EECS 127/227AT Optimization Models in Engineering
Spring 2020 Homework 3

This homework is due Friday, February 14, 2020 at 23:00 (11pm).
Self grades are due Friday, February 21, 2020 at 23:00 (11pm).

This version was compiled on 2020-02-08 08:10.

Questions marked (Practice) will not be graded.

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 your IPython notebook saved as a PDF.

1. (Practice) Interpreting the data matrix

When working in many fields, including machine learning, statistics, and any requiring general scientific data analysis, 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 $\mathbb{R}^{n\times m}$, for example — and interpreting its precise meaning can often be confusing. In this problem, we lead you through an example of data matrix interpretation.

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 2018. 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, where each row corresponds to values of different features for the same point, and each column corresponds to values of same feature for different points.

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

$$X = \begin{bmatrix} \leftarrow \vec{x}_1^\top \rightarrow \\ \leftarrow \vec{x}_2^\top \rightarrow \\ \vdots \\ \leftarrow \vec{x}_n^\top \rightarrow \end{bmatrix} \tag{1}$$

Here, $\vec{x}_i \in \mathbb{R}^m,\ i = 1, 2, \ldots, n$, and $\vec{x}_i^\top$ is a row vector that contains values of different features for the i^{th} data point. Second,

$$X = \begin{bmatrix} \uparrow & \uparrow & \cdots & \uparrow \\ \vec{x}^1 & \vec{x}^2 & \cdots & \vec{x}^m \\ \downarrow & \downarrow & \cdots & \downarrow \end{bmatrix} \tag{2}$$

Here $\vec{x}^j \in \mathbb{R}^n,\ j = 1, 2, \ldots, m$, and $\vec{x}^j$ is a column vector that contains values of the the j^{th} feature for different data points[1].

Consider the matrix X as described above. In the remainder of this problem, we explore how we can manipulate the data matrix to get some desirable properties. For subproblems that require answers in Python, assume X is stored as a $n \times m$ NumPy array `X`.

(a) Suppose we want to compute a vector that contains the *mean value* of each feature. What is the length of this vector? 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) Suppose we want to compute a vector that contains the *standard deviation* of each feature. 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., zero 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., zero mean with unit variance. How would you achieve this using Python code?

(e) Another metric of interest is *covariance*, which when computed between two sets of values, tells us how similar those sets of values are. You may have studied covariance in the context of probability; from a linear algebra perspective, we can view the empirical covariance as the *inner product* between two de-meaned vectors:

$$\operatorname{cov}(\vec{y}_1, \vec{y}_2) = \frac{\langle \vec{y}_1 - \vec{\mu}_{\vec{y}_1}, \vec{y}_2 - \vec{\mu}_{\vec{y}_2} \rangle}{k}, \quad y_1, y_2 \in \mathbb{R}^k$$

where $\vec{\mu}_{\vec{y}_i}$ is the expected value of vector $\vec{y}_i, i = 1, 2$. This makes sense, as the inner product between two vectors is also a measure of their similarity.

In our data matrix, we are not too interested in the relationship between data points, but knowing how the *features* are related is core to many statistical methods. Write an expression that computes the covariance between the i^{th} and j^{th} feature. In the next part, we will then consolidate the covariances of all pairwise features into a covariance matrix. What will be the size of this matrix? *Hint*: Note that in our above definition of covariance, k is the dimension of the covarying vectors, which in our case corresponds to the number of data points collected for each feature.

(f) For the remainder of this problem, assume that the data matrix is centered, so every feature is zero mean. Let C denote the covariance matrix. Show that C can be represented in the following two ways:

$$C = \frac{X^\top X}{n}$$

$$C = \frac{1}{n} \sum_{i=1}^{n} \vec{x}_i \vec{x}_i^\top .$$

[1] Note that you will sometimes encounter notation where the columns are referred to as $\vec{x}_1, \vec{x}_2, \ldots,$ instead of using superscripts as above, but it is important to understand the context and be clear on what columns and rows represent.

Recall that $\vec{x}_i^\top$ is the i^{th} row of X.
Hint: One (straightforward) way to show two matrices are equal is to show that for all i, j, their $(i, j)^{\text{th}}$ entries are equal.

