Homework 11: Singular Value Decomposition

EECS 245, Fall 2025 at the University of Michigan

due Sunday, December 7th, 2025 at 11:59PM Ann Arbor Time **(no slip days!)**

> Write your solutions to the following problems by either typing them up or handwriting them on another piece of paper. Homeworks are due to Gradescope by 11:59PM on the due date. See the **syllabus** for details on the slip day policy.
>
> Homework will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should always explain and justify your conclusions, using sound reasoning. Your goal should be to convince the reader of your assertions. If a question does not require explanation, it will be explicitly stated.
>
> Before proceeding, make sure you're familiar with the **collaboration policy**.

Total Points: 10 + 18 + 22 + 15 = 65

Advice: You will have time to work on this homework in lab on Wednesday, December 3rd; try and come out of lab with Problems 2a, 2b, and 3 completed. Remember that you cannot use slip days on this homework. (We set the deadline two days later than originally planned.)

Problem 1: Homework 10 Solutions Review (10 pts)

Review **the solutions to Homework 10**. Pick **two problem parts** (for example, Problem 6b and Problem 7c) from Homework 10 in which your solutions have the most room for improvement, i.e. where they have unsound reasoning, could be significantly more efficient or clearer, etc. Include a screenshot of your solution to each problem part, and in a few sentences, explain what was deficient and how it could be fixed.

Alternatively, if you think one of your solutions is significantly better than the posted one, copy it here and explain why you think it is better. If you didn't do Homework 10, choose two problem parts from it that look challenging to you, and in a few sentences, explain the key ideas behind their solutions in your own words.

Problem 2: SVD Fundamentals (18 pts)

Before getting started, make sure to refer to Chapter 5.3. These problems aren't as computationally intensive as they look; think about ways to be efficient.

a) (4 pts) Let A be a 2×2 matrix with singular value decomposition $A = U\Sigma V^T$ where:

- The first column of U is $\vec{u}_1 = \begin{bmatrix} 2/\sqrt{5} \\ 1/\sqrt{5} \end{bmatrix}$.
- $A\vec{v}_1 = 3\vec{u}_1$, where $\vec{v}_1$ is the first column of V.
- The second singular value of A is $\sigma_2 = 1$.

Given this information, find U, Σ, and V^T.

b) (6 pts) Let $X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 2 & -1 \\ 2 & 2 \end{bmatrix}$.

(i) Compute the singular value decomposition (that is, find U, Σ, and V^T) for X. Do this by hand, but use `np.linalg.svd` in Python to verify your work.

(ii) Now, compute the singular value decomposition for $X^T = \begin{bmatrix} 1 & 0 & 2 & 2 \\ 0 & 1 & -1 & 2 \end{bmatrix}$. How can you reuse your work in finding the SVD of X to compute the SVD of X^T?

c) (4 pts) Compute the singular value decomposition for the diagonal matrix $X = \begin{bmatrix} 3 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -2 \end{bmatrix}$.

d) (4 pts) Compute the singular value decomposition for the rank-one matrix $X = \begin{bmatrix} 0 & 0 \\ 3 & 4 \\ 6 & 8 \end{bmatrix}$.

Hint: Can you write X as an outer product of two vectors? If you can, how do those vectors relate to the singular values and singular vectors of X?

Problem 3: Frobenius Norm and Low-Rank Approximation (22 pts)

As we first saw in Chapter 2.1, the norm of a vector is a measure of its size. The "default" norm is the Euclidean, or L_2 norm, $\|\vec{v}\|_2 = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$.

Similarly, the norm of a matrix is a measure of its size. The most common matrix norm is the **Frobenius norm**, defined as

$$\|X\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{d} x_{ij}^2}$$

That is, $\|X\|_F$ is the square root of the sum of the squares of the elements of X; it treats X as a vector and computes its L_2 norm.

a) (2 pts) Verify that $\|X\|_F = \sqrt{15}$ for $X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 2 & -1 \\ 2 & 2 \end{bmatrix}$.

