then, we pull out the negative to turn the minimization problem into a maximization problem. Now, let \( W := [\vec{w}_1 \; \cdots \; \vec{w}_\ell] \). Show that

\[
\sum_{i=1}^n \|W^\top \vec{x}_i\|^2 = \sum_{k=1}^\ell \vec{w}_k^\top \Sigma \Sigma^\top \vec{w}_k
\]

You may use the fact that \( \sum_{i=1}^n \vec{x}_i \vec{x}_i^\top = XX^\top \). (HINT: Start by expanding out the norm squared expression as the sum of squares of the individual entries of \( W^\top \vec{x}_i \).)

(c) Use the result of the previous part to show that

\[
\sum_{i=1}^n \|W^\top \vec{x}_i\|^2 = \sum_{k=1}^\ell \vec{w}_k^\top \Sigma \Sigma^\top \vec{w}_k
\]

where \( \vec{w}_k = U^\top \vec{w}_k \). Then, argue that \( \Sigma \Sigma^\top \) can be written as

\[
\Sigma \Sigma^\top = \begin{bmatrix}
\sigma_1^2 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & 0
\end{bmatrix}
\]
where \( r = \text{rank}(X) \) (HINT: Use the SVD of \( X \) to simplify the \( XX^\top \) term from the previous part.)

(d) From the previous part, we have the following expression:

\[
\sum_{i=1}^{n} \left\| W^\top x_i \right\|^2 = \sum_{k=1}^{r} \bar{w}_k \begin{bmatrix}
\sigma_1^2 & \cdots \\
& \vdots \\
& 0
\end{bmatrix} \bar{w}_k
\tag{9}
\]

One may show (via Cauchy-Schwarz) that

\[
\sum_{k=1}^{r} \bar{w}_k \begin{bmatrix}
\sigma_1^2 & \cdots \\
& \vdots \\
& 0
\end{bmatrix} \bar{w}_k \leq \sum_{k=1}^{r} \sigma_k^2
\tag{10}
\]

if \( \bar{w}_k \) are required to be orthonormal (you are not required to show this). Using this fact, find some specific values of \( \bar{w}_k \) such that we attain eq. (10) with equality. Then, use this to show that \( U_k \) maximizes \( \sum_{i=1}^{n} \left\| W^\top x_i \right\|^2 \) and hence is a solution to the original optimization problem.
3. Rank 1 Decomposition and Image Compression With SVD

In this problem, we will introduce an important application of the singular value decomposition (SVD) called low-rank approximation in the context of image compression. Given a matrix $A \in \mathbb{R}^{m \times n}$ of rank $r \leq \min\{m, n\}$, we saw in Note 16 that we can write $A$ using the outer-product form of the SVD:

$$A = \sum_{i=1}^{r} \sigma_i \vec{u}_i \vec{v}_i^T. \quad (11)$$

If our matrix is high-rank, i.e., $r \approx \min\{m, n\}$, then almost all the $\sigma_i$ will be nonzero and non-negligible. However, if the data has some linear, low-rank essential structure, as is usually the case with real data such as images, most of our singular values will be very small (but usually nonzero due to noise or disturbances). If, say, the data has intrinsic linear rank $\ell$, then the first $\ell$ singular values are large, and the remaining $r - \ell$ are small:

$$A = \sum_{i=1}^{r} \sigma_i \vec{u}_i \vec{v}_i^T = \sum_{i=1}^{\ell} \sigma_i \vec{u}_i \vec{v}_i^T + \sum_{i=\ell+1}^{r} \sigma_i \vec{u}_i \vec{v}_i^T \approx \sum_{i=1}^{\ell} \sigma_i \vec{u}_i \vec{v}_i^T. \quad (13)$$

This motivates approximating the data as

$$A_\ell := \sum_{i=1}^{\ell} \sigma_i \vec{u}_i \vec{v}_i^T \quad (15)$$

and using this compressed data for further analysis. As images as readily represented as matrices, in this problem, we will see how to make use of this low rank approximation technique to compress images without severely deteriorating the image quality.

