EECS 127/227AT Optimization Models in Engineering UC Berkeley Fall 2023

Homework 3

This homework is due at 11 PM on September 20, 2023.

Submission Format: Your homework submission should consist of a single PDF file that contains all of your answers (any handwritten answers should be scanned), as well as a printout of your completed Jupyter notebook(s).

1. Low-Rank Approximation

Let $A \in \mathbb{R}^{4\times 3}$ be a matrix whose full SVD is

$$A = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}}_{V^\top}. \tag{1}$$

Give the best rank-k approximation to A for the given value of k, with justification. These are the solutions to the problem

$$\underset{\substack{B\in\mathbb{R}^{4\times 3} \\ \mathrm{rk}(B)\le k}}{\operatorname{argmin}} \|A - B\|_F^2 . \tag{2}$$

NOTE: Please leave your answer in terms of a matrix product.

(a) $k = 1$

(b) $k = 2$

(c) $k = 3$

2. Interpreting the Data Matrix

When working in many fields, you'll often find yourself working with a *data matrix* X. Notation can vary — sometimes it has dimensions $\mathbb{R}^{m\times n}$, while others it has dimensions $\mathbb{R}^{n\times d}$, for example — and interpreting its precise meaning can often be confusing. In this problem, we lead you through an example of data matrix interpretation and manipulation.

First, what exactly is a data matrix? As the name suggests, it is a collection of *data points*. Suppose you are collecting data about courses offered in the EECS department in Fall 2022. Each course has certain quantifiable attributes, or *features*, that you are interested in. Possible examples of features are the number of students in the course, the number of GSIs in the course, the number of units the course is worth, the size of the classroom that the course was taught in, the difficulty rating of the course on a numerical (1-5) scale, and so on. Suppose there were a total of 20 courses, and that for each course, we have 10 features. This gives us 20 data points, where each data point is a 10-dimensional vector. We can arrange these data points in a matrix of size 20×10.

Generalizing the above, suppose we have n data points, with each point containing values for d features. Our data matrix X would then be of size $n \times d$, i.e., $X \in \mathbb{R}^{n\times d}$. We can interpret X in the following two (equivalent) ways:

$$X = \begin{bmatrix} \leftarrow \vec{x}_1^\top \rightarrow \\ \leftarrow \vec{x}_2^\top \rightarrow \\ \vdots \\ \leftarrow \vec{x}_n^\top \rightarrow \end{bmatrix} = \begin{bmatrix} \uparrow & \uparrow & \dots & \uparrow \\ \vec{f}_1 & \vec{f}_2 & \dots & \vec{f}_d \\ \downarrow & \downarrow & \dots & \downarrow \end{bmatrix}. \tag{3}$$

Here, $\vec{x}_i \in \mathbb{R}^d,\ i = 1, 2, \dots, n$, and $\vec{x}_i^\top$ is a row vector that contains values of different features for the i-th data point. Also, $\vec{f}_j \in \mathbb{R}^n,\ j = 1, 2, \dots, d$, and $\vec{f}_j$ is a column vector that contains values of the the j-th feature for different data points.

In the remainder of this problem, we explore how we can interpret and use X. For subproblems that require answers in Python, assume X is stored as a $n \times d$ NumPy array `X`.

(a) We first introduce the *empirical mean* of each feature. Let $k \geq 1$ be a positive integer, and define $\vec{1}$ to be the vector with 1 in every entry. The empirical mean of a vector $\vec{y} \in \mathbb{R}^k$ is defined as

$$\mu(\vec{y}) \doteq \frac{1}{k}\vec{1}^\top \vec{y} = \frac{1}{k}\sum_{i=1}^{k} y_i. \tag{4}$$

Suppose we want to compute a vector that contains the empirical mean of each feature, i.e., all the $\mu(\vec{f}_j)$'s. What is the length of the vector of empirical means? Which of the following Python commands will generate this vector?

