Homework 3

EE 263 Stanford University Summer 2019

Due: July 17, 2019

Question **7**, and **8** are **optional** questions (they will not be graded) and you do not need to submit them to gradescope. However, doing them will help improve your understanding and grasp of the material. Q8 is quite interesting.

1. Memory of a linear, time-invariant system. Suppose an input signal $(u_t : t \in \mathbb{Z})$, and an output signal $(y_t : t \in \mathbb{Z})$ are related by a convolution operator:

$$y_t = \sum_{\tau=1}^{M} h_\tau u_{t-\tau},$$

where $h = (h_1, \ldots, h_M)$ are the impulse-response coefficients of the convolution system. (Convolution systems are also called linear, time-invariant systems.) If $h_M \neq 0$, then M is called the memory of the system. You are given the input and output signals for $t = 1, \ldots, T$:

$$u_1, \ldots, u_T \qquad \text{and} \qquad y_1, \ldots, y_T.$$

However, you do not know u_t or y_t for $t < 1$ or $t > T$, and you do not know the impulse response, h.

a) Explain how to find the smallest value of M, and a corresponding impulse response $(h_t : t = 1, \ldots, M)$ that is consistent with the given data. You may assume that $T > 2M$.

b) Apply your method to the data in `lti_memory_data.m`. Report the value of M that you find.

   Hint. The function `toeplitz` may be useful.

2. Norm preserving implies orthonormal columns. In lecture we saw that if $A \in \mathbb{R}^{m \times n}$ has orthonormal columns, *i.e.*, $A^\mathsf{T} A = I$, then for any vector $x \in \mathbb{R}^n$ we have $\|Ax\| = \|x\|$. In other words, multiplication by such a matrix preserves norm.

Show that the converse holds: If $A \in \mathbb{R}^{m \times n}$ satisfies $\|Ax\| = \|x\|$ for all $x \in \mathbb{R}^n$, then A has orthonormal columns (and in particular, $m \geq n$).

Hint. Start with $\|Ax\|^2 = \|x\|^2$, and try $x = e_i$, and also $x = e_i + e_j$, for all $i \neq j$.

3. **Sensor integrity monitor.** A suite of m sensors yields measurement $y \in \mathbb{R}^m$ of some vector of parameters $x \in \mathbb{R}^n$. When the system is operating normally (which we hope is almost always the case) we have $y = Ax$, where $m > n$. If the system or sensors fail, or become faulty, then we no longer have the relation $y = Ax$. We can exploit the redundancy in our measurements to help us identify whether such a fault has occured. We'll call a measurement y *consistent* if it has the form Ax for some $x \in \mathbb{R}^n$. If the system is operating normally then our measurement will, of course, be consistent. If the system becomes faulty, we hope that the resulting measurement y will become inconsistent, *i.e.*, not consistent. (If we are *really* unlucky, the system will fail in such a way that y is still consistent. Then we're out of luck.) A matrix $B \in \mathbb{R}^{k\times m}$ is called an *integrity monitor* if the following holds:

   - $By = 0$ for any y which is consistent.
   - $By \neq 0$ for any y which is inconsistent.

   If we find such a matrix B, we can quickly check whether y is consistent; we can send an alarm if $By \neq 0$. Note that the first requirement says that every consistent y does not trip the alarm; the second requirement states that every inconsistent y does trip the alarm. Finally, the problem. Find an integrity monitor B for the matrix

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 1 & -1 & -2 \\ -2 & 1 & 3 \\ 1 & -1 & -2 \\ 1 & 1 & 0 \end{bmatrix}.$$

   Your B should have the smallest k (*i.e.*, number of rows) as possible. As usual, you have to explain what you're doing, as well as giving us your explicit matrix B. You must also verify that the matrix you choose satisfies the requirements. *Hints:*

   - You might find one or more of the matlab commands `orth`, `null`, or `qr` useful. Then again, you might not; there are many ways to find such a B.
   - When checking that your B works, don't expect to have By exactly zero for a consistent y; because of roundoff errors in computer arithmetic, it will be really, really small. That's OK.
   - Be very careful typing in the matrix A. It's not just a random matrix.

4. **Coin collector robot.** Consider a robot with unit mass which can move in a frictionless two dimensional plane. The robot has a constant unit speed in the y direction (towards north), and it is designed such that we can only apply force in the x direction. We will apply a force at time t given by f_j for $2j - 2 \leq t < 2j$ where $j = 1, \ldots, n$, so that the applied force is constant over time intervals of length 2. The robot is at the origin at time $t = 0$ with zero velocity in the x direction.

   There are $2n$ coins in the plane and the goal is to design a sequence of input forces for the robot to collect the maximum possible number of coins. The robot is designed such that it can collect the ith coin only if it exactly passes through the location of the coin (x_i, y_i). In this problem, we assume that $y_i = i$.

a) Find the coordinates of the robot at time t, where t is a positive integer. Your answer should be a function of t and the vector of input forces $f \in \mathbf{R}^n$.

b) Given a sequence of k coins $(x_1, y_1), \ldots, (x_{2n}, y_{2n})$, describe a method to find whether the robot can collect them.

c) For the data provided in `robot_coin_collector.m`, show that the robot cannot collect all the coins.

d) Suppose that there is an arrangement of the coins such that it is not possible for the robot to collect all the coins. Suggest a way to check if the robot can collect all but one of the coins.

e) Run your method on data in `robot_coin_collector.m` and report which coin cannot be collected. Report the input that results in collecting $2n-1$ coins. Plot the location of the coins and the location of the robot at integer times.

