EE263 mid-term exam, November 2022

- This is a 24-hour take-home midterm. Please turn it in on Gradescope. Be aware that you *must* turn it in within 24 hours of downloading it. After that, Gradescope will not let you turn it in and we cannot accept it.
- You may use any books, notes, or computer programs. You may not discuss the exam or course material with others, or work in a group.
- The exam should not be discussed at all until 11/7 after everyone has taken it.
- If you have a question, please submit a private question on Ed, or email the staff mailing list. We have tried very hard to make the exam unambiguous and clear, so unless there is a mistake on the exam we're unlikely to say much.
- We expect your solutions to be legible, neat, and clear. Do not hand in your rough notes, and please try to simplify your solutions as much as you can. We will deduct points from solutions that are technically correct, but much more complicated than they need to be.
- Please check your email during the exam, just in case we need to send out a clarification or other announcement.
- Start each question on a new page. Correctly assign pages to problems in gradescope. We may take off points if a submission does not do so.
- We will be more thorough grading the midterm than with the homeworks. Please show the work you do, as it especially helps us give partial credit.
- When a problem involves some computation (say, using Julia), we do not want just the final answers. We want a clear discussion and justification of exactly what you did as well as the final numerical result.
- Because this is an exam, **you must turn in your code**. Include the code in your pdf submission. We reserve the right to deduct points for missing code.
- In the portion of your solutions where you explain the mathematical approach, you *cannot* refer to Julia operators, such as the backslash operator. (You can, of course, refer to inverses of matrices, or any other standard mathematical constructs.)
- Some of the problems require you to download data or other files. These files can be found at the URL

  `http://ee263.stanford.edu/mid22.html`
- Good luck!

1. **Recursive estimation.** A piecewise constant signal is filtered by convolving it with a smooth function. We start with $x \in \mathbb{R}^n$, and upsample (repeat values) to create $u \in \mathbb{R}^m$. The upsampling repeats each value k times, so that $m = kn$. The constant signal $u \in \mathbb{R}^m$ is given by

$$u_i = x_j \text{ for } k(j-1) < i \leq kj$$

The signal u is convolved with a smooth function r, given by

$$r_j = \exp(-j^2/\sigma^2)$$

which is defined for $-q \leq j \leq q$. The convolution operation generates output $y \in \mathbb{R}^m$, given by

$$y_i = w_i + \sum_{j=\max(1,i-q)}^{\min(m,i+q)} r_{i-j} u_j \tag{1}$$

where w is random measurement noise. We have $n = 10$, $k = 5$, $\sigma = 2$, $q = 10$. We will use regularization parameter $\mu = 0.1$. The file `recursive.json` contains x, y and w, which satisfy equation (1).

   a) Find matrix C such that $u = Cx$.

   b) Find matrix B such that $y = Bu + w$.

   c) Let $A = BC$. Find x^{reg}, the regularized least-squares estimate of x given y. That is, x^{reg} is the x that minimizes

   $$\|Ax - y\|^2 + \mu\|x\|^2$$

   Plot x^{reg} and x on the same plot. (*i.e.*, plot x_i versus i)

   d) We would like to use a recursive method to compute the regularized least-squares estimate. Recall the usual recursive-least-squares algorithm:

   $$\begin{aligned}
   &P(0) = 0 \in \mathbb{R}^{n\times n} \\
   &q(0) = 0 \in \mathbb{R}^{n} \\
   &\textbf{for } i = 0, 1, \ldots, \\
   &\qquad P(i+1) = P(i) + a_{i+1}a_{i+1}^{\mathsf{T}} \\
   &\qquad q(i+1) = q(i) + y_{i+1}a_{i+1}
   \end{aligned}$$

   where a_i^{T} is the i'th row of A, and y_i is the corresponding ith measurement. Then the estimate based on $y_1, \ldots, y_i$ is $x_{\text{ls}}(i) = P(i)^{-1}q(i)$.