i. `feature_means = numpy.mean(X, axis = 0)`

ii. `feature_means = numpy.mean(X, axis = 1)`

(b) The next quantity we will discuss is the *empirical variance*, and through it, the *empirical standard deviation*. The empirical variance of a vector $\vec{y} \in \mathbb{R}^k$ is defined as

$$\sigma^2(\vec{y}) \doteq \frac{1}{k}\left\|\vec{y} - \mu(\vec{y})\vec{1}\right\|_2^2 = \frac{1}{k}\sum_{i=1}^{k}(y_i - \mu(\vec{y}))^2. \tag{5}$$

As the choice of notation would have you expect, the empirical standard deviation is defined as

$$\sigma(\vec{y}) \doteq \sqrt{\sigma^2(\vec{y})}. \tag{6}$$

Suppose we want to compute a vector that contains the empirical standard deviation of each feature, i.e., all the $\sigma(\vec{f}_j)$'s. What is the length of this vector? Which of the following Python commands will generate this vector?

i. `feature_stddevs = numpy.std(X, axis = 0)`

ii. `feature_stddevs = numpy.std(X, axis = 1)`

(c) Suppose we want to modify `X` so that each feature vector is "*centered*", i.e., has zero empirical mean. How would you achieve this using Python code?

(d) Suppose we want to modify `X` so that each feature vector is "*standardized*", i.e., has zero empirical mean with empirical variance equal to 1. How would you achieve this using Python code?

NOTE: This standardization technique is a very common data pre-processing step.

(e) The last quantity we will discuss is the *empirical covariance*. For two vectors $\vec{w}, \vec{y} \in \mathbb{R}^k$, the empirical covariance is defined as

$$\sigma(\vec{w}, \vec{y}) \doteq \frac{1}{k}\left(\vec{w} - \mu(\vec{w})\vec{1}\right)^\top \left(\vec{y} - \mu(\vec{y})\vec{1}\right) = \frac{1}{k}\sum_{i=1}^{k}(w_i - \mu(\vec{w}))(y_i - \mu(\vec{y})). \tag{7}$$

What is $\sigma(\vec{y}, \vec{y})$ in terms of the empirical statistics we have previously defined (e.g. mean, variance, and/or standard deviation)?

(f) For the remainder of this problem, assume that the data matrix is centered, so every feature has zero empirical mean; that is, $\mu(\vec{f}_j) = 0$ for every j.

Let $\Sigma(X) \in \mathbb{R}^{d\times d}$ denote the *empirical covariance matrix* of X. This matrix contains the empirical covariance of each pair of feature vectors $(\vec{f}_i, \vec{f}_j)$. Correspondingly it is defined entry-wise as

$$\Sigma(X)_{i,j} \doteq \sigma(\vec{f}_i, \vec{f}_j). \tag{8}$$

Show that $\Sigma(X)$ can be represented in the following two ways:

$$\Sigma(X) = \frac{X^\top X}{n} \tag{9}$$

$$\Sigma(X) = \frac{1}{n}\sum_{i=1}^{n}\vec{x}_i\vec{x}_i^\top. \tag{10}$$

HINT: One (straightforward) way to show two matrices are equal is to show that for all i, j, their (i, j)-th entries are equal.

(g) In this class, we consider three different interpretations of the term "projection". We define them explicitly here for this problem.

Consider vectors $\vec{a}$ and $\vec{b}$ in $\mathbb{R}^n$. Let $\vec{b}$ be unit norm (i.e., $\left\|\vec{b}\right\|_2^2 = \vec{b}^\top\vec{b} = 1$). We define the following:

i. The **vector projection** of $\vec{a}$ on $\vec{b}$ is given by $(\vec{a}^\top\vec{b})\vec{b}$. The vector projection is a vector in $\mathbb{R}^n$.

ii. The **scalar projection** of $\vec{a}$ on $\vec{b}$ is given by $\vec{a}^\top\vec{b}$. The scalar projection is a scalar but can take both positive and negative values.

iii. The **projection length** of $\vec{a}$ on $\vec{b}$ is given by $\left|\vec{a}^\top\vec{b}\right|$ and is the absolute value of the scalar projection.