Throughout this problem we will work in the `image_compression.ipynb` Jupyter Notebook. Part 1 of the notebook relates to finding rank 1 decompositions of images, while Parts 2 and 3 focus on image compression. Make sure to read through all the sections of the notebook to understand how we will use what we know about the SVD in order to compress images! You should write all your answers in the written submission and you don’t need to submit the notebook.

We will start by decomposing a few images into linear combinations of rank 1 matrices. Remember that outer product of two vectors $\vec{s} \vec{g}^T$ gives a rank 1 matrix. It has rank 1 because the column span is one-dimensional — multiples of $\vec{s}$ only — and the row span is also one dimensional — multiples of $\vec{g}^T$ only. More generally, this exercise should provide intuition into how the outer product form of the SVD decomposes the underlying matrix into a sum of constituent rank 1 matrices with different weightings given by the singular values.

(a) Read and run through the cells in Part 1. Here we first consider a standard $4 \times 4$ checkerboard shown in Figure 1. Assume that black colors represent $-1$ and that white colors represent $1$. 
In particular, the checkerboard is given by the following $4 \times 4$ matrix $L$:

$$L = \begin{bmatrix}
1 & -1 & 1 & -1 \\
-1 & 1 & -1 & 1 \\
1 & -1 & 1 & -1 \\
-1 & 1 & -1 & 1 
\end{bmatrix}$$  \hspace{1cm} (16)

Express $L$ as a linear combination of outer products.

(HINT: In order to determine how many rank 1 matrices you need to combine to represent the matrix, find the rank of the matrix you are trying to represent.)

(b) We next consider the same checkerboard given by the following $4 \times 4$ matrix $C$ where black colors now represent 0 and that white colors represent 1.

$$C = \begin{bmatrix}
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 
\end{bmatrix}$$  \hspace{1cm} (17)

Express $C$ as a linear combination of outer products. (HINT: Note that $C$ is a rank 2 matrix and therefore you need 2 outer products to represent it. To find these individual outer products consider the columns of $C$.)

(c) Now consider the Swiss flag shown in Figure 2. Assume that red colors represent 0 and that white colors represent 1.

Figure 1: $4 \times 4$ checkerboard.

Figure 2: Swiss flag.
Assume that the Swiss flag is given by the following $5 \times 5$ matrix $S$:

$$S = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \quad (18)$$

Express $S$ as a linear combination of outer products.

(d) Run all the cells in Part 2. Thoroughly read through the `rank_k_approx` function. Since we are now only using the top $k$ singular values, in terms of $k$, $m$, and $n$, how many real numbers do we require to describe the rank $k$ approximation of $A$?

(e) Run the cells in Part 3 of the notebook. Notice there are two images: One compressed, one uncompressed. Can you easily visually detect any difference between the two images even though we are using only about 40% of the data?

(f) Play around with the slider for the kitten picture. What is the lowest acceptable value of $k$ that you can go without losing too much image quality? What were the memory savings?

*NOTE:* Use your best judgment since this depends on eyesight, screen resolution, etc.

(g) Plot the singular values for the image of the US flag. What do you notice about the singular values here in comparison to the kitten’s singular values? What does this mean for our low rank compression?

(h) Play around with the interactive slider for the US flag. What is the lowest acceptable value of $k$ here? What is the memory saving?
4. Movie Ratings and PCA

Recall from the lecture on PCA that we can think of movie ratings as a structured set of data. For every person \( i \) and movie \( j \), we have that person’s rating \( R_{i,j} \) (thought of as a real number).

Suppose that there are \( m \) movies and \( n \) people. Let’s think about arranging this data into a big \( n \times m \) matrix \( R \) with people corresponding to rows and movies corresponding to columns. So the entry in the \( i \)-th row and \( j \)-th column should be \( R_{i,j} \) above. Note that this is organized differently from how it was in lecture. Each row corresponds to a unique person and each column to a unique movie.

\[
R = \begin{bmatrix}
R_{11} & R_{12} & \cdots & R_{1m} \\
R_{21} & R_{22} & \cdots & R_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
R_{n1} & R_{n2} & \cdots & R_{nm}
\end{bmatrix}
\]  

(a) Suppose we believe that there is actually an underlying pattern to this rating data and that a separate study has revealed that every movie is characterized by a set of characteristics: say action and comedy. This means that every movie \( j \) has a pair of numbers \( a_j \) (for action) and \( c_j \) (for comedy) that define it. At the same time, every person \( i \) has a pair of sensitivities \( f_i \) and \( g_i \) that defines that person’s preferences for action vs. comedy movies respectively. A person \( i \) will rate the movie \( j \) as \( R_{i,j} = f_i a_j + g_i c_j \).