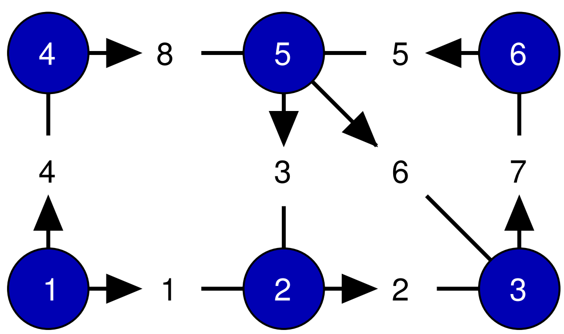
   Explain how to modify this algorithm to recursively compute the regularized least-squares estimate.

e) Apply your algorithm to the given data. Plot your estimate when $i = 18$ and when $i = 30$.

f) After applying your algorithm, when $i = m$, you will have computed the same regularized least-squares estimate you did in part (c), but in a different way. At this point, you realize that the data $y_1, \dots, y_{20}$ was incorrect, and you would like to remove it from your estimate. However, you have already thrown away $y_{21}, \dots, y_m$. Give an algorithm to adjust your estimate to remove the effect of measurments $y_1, \dots, y_{20}$. Plot the resulting estimate of x. Note that you only have access to $y_1, \cdots, y_{20}$, the final P and q from part (d), and $a_1, \cdots, a_{20}$.

2. Synchronicity.

a) The following graph shows a cluster of n machines in a data center, arranged in the form of a directed graph. Each machine has a clock, and communicates with its neighbors, to determine the *clock difference* between them. Specifically, machine i has a clock which reads c_i, in seconds. For each edge $i \to j$, we measure the clock difference $c_i - c_j$. For simplicity, we assume that this clock difference can be (approximately) measured by accounting for the known communication latency between the machines.



Edges are numbered $1, \dots m$. Let y_e be the clock difference measured along edge e. Then we have $y = B^\mathsf{T} c$ for some matrix B. Find B for the graph shown above.

b) Since we are only measuring clock *differences*, we do not expect to be able to determine c unambiguously. For the above graph, given a measurement $y \in \mathbb{R}^m$, what is the set of c consistent with this measurement?

c) A *cycle* in the graph is a sequence of distinct vertices, such at $1, 2, 5, 4, 1$ which start and end at the same vertex, and form a loop. The direction of the arrow is ignored in a cycle, all that matters is that there is an edge between successive vertices. For example, $2, 3, 5, 2$ is a cycle but $1, 2, 4, 1$ is not.

Show that the sum of clock differences around a cycle is zero, for a general graph.

d) Given a cycle in the graph, show how to construct a vector x in the nullspace of B. Hence construct a matrix K with the maximum number of columns such that

$$BK = 0 \qquad \text{and} \qquad \text{null}(K) = 0$$

and all entries of K are -1, 0 or 1.

e) We measure the following clock differences

$$y = (-0.56, -0.7, 1.13, -1.12, -1.45, 0.43, 1.02, -0.57)$$

Show that these values are *consistent* difference measurements; that is, there exists a vector of clock values c such that $y = B^\mathsf{T} c$. Do this without solving the least squares for c and then comparing y and $B^\mathsf{T} c$.

f) Assume the first clock has $c_1 = 0$. Given y in the previous part, find $c_2, \ldots, c_n$.

g) Now consider the case where we cannot measure clock differences perfectly; we measure instead

$$y = B^\mathsf{T} c + w$$

where w is some small error. We would like to find an estimate of the clocks c. To do this, we decide to solve

$$\begin{array}{ll} \text{minimize} & \|y - B^\mathsf{T} c\| \\ \text{subject to} & c_1 = 0 \end{array}$$

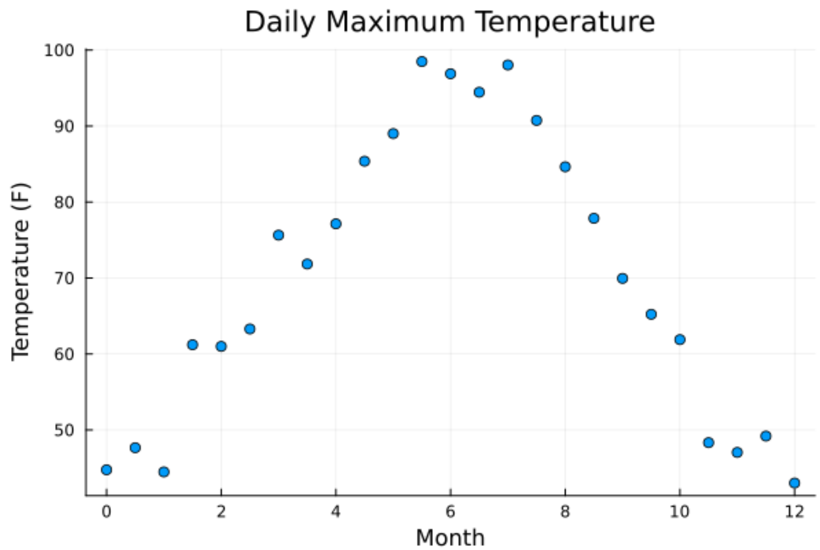
Give an algorithm for doing this.