(g) In this class, we consider three different interpretations of the term "projection"; these are often used interchangeably, and corresponding notation is often abused, which can make it confusing at times, so we define them explicitly here.

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

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

ii. The **scalar projection** of $\vec{a}$ on $\vec{b}$ is given $\vec{a}^\top \vec{b}$. 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 $|\vec{a}^\top \vec{b}|$ and is the absolute value of the scalar projection.

Recall that our data points, $\vec{x}_i$ are contained in the rows of X. 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

$$\vec{z} = X\vec{u}$$

(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. (In fact, we're looking for the "principal components", i.e., directions along which the variance of these scalar projections is maximized!) Let Z be a random variable corresponding to the scalar projection of data points along direction $\vec{u}$. We now treat the entries of $\vec{z}$ (i.e., $z_1, z_2, \ldots, z_n$) as samples of Z. (Here, note that the randomness in Z comes from the data points. The direction $\vec{u}$ is fixed.) Show that the empirical variance $\sigma_{\vec{z}}^2$ of Z can be calculated as

$$\sigma_{\vec{z}}^2 = \frac{1}{n}\vec{u}^\top X^\top X\vec{u} = \vec{u}^\top C\vec{u}.$$

Hint: The empirical variance is given by $\sigma_{\vec{z}}^2 = \frac{1}{n}\sum\limits_{i=1}^{n}(z_i - \mu_{\vec{z}})^2$, where $\mu_{\vec{z}} = \frac{1}{n}\sum\limits_{i=1}^{n} z_i$ is the empirical mean. Recall that X is assumed to be centered.

2. 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 $n \times m$ 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_qns.ipynb`.

(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 $\vec{a} \in \mathbb{R}^m$ and a scalar $b \in \mathbb{R}$. We define the score for senator i as:

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

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

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

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

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 $\vec{\mu}_X^\top \in \mathbb{R}^m$ denote the row vector containing the mean of each column of X. Then

$$\begin{aligned}\mu_{\vec{z}} &= \frac{1}{n}\sum_{i=1}^{n} f(\vec{x}_i, \vec{a}, b)\\ &= \vec{a}^\top \vec{\mu}_X + b\end{aligned}$$

The empirical variance of the scores can then be obtained as

$$\begin{aligned}\operatorname{var}(f(X, \vec{a}, b)) &= \operatorname{var}(\vec{z})\\ &= \frac{1}{n}(\vec{z} - \mu_{\vec{z}}\vec{1})^\top(\vec{z} - \mu_{\vec{z}}\vec{1})\\ &= \frac{1}{n}(X\vec{a} + b\vec{1} - \vec{a}^\top\vec{\mu}_X\vec{1} - b\vec{1})^\top(X\vec{a} + b\vec{1} - \vec{a}^\top\vec{\mu}_X\vec{1} - b\vec{1})\\ &= \frac{1}{n}(X\vec{a} - \vec{1}\vec{\mu}_X^\top\vec{a})^\top(X\vec{a} - \vec{1}\vec{\mu}_X^\top\vec{a})\\ &= \frac{1}{n}\vec{a}^\top(X - \vec{1}\vec{\mu}_X^\top)^\top(X - \vec{1}\vec{\mu}_X^\top)\vec{a}\end{aligned}$$

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 $f(X, \vec{a}) \doteq f(X, \vec{a}, 0)$, we can then write simpler variance formula

$$\operatorname{var}(f(X, \vec{a})) = \frac{1}{n}\vec{a}^\top X^\top X\vec{a}$$

Suppose we restrict $\vec{a}$ to have unit-norm. In the provided code, find $\vec{a}$ that maximizes $\operatorname{var}(f(X, \vec{a}))$. What is the value of the maximum variance?

(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 variance as

$$\operatorname{var}(\vec{z}) = \vec{u}^\top C\vec{u},$$

where $C = \frac{X^\top X}{n}$. We will show in a future homework problem that the variance along each principal component $\vec{a}_i$ is precisely its corresponding eigenvalue of C, $\lambda_i(C)$. (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 several possible measures of "contentiousness" for each bill, plot the vote counts for exemplar bills, and comment on the metrics' relationship to each other.

(e) Finally, we can use the defined score $f(X, \vec{a}, b)$, computed along 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 affliation.