If we arrange the sensitivities into a pair of \( n \)-dimensional vectors \( \vec{f}, \vec{g} \) for our group of \( n \) people, and the movie characteristics into a pair of \( m \)-dimensional vectors \( \vec{a}, \vec{c} \) for our group of \( m \) movies, use outer products to express the rating matrix \( R \) in terms of these vectors \( \vec{f}, \vec{g}, \vec{a}, \vec{c} \).

(b) Now suppose that we want to discover the underlying nature of movies from the data \( R \) itself.

Suppose for this part, that you have four observed rating data vectors (corresponding to four different movies being rated by six individuals).

All of the movie data vectors just happened to be multiples of the following 6-dimensional vector

\[
\vec{\omega} = \begin{bmatrix}
2 \\
-2 \\
3 \\
-4 \\
4 \\
0
\end{bmatrix} \text{. (For your convenience, note that } ||\vec{\omega}|| = 7.)
\]

You arrange the movie data vectors as the columns of a matrix \( R \) given by:

\[
R = \begin{bmatrix}
\vec{\omega} & -2\vec{\omega} & 2\vec{\omega} & 4\vec{\omega}
\end{bmatrix}
\]

You want to perform PCA (for movies) using the SVD of the matrix \( R \) to better understand the pattern in your data.

The first “principal component vector” is a unit vector that tells which direction we would want to project the columns of \( R \) onto to get the best rank-1 approximation for \( R \).
Find this first principal component vector of the columns of $R$ to explain the nature of your movie data points.

(HINT: You don’t need to actually compute any SVDs in this simple case. Also, be sure to think about what size vector you want as the answer. Don’t forget that you want a unit vector!)

c) Suppose that now, we have two more data points (corresponding to two more movies being rated by the same set of six people, i.e. we added two columns to our matrix) that are multiples of a different vector $\vec{p}$ where:

$$\vec{p} = \begin{bmatrix} 6 \\ 3 \\ -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$  (For your convenience, note that $\|\vec{p}\| = 7$ and that $\vec{p}^\top \vec{w} = 0$.)

We augment our ratings data matrix with these two new data points to get:

$$R = \begin{bmatrix} -\vec{w} & -2\vec{w} & 2\vec{w} & 4\vec{w} & -3\vec{p} & 3\vec{p} \\ -2\vec{w} & 2\vec{w} & 4\vec{w} & -3\vec{p} & 3\vec{p} \end{bmatrix}$$  (21)

Find the first two principal components corresponding to the nonzero singular values of $R$. This is what we would use to best project the movie data points onto a two-dimensional subspace.

**What is the first principal component vector? What is the second principal component vector?** Justify your answer. (HINT: Think about the inner product of $\vec{w}$ and $\vec{p}$ and what that implies for being able to appropriately decompose $R$. Again, very little computation is required here.)

(d) In the previous part, you had $R = \begin{bmatrix} -\vec{w} & -2\vec{w} & 2\vec{w} & 4\vec{w} & -3\vec{p} & 3\vec{p} \\ -2\vec{w} & 2\vec{w} & 4\vec{w} & -3\vec{p} & 3\vec{p} \end{bmatrix}$ with $\|\vec{w}\| = 7$ and $\|\vec{p}\| = 7$, satisfying $\vec{p}^\top \vec{w} = 0$.

If we use $\vec{r}_i$ to denote the $i$-th column of $R$, plot the movie data points $\vec{r}_i$ (for all $i$) projected onto the first and second principal component vectors along the columns of $R$. The coordinate along the first principal component should be represented by horizontal axis and the coordinate along the second principal component should be the vertical axis. **Label each point, and the axes. Remember that principal component vectors are normalized.**
Contributors:
• Daniel Abraham.
• Ashwin Vangipuram.
• Tanmay Gautam.
• Nikhil Shinde.
• Nathan Lambert.
• Anant Sahai.
• Wahid Rahman.