*Notice that $\sqrt{15} = \sqrt{10+5}$, and in Problem 2a), you found that X's singular values were $\sigma_1 = \sqrt{10}$ and $\sigma_2 = \sqrt{5}$. We build on this idea in part **c)**.*

b) (4 pts) Another equivalent formula for the Frobenius norm is

$$\|X\|_F^2 = \text{trace}(X^T X)$$

where $\text{trace}(X^T X)$ is the sum of the diagonal entries of $X^T X$. (Notice the square on the left-hand side!) **Explain why** this is equivalent to the first definition of the Frobenius norm.

c) (4 pts) Another equivalent formula for the Frobenius norm is

$$\|X\|_F^2 = \sum_{i=1}^{r} \sigma_i^2$$

where $\sigma_1, \sigma_2, \ldots, \sigma_r$ are the singular values of X and $r = \text{rank}(X)$. **Explain why** this is equivalent to the definition of the Frobenius norm from part **b)**. *Hint: What is the relationship between the singular values of X and the eigenvalues of some other matrix?*

The Frobenius norm allows us to make more precise the idea of a rank-k approximation of a matrix, first introduced in **Chapter 5.3**.

Suppose $X = U\Sigma V^T$ is the singular value decomposition of the $n \times d$ matrix X, where the columns of U are $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_n \in \mathbb{R}^n$, the singular values of X are $\sigma_1, \sigma_2, \ldots, \sigma_r > 0$, the columns of V^T are $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_d \in \mathbb{R}^d$, and $r = \text{rank}(X)$.

The Eckart–Young–Mirsky theorem states that, for any integer k between 1 and r, the $n \times d$ matrix

$$X_k = \sum_{i=1}^{k} \sigma_i \vec{u}_i \vec{v}_i^T$$

is the closest rank-k matrix to X, in terms of Frobenius norm. That is, if Y is any other $n \times d$ matrix of rank k, then $\|X - X_k\|_F \leq \|X - Y\|_F$. More intuitively, this says that X_k is the matrix with the smallest mean squared error from X, among all $n \times d$ matrices of rank k. We will not prove this theorem in class.

d) (6 pts) Let's illustrate the above with an example. Consider the 3×4 matrix X, whose singular value decomposition is given by

$$\underbrace{\begin{bmatrix} 24 & 0 & 0 & 24 \\ 7 & 25 & 25 & 7 \\ 1 & -1 & 1 & -1 \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} 0.6 & 0.8 & 0 \\ 0.8 & -0.6 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 40 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 \\ 0 & 0 & 2 & 0 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} 1/2 & 1/2 & 1/2 & 1/2 \\ 1/2 & -1/2 & -1/2 & 1/2 \\ 1/2 & -1/2 & 1/2 & -1/2 \\ -1/2 & -1/2 & 1/2 & 1/2 \end{bmatrix}}_{V^T}$$

For $k = 1, 2, 3$, compute the rank-k approximation $X_k = \sum_{i=1}^{k} \sigma_i \vec{u}_i \vec{v}_i^T$ and the Frobenius norm of the approximation error, $\|X - X_k\|_F$.

Feel free to do the computations by hand or using `numpy`. If you use `numpy`, make sure to include screenshots of any code you write and its outputs, and **don't** use `np.linalg.svd`; instead, enter the SVD we provided you with and use `np.outer` to compute the outer product of two vectors.

e) (6 pts) Open the **the supplemental Jupyter Notebook** we've created for Homework 11, which can either be found **here** on DataHub, or **here** in the course GitHub repository.

There, you're asked to implement the rank-k approximation of an image of your choosing, similar to the **Image Compression example in Chapter 5.3**.
More instructions are provided in the notebook. This problem is **not autograded**. Rather, in your submission to this part, include screenshots of all of your code and outputs here.

Problem 4: Principal Components Analysis (15 pts)

Make sure you've completed Problem 3 before attempting this problem.

This problem involves a practical exploration of principal components analysis (PCA), perhaps the most interesting application of the singular value decomposition.

There are two ways to access the supplemental Jupyter Notebook:

- **Option 1**: Click **here** to open `hw11.ipynb` on DataHub. Before doing so, read the instructions on the **Tech Support** page on how to use the DataHub.
- **Option 2**: Set up a Jupyter Notebook environment locally, use `git` to clone our course repository, and open `homeworks/hw11/hw11.ipynb`. For instructions on how to do this, see the **Tech Support** page of the course website.

This problem is NOT autograded. Instead, complete the five tasks mentioned in Problem 4, and include screenshots of all of your code and outputs here, along with your written answers to Tasks 3 and 5.