Suppose we want to obtain a column vector $\vec{z} \in \mathbb{R}^n$ whose i-th entry is the *scalar* projection of data point $\vec{x}_i$ along the direction given by the unit vector $\vec{u}$. Show that $\vec{z}$ is given by

$$\vec{z} = X\vec{u}. \tag{11}$$

(h) Performing this kind of projection onto a unit vector $\vec{u}$ is at the heart of the PCA computation, which also requires computing the *variance* of these scalar projections.

Formally, for $i \in \{1, \ldots, n\}$, define $z_i \doteq \vec{x}_i^\top \vec{u}$, and define $\vec{z} \doteq \begin{bmatrix} z_1, \ldots, z_n \end{bmatrix}^\top$. Show that its empirical variance $\sigma^2(\vec{z})$ can be calculated as

$$\sigma^2(\vec{z}) = \frac{1}{n} \vec{u}^\top X^\top X \vec{u} = \vec{u}^\top \Sigma(X) \vec{u}. \tag{12}$$

3. PCA and Senate Voting Data

In this problem, we consider a matrix of senate voting data, which we manipulate in Python. The data is contained in a $n \times d$ data matrix X, where each row corresponds to a senator and each column to a bill. Each entry of X is either $1, -1$ or 0, depending on whether the senator voted for the bill, against the bill, or abstained, respectively. Please compute your answers using the attached Jupyter notebook `senator_pca.ipynb`. The sub-parts of this problem can be answered in the notebook itself in the space provided and can be submitted as an attachment to this PDF using the "Download as PDF" feature that Jupyter Notebook supports.

(a) Suppose we want to assign a *score* to each senator based on their voting pattern, and then observe the empirical variance of these scores. To describe this, let us choose a $\vec{a} \in \mathbb{R}^d$ and a scalar $b \in \mathbb{R}$. We define the score for senator i as:

$$f(\vec{x}_i, \vec{a}, b) = \vec{x}_i^\top \vec{a} + b, \quad i = 1, 2, \ldots, n. \tag{13}$$

Note that $\vec{x}_i^\top$ denotes the i^{th} row of X and is a row vector of length d, as in the previous problem.

Let us denote by $\vec{z} = f(X, \vec{a}, b)$ the column vector of length n obtained by stacking the scores for each senator. Then

$$\vec{z} = f(X, \vec{a}, b) = X\vec{a} + b\vec{1} \in \mathbb{R}^n \tag{14}$$

where $\vec{1}$ is a vector with all entries equal to 1. Let us denote the mean value of $\vec{z}$ by $\mu(\vec{z}) = \frac{1}{n}\vec{1}^\top \vec{z}$. Let $\vec{\mu}(X) \in \mathbb{R}^d$ denote the vector containing the mean of each column of X. Then

$$\begin{aligned}
\mu(\vec{z}) &= \frac{1}{n}\sum_{i=1}^{n} z_i && (15)\\
&= \frac{1}{n}\sum_{i=1}^{n} \left(\vec{a}^\top \vec{x}_i + b\right) && (16)\\
&= \vec{a}^\top \left(\frac{1}{n}\sum_{i=1}^{n} \vec{x}_i\right) + b && (17)\\
&= \vec{a}^\top \vec{\mu}(X) + b && (18)
\end{aligned}$$