h) We make a noisy measurement of the clock differences

$$y = (0.487, -0.128, 0.789, 0.245, 0.184, 0.506, -0.839, -0.647)$$

Using your algorithm from the previous part, estimate c.

3. Fitting a Piecewise Linear Function to Data. Last year, we sampled the maximum daily temperature outside Packard twice a month. Looking at the plotted results, we believe there's a clear trend in the temperatures over the time of year. We have collected $n = 25$ datapoints. Each data point consists of two values: x and y. The x value ranges from 0 to 12 and describe when the data was collected in months since the start of the year. The y value is the recorded temperature in Fahrenheit. We have two data sets, a training set and a test set, in the file `tempdata.json`.



In order to better describe the relationship between time of year and daily maximum temperature, we will fit a piecewise linear function g to the training data set. We will use $m+1$ piecewise affine functions $f_0, f_1, \ldots, f_m$, and approximate the data by the function g, which is a linear combination of them.

$$g(x) = \sum_{j=0}^{m} \alpha_j f_j(x)$$

Here $f_0 = 1$ and for $j = 1, \ldots, m$ we have

$$f_j(x) = \begin{cases} 0 & \text{if } x < (j-1)\frac{12}{m} \\ \frac{mx}{12} - (j-1) & \text{if } (j-1)\frac{12}{m} \leq x \leq j\frac{12}{m} \\ 1 & \text{if } j\frac{12}{m} < x \end{cases}.$$

The functions f_i are very simple piecewise linear step functions. We recommend graphing a couple for variable m and j to get intuition for what these functions are.

a) Our objective is to select weights $\alpha_0, \ldots, \alpha_m$ to minimize

$$\sum_{i=1}^{n} \|y_i - g(x_i)\|_2^2.$$

Express this objective in the form

$$\text{minimize } \|y - F\alpha\|_2^2$$

for some known vector y, known matrix F, and unknown vector α.

b) Suppose n is a multiple of m, and there are $n+1$ data points that are evenly spaced in x from 0 to 12 inclusive, so that gap between points is $12/n$. Show that the matrix F is full rank.

c) Use Julia to solve this problem for $m = 3, 6, 12, 24$ for the training data x^{train} and y^{train}. Plot the data points (x, y) along with the fitted function g for each value of m.

d) Plot the minimal squared 2-norm error ($\|y - F\alpha\|_2^2$) for $m = 3, 6, 12, 24$. Is there a point where adding more complexity to the model (increasing m) offers clearly diminishing returns?

e) We now turn to validation of the fit. We have an additional data set, x^{test} and y^{test}, which we will use to test the accuracy of our model. The test error is

$$J^{\text{test}} = \sum_{i=1}^{n} \|y_i^{\text{test}} - g(x_i^{\text{test}})\|_2^2.$$

Here g is the function you found in part (c) above. Plot the test error J^{test} versus m for $m = 3, 6, 12, 24$. Note that this does not involve recomputing α. What does this say about your answer to part (d).

4. Some true or false questions. For each of the statements below, state whether it is true or false. If true, give a brief one-sentence explanation why. If false, give a counterexample.

a) If $f : \mathbb{R}^n \to \mathbb{R}^n$ is a linear function and $A \in \mathbb{R}^{n \times n}$, then $f(Ax) = Af(x)$.

b) There exists a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $x \in \mathbb{R}^n$ such that $A^\mathsf{T} A x = 0$ but $Ax \neq 0$.

c) If $u, v, w \in \mathbb{R}^n$, and $\operatorname{rank} \begin{bmatrix} u & v & w \end{bmatrix} = 3$, then $\operatorname{rank} \begin{bmatrix} u+v & v+w & w+u \end{bmatrix} = 3$.