3. Interpretation of the covariance matrix

Suppose we are given m data points $\vec{x}_1, \ldots, \vec{x}_m$ in $\mathbb{R}^n$. Let $\hat{\vec{x}} \in \mathbb{R}^n$ denote the sample average of the points:

$$\hat{\vec{x}} \doteq \frac{1}{m}\sum_{i=1}^{m} \vec{x}_i.$$

Given a normalized direction vector $\vec{w} \in \mathbb{R}^n$ with $\|\vec{w}\|_2 = 1$, we consider the line with direction $\vec{w}$ passing through the origin: $\mathcal{L}(\vec{w}) \doteq \{t\vec{w} : t \in \mathbb{R}\}$. We then consider the projection of the points $\vec{x}_i$, $i = 1, \ldots, m$ onto the line $\mathcal{L}(\vec{w})$. For example, the projection of point $\vec{x}_i$ onto the line $\mathcal{L}(\vec{w})$ is given by $t_i(\vec{w})\vec{w}$, where

$$t_i(\vec{w}) = \arg\min_t \|t\vec{w} - \vec{x}_i\|_2.$$

Note that the minimizer $t_i(\vec{w})$ is given by $\vec{w}^\top \vec{x}_i$, i.e., the scalar projection of $\vec{x}_i$ along direction $\vec{w}$. (If you don't remember why this is true, use this opportunity to refresh your memory on the proof. Alternatively, you can calculate this value by solving the optimization problem directly using vector calculus tools.)

(a) For any $\vec{w}$, let $\hat{t}(\vec{w})$ denote the sample average of $t_i(\vec{w})$:

$$\hat{t}(\vec{w}) = \frac{1}{m}\sum_{i=1}^{m} t_i(\vec{w}),$$

Assume that $\hat{t}(\vec{w})$ is constant — i.e., it is independent of the direction of $\vec{w}$. Show that the sample average of the data points, $\hat{\vec{x}}$, is zero.

(b) We will now consider the data's covariance matrix

$$\Sigma \doteq \frac{1}{m}\sum_{i=1}^{m} (\vec{x}_i - \hat{\vec{x}})(\vec{x}_i - \hat{\vec{x}})^\top.$$

and show that it provides a powerful visualization of where the data points reside; we will do so by examining data points' locations along several values of $\vec{w}$ using the mechanics we just developed. Assume now that the points $\vec{x}_1, \ldots, \vec{x}_m$ are in $\mathbb{R}^2$ for easier plotting and analysis, and that **the sample average of the points, $\hat{\vec{x}}$, is zero**, as in part (a) above.

Let $\sigma^2(\vec{w})$ denote the empirical variance of $t_i(\vec{w})$:

$$\sigma^2(\vec{w}) = \frac{1}{m}\sum_{i=1}^{m}(t_i(\vec{w}) - \hat{t}(\vec{w}))^2$$

and correspondingly, let $\sigma(\vec{w})$ denote its standard deviation. For the remainder of this problem, we will consider only points that lie within three standard deviations of our data mean, as measured along a given unit vector $\vec{w}$. In other words, for each unit vector $\vec{w}$, let us assume that the points $\vec{x}_1, \ldots, \vec{x}_m$ belong to the set

$$S(\vec{w}) \doteq \left\{\vec{x} \in \mathbb{R}^2 : \hat{t}(\vec{w}) - 3\sigma(\vec{w}) \leq \vec{w}^\top \vec{x} \leq \hat{t}(\vec{w}) + 3\sigma(\vec{w})\right\}.$$

Now consider a single unit vector $\vec{w} = \begin{bmatrix}1 & 0\end{bmatrix}^\top$ and assume $\sigma(\vec{w}) = 1$. Describe the shape of $S(\vec{w})$ and sketch this set in $\mathbb{R}^2$.