The empirical variance of the scores can then be obtained as

$$\begin{aligned}
\sigma^2(\vec{z}) &= \frac{1}{n}(\vec{z} - \mu(\vec{z})\vec{1})^\top (\vec{z} - \mu(\vec{z})\vec{1}) && (19)\\
&= \frac{1}{n}((X\vec{a} + b\vec{1}) - (\vec{a}^\top \vec{\mu}(X) + b)\vec{1})^\top ((X\vec{a} + b\vec{1}) - (\vec{a}^\top \vec{\mu}(X) + b)\vec{1}) && (20)\\
&= \frac{1}{n}(X\vec{a} + b\vec{1} - (\vec{a}^\top \vec{\mu}(X))\vec{1} - b\vec{1})^\top (X\vec{a} + b\vec{1} - (\vec{a}^\top \vec{\mu}(X))\vec{1} - b\vec{1}) && (21)\\
&= \frac{1}{n}(X\vec{a} - (\vec{a}^\top \vec{\mu}(X))\vec{1})^\top (X\vec{a} - (\vec{a}^\top \vec{\mu}(X))\vec{1}) && (22)\\
&= \frac{1}{n}(X\vec{a} - \vec{1}\vec{\mu}(X)^\top \vec{a})^\top (X\vec{a} - \vec{1}\vec{\mu}(X)^\top \vec{a}) && (23)\\
&= \frac{1}{n}((X - \vec{1}\vec{\mu}(X)^\top)\vec{a})^\top ((X - \vec{1}\vec{\mu}(X)^\top)\vec{a}) && (24)\\
&= \frac{1}{n}\vec{a}^\top (X - \vec{1}\vec{\mu}(X)^\top)^\top (X - \vec{1}\vec{\mu}(X)^\top)\vec{a}. && (25)
\end{aligned}$$

Note that this variance is therefore a function of the "centered" data matrix $X - \vec{1}\vec{\mu}(X)^\top$ in which the mean of each column is zero. It also does not depend on b.

For the remainder of this problem, we assume that the data has been pre-centered (i.e., $\vec{\mu}(X) = \vec{0}$); note that this has been pre-computed for you in the code notebook. Assume also that $b = 0$, so that $\mu(\vec{z}) = 0$. Defining $f(X, \vec{a}) \doteq f(X, \vec{a}, 0)$ and replacing $\vec{z}$ with $f(X, \vec{a})$, we can then write the simpler empirical variance formula

$$\sigma^2(f(X, \vec{a})) = \frac{1}{n}\vec{a}^\top X^\top X \vec{a}. \tag{26}$$

Suppose we restrict $\vec{a}$ to have unit-norm. In the provided code, find the maximum empirical variance $\sigma^2(f(X, \vec{a}))$ over all unit-norm $\vec{a}$, and find the $\vec{a}$ that maximizes it.

(b) We next consider party affiliation as a predictor for how a senator will vote. Follow the instructions in the notebook to compute the mean voting vector for each party and relate it to the direction of maximum variance.

(c) Recall from problem 1 that given a vector $\vec{z} = X\vec{u}$ (i.e., the vector of scalar projections of each row of X along $\vec{u}$), we can compute its empirical variance as

$$\sigma^2(\vec{z}) = \vec{u}^\top \Sigma \vec{u}, \tag{27}$$

where $\Sigma(X) = \frac{X^\top X}{n}$ is the empirical covariance matrix of X. We will show in a future homework problem that the variance along each principal component $\vec{a}_i$ is precisely its corresponding eigenvalue of $\Sigma(X)$, i.e., $\lambda_i\{\Sigma(X)\}$. (For now, just note that this fact should make intuitive sense, since PCA is searching for directions of maximum variance of the data, and these occur along the covariance matrix's eigenvectors.) In the Notebook, compute the sum of the variance along $\vec{a}_1$ and $\vec{a}_2$ and plot the data projected on the $\vec{a}_1$–$\vec{a}_2$ plane.

(d) Suppose we want to find the bills that are most and least contentious — i.e., those that have high variability in senators' votes, and those for which voting was almost unanimous. Follow the instructions in the Jupyter notebook to compute a measure of "contentiousness" for each bill, plot the vote counts for exemplar bills, and comment on the voting trends.

(e) Suppose we want to infer the political affiliations of two senators whose voting records are known to us. Follow the instructions in the Jupyter notebook to infer the political affiliation of the Green and Grey colored senators using PCA.

(f) Finally, we can use the defined score $f(X, \vec{a}, b)$, computed along the first principal component $\vec{a}_1$ to classify the most and least "extreme" senators based on their voting record. Follow the instructions in the Jupyter Notebook to compute these scores and comment on their relationship to partisan affiliation.

