**EE 516: Mathematical Foundations of Machine Learning** **Winter 2023**

# Homework 3

## Due: February 3, 2023, 11:59PM PT

*Student Name:* *Instructor Name: John Lipor*

**Problem 1** (5 pts each)

Let $A \in \mathbb{R}^{m \times n}$ and suppose $W \in \mathbb{R}^{m \times m}$ and $Q \in \mathbb{R}^{n \times n}$ are each orthogonal matrices.

(a) Show that $A$ and $WAQ$ have the same singular values. Consequently, $A$ and $WAQ$ have the same rank, Frobenius norm, and operator norm. This is why the Frobenius and operator norms are called *unitarily invariant*; their value does not change when the matrix is multiplied from the left and/or right by a unitary (or orthogonal) matrix. Any norm that depends on the singular values of $A$ will, by definition, be unitarily invariant.

(b) Suppose $W$ and $Q$ are nonsingular but not necessarily orthogonal matrices. Do $A$ and $WAQ$ still have the same rank? Prove or give a counterexample.

(c) In the setting of part (b), do $A$ and $WAQ$ have the same singular values? Prove or give a counterexample.

**Problem 2** (5 pts each)

Let $A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$.

(a) Determine the nullspace of $A$, denoted by $\mathcal{N}(A)$, and the range or column space of $A$, denoted by $\mathcal{R}(A)$.

(b) Are $\mathcal{N}(A)$ and $\mathcal{R}(A)$ equal? Is this true in general? If not, provide a counter-example.

**Problem 3** (7 pts, 3 pts)

(a) Determine how the eigenvalues of

$$B = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$$

are related to the singular values of $A \in \mathbb{R}^{n \times n}$.

(b) Verify that the eigenvectors of $B$ are the normalized versions of

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} \text{ and } \begin{bmatrix} u_i \\ -v_i \end{bmatrix},$$

where $u_i$ and $v_i$ are arbitrary left and right (respectively) singular vectors of $A$. **Hint:** To determine the eigenvalues, start off with some numerical experiments on self-generated $A$ matrices and form a conjecture. The commands `eig` and `svd` will be helpful. You should also write $A$ in terms of its SVD, i.e., $A = U\Sigma V^T$.

**Problem 4** (7 pts)

The *polar factorization* of a matrix $A \in \mathbb{R}^{n\times n}$ is given by $A = QS$, where $Q$ is orthogonal and $S = S^T \succeq 0$, i.e., $S$ is positive semi-definite. Express $Q$ and $S$ in terms of $U$, $\Sigma$, and $V$, where $A = U\Sigma V^T$.

**Problem 5** (3 pts each)

Let $A = xy^T$, where neither $x$ nor $y$ is 0.

(a) How many linearly-independent columns does $A$ have? **Explain why!**

(b) What is the rank of $A$? **Explain why!**

(c) Generate $A$ in MATLAB or PYTHON using random (`randn`) $x, y \in \mathbb{R}^{100}$. Plot the singular values of $A$ and include the plot here, as well as your code. The `stem` function works best for this.

**Problem 6** (5 pts, 2 pts)

When $D$ is an $m \times n$ rectangular diagonal matrix, its pseudo-inverse $D^+$ is an $n \times m$ rectangular diagonal matrix whose nonzero entries are the reciprocals $1/d_k$ of the diagonal entries of $D$. Thus, a matrix $A$ having SVD $A = U\Sigma V^T$ has $A^+ = V\Sigma^+U^T = V_r\Sigma_r^{-1}U_r^T$. **Working with just this definition**, determine by hand the pseudo-inverse of

(a) $A = xy^T$, where neither $x$ nor $y$ is 0

(b) $A = xx^T$, where $x \neq 0$.

You may again wish to from a conjecture using computational experiments.

**Problem 7** (7 pts)

Prove that the orthogonal complement of the range of a matrix $A$ equals the nullspace of $A^T$, i.e.,

$$\mathcal{R}(A)^\perp = \mathcal{N}(A^T).$$

**Hint:** A common strategy for proving that two sets $X, Y$ are equal is to prove that $X \subset Y$ and $Y \subset X$.

**Problem 8** (5 pts)

In this problem, you will implement another tool for use in the computer vision algorithm *photometric stereo*, which allows us to reconstruct a 3D object's surface from 2D images of it under different lighting conditions.

Suppose we are in a dark room with an object on a dark table, a camera fixed above it, and a moveable light source. We model the object surface as $z = f(x, y)$, where $(x, y)$ denotes the coordinates on the table and $z$ is the height above the table. Assume that the $m \times n$ sized image $I$ is a representation of $f(x, y)$ for each $(x, y)$ tuple. (Given one light source, $f(x, y)$ is the $z$ coordinate of the position where a ray of light hits the surface at $(x, y, z)$.) The pixel intensity $I(x, y)$ indicates how much light reflects off the surface $f(x, y)$. If our object is diffuse (also called matte or Lambertian), one can derive the relationship

$$I(x, y) = \alpha(x, y)(\ell^T n(x, y)), \qquad (1)$$

where $\ell \in \mathbb{R}^3$ is a unit-norm vector describing the orientation of the incident light rays on the surface, $n(x, y) \in \mathbb{R}^3$ is the unit-norm surface normal vector of $f$ at $(x, y, f(x, y))$, and $\alpha(x, y) > 0$ is a scaling constant called the surface albedo. Note that (1) is a *scalar* value.

Now suppose that we take $d$ images $I_1, \ldots, I_d$ of our object, with lighting directions $\ell_1, \ldots, \ell_d$. For any $(x, y)$, we can stack (1) into an overdetermined system of equations

$$\underbrace{\begin{bmatrix} I_1(x,y) \\ \vdots \\ I_d(x,y) \end{bmatrix}}_{=:I_{xy}} \approx \underbrace{\begin{bmatrix} \ell_1 & \ldots & \ell_d \end{bmatrix}^T}_{=:L^T} \underbrace{(\alpha(x,y) n(x,y))}_{=:\rho(x,y)}. \tag{2}$$

We could solve (2) for $\rho(x, y)$ when $d = 3$, but in practice when there is noise or our assumptions do not hold exactly (they never do), a more robust approach is to take $d > 3$ images and solve the least-squares problem

$$\rho(x, y) = \arg \min_{\rho \in \mathbb{R}^3} \left\| I_{xy} - L^T \rho \right\|_2^2 \tag{3}$$

and approximate the surface norm $n$ by

$$n^*(x, y) = \frac{\rho(x, y)}{\|\rho(x, y)\|_2}. \tag{4}$$

**Your task** for this problem is to complete the function called `compute_normals` (in files for download) that computes the unit-norm surface normal vectors for each pixel in a scene by solving (3) and (4). You may use the \ or `linalg.lstsq` command, where $A \backslash b$ computes the solution $x_{\mathrm{LS}} = \arg\min_x \|Ax - b\|_2^2$. **Turn in** your code for the implemented function and the resulting images generated by the `catdemo` script, which should match Fig. 1 below.

**Hint:** You can solve this for all $mn$ points simultaneously by vectorizing $I$.

**Problem 9 Reflection** (5 pts per problem)

In this problem, your task is to look back through Homeworks 1-2 and identify the **two** problems you found most difficult to solve and reflect on the problem solving strategies that got you through them. Complete the below tasks.

- For each problem, identify the top 1-3 tools/definitions from the course material you used.
- For each problem, write down the problem solving advice you would give to a peer attempting to solve the problem at the same point in the course. **Do not** simply give away the answer—place yourself in the role of an instructor during office hours.

Figure 1: Output images from Problem 8.