d) If $\operatorname{rank}(A) < \operatorname{rank} \begin{bmatrix} A & B \end{bmatrix}$, then $\operatorname{rank}(A) < \operatorname{rank} \begin{bmatrix} A \\ B \end{bmatrix}$.

e) For any $A, B \in \mathbb{R}^{m \times n}$, $\operatorname{rank} \begin{bmatrix} A & B \end{bmatrix} = \operatorname{rank} \begin{bmatrix} A & A+B \end{bmatrix}$.

f) If $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{m \times k}$, then $\operatorname{range}(A) \perp \operatorname{range}(B)$ if and only if $A^\mathsf{T} B = 0$.

g) If $A \in \mathbb{R}^{n \times n}$ and $\operatorname{rank}(A) = r$, then $\dim \operatorname{null} \begin{bmatrix} A & A^2 \end{bmatrix} = 2n - r$.

h) Suppose $A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$ with $\begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ orthogonal, with $Q_1 \in \mathbb{R}^{m \times r}$ and $R_1 \in \mathbb{R}^{r \times n}$. If $\operatorname{null}(R_1^\mathsf{T}) = \{0\}$ then $\operatorname{range}(A) = \operatorname{range}(Q_1)$.

i) Suppose $A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$ with $\begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ orthogonal, with $Q_1 \in \mathbb{R}^{m \times r}$ and $R_1 \in \mathbb{R}^{r \times n}$. If $\text{null}(R_1^\mathsf{T}) = \{0\}$ and $x \notin \text{range}(A)$, then $Q_1^\mathsf{T} x = 0$.

j) Suppose $A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} I & N \\ 0 & 0 \end{bmatrix}$ with $\begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ orthogonal, with $Q_1 \in \mathbb{R}^{m \times r}$ and $N \in \mathbb{R}^{r \times (n-r)}$. Then $\text{null}(A) = \text{range} \begin{bmatrix} -N \\ I \end{bmatrix}$.

5. **Some properties of the rank.** In this problem, we will prove a useful inequality about the rank of matrices, intuitively indicating that the rank of the multiplication of two matrices cannot be *too small* if the original matrices have high ranks. Each part of the problem consists of a small step of the proof.

a) Let I_k be the $k \times k$ identity matrix. Then, for arbitrary matrices A and B with appropriate dimensions, what is the rank of the following matrices M_1 and M_2? Justify your answer.

$$M_1 = \begin{pmatrix} I_k & 0 \\ A & I_\ell \end{pmatrix}, \quad M_2 = \begin{pmatrix} I_k & B \\ 0 & I_\ell \end{pmatrix}$$

b) Let $U \in \mathbb{R}^{(k+\ell) \times m}$ be an arbitrary matrix of rank r. Express the rank of $M_1 U$ and $U^T M_2$ in terms of k, ℓ, m, and r, and justify your answer.

c) Let $C \in \mathbb{R}^{p \times n}$ and $D \in \mathbb{R}^{n \times q}$ be arbitrary matrices. Find two block matrices M_1 and M_2 with similar block structure to the ones introduced in part (a) such that

$$M_1 \begin{pmatrix} I_n & D \\ C & 0 \end{pmatrix} M_2 = \begin{pmatrix} I_n & 0 \\ 0 & -CD \end{pmatrix}.$$

d) Using the results from the previous parts, show that

$$\text{rank} \begin{pmatrix} I_n & D \\ C & 0 \end{pmatrix} - \text{rank}(CD)$$

is a constant. Find this constant in terms of n, p, and q, and justify your answer.

e) Show that

$$\text{rank}(C) + \text{rank}(D) \leq \text{rank} \begin{pmatrix} I_n & D \\ C & 0 \end{pmatrix}.$$

Note that you can solve this part independently, without needing any results from the previous parts. Also, even if you cannot prove the result of this part, you can use if for the next part.

Hint: one way to approach this problem is to first show that, if C and D are full-column rank, then for any matrix J,

$$\text{rank} \begin{bmatrix} J & D \\ C & 0 \end{bmatrix} = \text{rank}(D) + \text{rank}(C)$$

f) Use the results from parts (d) and (e) to show that

$$\text{rank}(C) + \text{rank}(D) - \text{rank}(CD) \le s,$$

where s is the constant you found in part (d).