4. FTLA, SVD, Pseudoinverse, and Least-Squares

Let $A \in \mathbb{R}^{m \times n}$ be a matrix, and let $\vec{y} \in \mathbb{R}^m$. Let $r \doteq \text{rank}(A)$, and let

$$A = U\Sigma V^\top = \begin{bmatrix} U_r & U_{m-r} \end{bmatrix} \begin{bmatrix} \Sigma_r & 0_{r\times(n-r)} \\ 0_{(m-r)\times r} & 0_{(m-r)\times(n-r)} \end{bmatrix} \begin{bmatrix} V_r^\top \\ V_{n-r}^\top \end{bmatrix} = U_r \Sigma_r V_r^\top = \sum_{i=1}^{r} \sigma_i \vec{u}_i \vec{v}_i^\top \tag{28}$$

be an SVD of A. In this problem, we will prove some properties about the SVD, and then apply them to derive a unique solution to an optimization problem which generalizes minimum-norm and least-squares.

(a) Prove that $\mathcal{R}(A) = \mathcal{R}(U_r)$ and that $\mathcal{R}\left(A^\top\right) = \mathcal{R}(V_r)$.

(b) Use the fundamental theorem of linear algebra to prove that $\mathcal{N}(A) = \mathcal{R}(V_{n-r})$ and $\mathcal{N}\left(A^\top\right) = \mathcal{R}(U_{m-r})$.

Now, we will use the SVD to derive the unique solution to the *least-norm least-squares* problem. That is, we want to use the SVD to find the unique vector in $\mathbb{R}^n$ which solves the least-squares problem with minimum norm. More formally, we wish to solve the least-norm least-squares problem:

$$\min_{\vec{x} \in S} \|\vec{x}\|_2^2 \qquad \text{where} \qquad S \doteq \operatorname*{argmin}_{\vec{x} \in \mathbb{R}^n} \|A\vec{x} - \vec{y}\|_2^2 . \tag{29}$$

In words, we want to find the minimum-norm vector $\vec{x}$ in the set of all least-squares solutions S.[a] This problem generalizes both the traditional least-squares and minimum-norm problems.

To solve the least-norm least-squares problem, we will use the SVD to define a concept called the *pseudoinverse*. The matrix $A^\dagger \in \mathbb{R}^{n \times m}$ is called a *pseudoinverse* (sometimes a *Moore-Penrose pseudoinverse*) of A, where

$$A^\dagger = V\Sigma^\dagger U^\top = \begin{bmatrix} V_r & V_{n-r} \end{bmatrix} \begin{bmatrix} \Sigma_r^{-1} & 0_{r\times(m-r)} \\ 0_{(n-r)\times r} & 0_{(n-r)\times(m-r)} \end{bmatrix} \begin{bmatrix} U_r^\top \\ U_{m-r}^\top \end{bmatrix} = V_r \Sigma_r^{-1} U_r^\top = \sum_{i=1}^{r} \frac{1}{\sigma_i} \vec{v}_i \vec{u}_i^\top . \tag{30}$$

In this problem, we will show that $A^\dagger \vec{y}$ is the unique solution to the least-norm least-squares problem (29).

NOTE: In this problem, we *do not* assume that A has full column rank or full row rank — in fact, A could even be the matrix of all zeros — so the least-squares solution we learn in class does not apply.

[a] Here, the argmin is a set – that is, it is the set of minimizers of the least-squares objective. A simpler example of this notation is that if $f(x) \doteq 0$ for all $x \in \mathbb{R}$, then $\text{argmin}_{x \in \mathbb{R}} f(x) = \mathbb{R}$. More information about this notation is contained in section 1.3 of the course reader.

(c) Show that $\vec{x} \in S$ if and only if $A^\top A\vec{x} = A^\top \vec{y}$ (these are the so-called *normal equations*, which you may remember from our discussion on least-squares).

(d) Show that $S = \{A^\dagger \vec{y} + \vec{z} \mid \vec{z} \in \mathcal{N}(A)\}$.