(c) We now consider several values of $\vec{w}$ and and again assume that points $\vec{x}_1, \ldots, \vec{x}_m$ reside in $S(\vec{w})$ for all considered $\vec{w}$, as well as that $\hat{\vec{x}}$ remains zero. We can visualize the region occupied by these points by finding the intersection of all sets $S(\vec{w})$. Let the sample covariance matrix Σ be $\begin{bmatrix}3 & -1\\ -1 & 3\end{bmatrix}$. For each of the following values of $\vec{w}$:

$$\vec{w} \in \left\{\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}\frac{\sqrt{2}}{2}\\ \frac{\sqrt{2}}{2}\end{bmatrix}, \begin{bmatrix}-\frac{\sqrt{2}}{2}\\ \frac{\sqrt{2}}{2}\end{bmatrix}\right\}$$

use the sample covariance matrix to calculate $\sigma(\vec{w})$, and shade each region $S(\vec{w})$ in $\mathbb{R}^2$ on the same axes. What does their intersection look like?

(d) Suppose we were able to calculate $S(\vec{w})$ for every possible value of $\vec{w}$ (i.e., every vector in the unit circle). Assuming the same value of Σ fom part (c), what would the intersection of all $S(\vec{w})$ look like? Sketch this intersection on your plot from part (c).

4. Gradients and Hessians

The *gradient* of a scalar-valued function $g : \mathbb{R}^n \to \mathbb{R}$, is the column vector of length n, denoted as ∇g, containing the derivatives of components of g with respect to the variables:

$$(\nabla g(\vec{x}))_i = \frac{\partial g}{\partial x_i}(\vec{x}),\ i = 1, \ldots n.$$

The *Hessian* of a scalar-valued function $g : \mathbb{R}^n \to \mathbb{R}$, is the $n \times n$ matrix, denoted as $\nabla^2 g$, containing the second derivatives of components of g with respect to the variables:

$$(\nabla^2 g(\vec{x}))_{ij} = \frac{\partial^2 g}{\partial x_i \partial x_j}(\vec{x}),\ \ i = 1, \ldots, n,\ \ j = 1, \ldots, n.$$

For the remainder of the class, we will repeatedly have to take gradients and Hessians of functions we are trying to optimize. This exercise serves as a warm up for future problems.

Compute the gradients and Hessians for the following functions:

(a) Compute the gradient and Hessian (with respect to $\vec{x}$) for $g(\vec{x}) = \vec{y}^\top A\vec{x}$.

(b) Compute the gradient and Hessian of $g(\vec{x}) = \|A\vec{x} - \vec{b}\|_2^2$. Recall from lecture and discussion that $\nabla_{\vec{x}}(\vec{x}^\top A\vec{x}) = (A + A^\top)\vec{x}$ and $\nabla_{\vec{x}}^2(\vec{x}^\top A\vec{x}) = A + A^\top$. (If you don't remember why these identities are true, this is a good opportunity to remind yourself).

5. Matrix norms

For a matrix $A \in \mathbb{R}^{m\times n}$, the *induced norm* or *operator norm* $\|A\|_p$ is defined as

$$\|A\|_p := \max_{\vec{x}\neq 0} \frac{\|A\vec{x}\|_p}{\|\vec{x}\|_p}.$$

Note that for our purposes, we use max instead of sup.

In this problem, we provide a characterization of the induced norm for certain values of p. Let a_{ij} denote the $(i, j)^{\text{th}}$ entry of A. Prove the following:

(a) $\|A\|_1$ is the maximum absolute column sum of A,

$$\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{m} |a_{ij}|.$$

Hint: Write $A\vec{x}$ as a linear combination of the columns of A to obtain $\|A\vec{x}\|_1 = \|\sum_{i=1}^{n} x_i \cdot \vec{a}_i\|_1$, where $\vec{a}_i$ denotes the i^{th} column of A. Then apply triangle inequality to terms within the sum.

(b) $\|A\|_\infty$ is the maximum absolute row sum of A,

$$\|A\|_\infty = \max_{1\leq i\leq m} \sum_{j=1}^{n} |a_{ij}|.$$

Hint: First write $\|A\vec{x}\|_\infty = \max_i \left|\sum_{j=1}^{n} a_{ij}x_j\right|$. Then apply triangle inequality and use the fact that $|x_j| \leq \max_i |x_i|, \ \forall j$.

(c) $\|A\|_2 = \sigma_{\max}(A)$, the maximum singular value of A.
Hint: One approach is to start by considering the SVD of A and use properties of orthonormal/orthogonal matrices.

6. Homework process

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