HINT: First, show that $A^\dagger \vec{y} \in S$. Then show that for any two $\vec{x}_1, \vec{x}_2 \in S$, that $\vec{x}_1 - \vec{x}_2 \in \mathcal{N}(A)$. Argue that this implies the conclusion.

(e) Now show that $\{A^\dagger \vec{y}\} = \text{argmin}_{\vec{x} \in S} \|\vec{x}\|_2^2$.

HINT: Pick any $\vec{z} \in \mathcal{N}(A)$. Show that $A^\dagger \vec{y}$ is orthogonal to $\vec{z}$. Then show that $\left\|A^\dagger \vec{y}\right\|_2^2 \leq \left\|A^\dagger \vec{y} + \vec{z}\right\|_2^2$, with equality if and only if $\vec{z} = \vec{0}$. Argue that this implies the conclusion.

5. Properties of the Frobenius Norm

The Frobenius norm of a matrix A is defined as

$$\|A\|_F = \sqrt{\langle A,\ A\rangle} = \sqrt{\sum_{\substack{1\le i\le m\\ 1\le j\le n}} A_{ij}^2} \tag{31}$$

where for two matrices $A, B \in \mathbb{R}^{m\times n}$, the canonical inner product defined over this space is $\langle A,\ B\rangle := \operatorname{tr}\left(A^\top B\right) = \sum_{i,j} A_{ij}B_{ij}$. The previous definition of the inner product is equivalent to interpreting the matrices A and B as vectors of length mn and taking the vector inner product of the respective mn-dimensional vectors. The Cauchy-Schwarz inequality for the inner product follows in a straightforward way from the Cauchy-Schwarz inequality for vectors:

$$\langle A,\ B\rangle = \sum_{\substack{1\le i\le m\\ 1\le j\le n}} A_{ij}B_{ij} \le \left(\sum_{\substack{1\le i\le m\\ 1\le j\le n}} A_{ij}^2\right)^{1/2} \left(\sum_{\substack{1\le i\le m\\ 1\le j\le n}} B_{ij}^2\right)^{1/2} = \|A\|_F \|B\|_F . \tag{32}$$

(a) Show that the Frobenius norm satisfies all three properties of a norm.

HINT: The easiest way to do this problem is to ***not*** *look at the individual components A_{ij}, but instead use the inner product formulation $\|A\|_F = \sqrt{\langle A,\ A\rangle}$.*

(b) Write the Frobenius norm squared in terms of singular values.

HINT: The cyclic property of traces might be helpful: $\operatorname{tr}(ABC) = \operatorname{tr}(CAB) = \operatorname{tr}(BCA)$.

(c) Express the Frobenius norm squared in terms of the ℓ_2-norm of the columns of A with $\vec{a}_i$ denoting column i. Concretely, prove $\|A\|_F^2 = \sum_{i=1}^n \|\vec{a}_i\|_2^2$ where $\vec{a}_i$ are the columns of A.

(d) A generalization of the least squares problem is to find a *matrix* X that most closely solves the problem $AX = B$. This is sometimes called the *matrix least squares problem*, and when X and B are vectors, reduces to the ordinary least squares problem you are familiar with. Formally, we can define the problem given $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{m\times k}$:

$$\min_{X\in\mathbb{R}^{n\times k}} \|AX - B\|_F \tag{33}$$

However, at this point we only know how to solve the vector least squares problem. Re-formulate the above objective in terms of vector least squares problems that we would know how to solve.

HINT: The result derived in part (c) may be particularly useful in addition to the following fact:

$$\min_{\vec{x},\vec{y}}\{f(\vec{x}) + g(\vec{y})\} = \min_{\vec{x}} f(\vec{x}) + \min_{\vec{y}} g(\vec{y}). \tag{34}$$

6. Matrix Norm Calculations

Let $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$.

(a) Compute $\|A\|_F$, the Frobenius norm of A.

(b) Compute $\|A\|_2^2$, the *squared* spectral norm of A.

7. Homework Process

With whom did you work on this homework? List the names and SIDs of your group members.

NOTE: If you didn't work with anyone, you can put "none" as your answer.