5. Solving linear equations via QR factorization. Consider the problem of solving the linear equations $Ax = y$, with $A \in \mathbb{R}^{n\times n}$ nonsingular, and y given. We can use the Gram-Schmidt procedure to compute the QR factorization of A, and then express x as $x = A^{-1}y = R^{-1}(Q^\mathsf{T}x) = R^{-1}z$, where $z = Q^\mathsf{T}y$. In this exercise, you'll develop a method for computing $x = R^{-1}z$, *i.e.*, solving $Rx = z$, when R is upper triangular and nonsingular (which means its diagonal entries are all nonzero).

The trick is to first find x_n; then find x_{n-1} (remembering that now you know x_n); then find x_{n-2} (remembering that now you know x_n and x_{n-1}); and so on. The algorithm you will discover is called *back substitution*, because you are substituting known or computed values of x_i into the equations to compute the next x_i (in reverse order). Be sure to explain why the algorithm you describe cannot fail.

6. Quadratic extrapolation of a time series, using least-squares fit. We are given a series z up to time t. We extrapolate, or predict, $z(t+1)$ based on a least-squares fit of a quadratic function to the previous ten elements of the series, $z(t), z(t-1), \ldots, z(t-9)$. We'll denote the predicted value of $z(t+1)$ by $\hat{z}(t+1)$. More precisely, to find $\hat{z}(t+1)$, we find the quadratic function $f(\tau) = a_2\tau^2 + a_1\tau + a_0$ for which

$$\sum_{\tau=t-9}^{t} (z(\tau) - f(\tau))^2$$

is minimized. The extrapolated value is then given by $\hat{z}(t+1) = f(t+1)$.

a) Show that

$$\hat{z}(t+1) = c \begin{bmatrix} z(t) \\ z(t-1) \\ \vdots \\ z(t-9) \end{bmatrix},$$

where $c \in \mathbb{R}^{1\times 10}$ does not depend on t. Find c explicitly.

b) Use the following matlab code to generate a time series z:

```
t = 1:1000;
z = 5*sin(t/10 + 2) + 0.1*sin(t) + 0.1*sin(2*t - 5);
```

Use the quadratic extrapolation method from part (a) to find $\hat{z}_{ls}(t)$ for $t = 11, \ldots, 1000$. Find the relative root-mean-square (RMS) error, which is given by

$$\left(\frac{(1/990)\sum_{j=11}^{1000}(\hat{z}(j) - z(j))^2}{(1/990)\sum_{j=11}^{1000} z(j)^2}\right)^{1/2}.$$

c) In a previous problem you developed a similar predictor for the same time series z. In that case you obtained the quadratic extrapolator by *interpolating* the last three samples; now you are obtaining it as the *least squares fit to the last ten samples*. Compare the RMS error for these methods and plot z (the true values), $\hat{z}_{\text{ls}}$ (the estimated values using least-squares), and $\hat{z}_{\text{int}}$ (the estimated values using interpolation), on the same plot. Restrict your plot to $t = 1, \ldots, 100$.

7. Householder reflections. A *Householder matrix* is defined as

$$Q = I - 2uu^{\mathsf{T}},$$

where $u \in \mathbb{R}^n$ is normalized, that is, $u^{\mathsf{T}}u = 1$.

a) Show that Q is orthogonal.

b) Show that $Qu = -u$. Show that $Qv = v$, for any v such that $u^{\mathsf{T}}v = 0$. Thus, multiplication by Q gives reflection through the plane with normal vector u.

c) Show that $\det Q = -1$.

d) Given a vector $x \in \mathbb{R}^n$, find a unit-length vector u for which Qx lies on the line through e_1. *Hint:* Try a u of the form $u = v/\|v\|$, with $v = x + \alpha e_1$ (find the appropriate α and show that such a u works …) Compute such a u for $x = (3, 2, 4, 1, 5)$. Apply the corresponding Householder reflection to x to find Qx.

Note: Multiplication by an orthogonal matrix has very good numerical properties, in the sense that it does not accumulate much roundoff error. For this reason, Householder reflections are used as building blocks for fast, numerically sound algorithms.

8. Vector space multiple access (VSMA). We consider a system of k transmitter-receiver pairs that share a common medium. The goal is for transmitter i to transmit a vector signal $x_i \in \mathbb{R}^{n_i}$ to the ith receiver, without interference from the other transmitters. All receivers have access to the same signal $y \in \mathbb{R}^m$, which includes the signals of all transmitters, according to

$$y = A_1 x_1 + \cdots + A_k x_k,$$

where $A_i \in \mathbb{R}^{m \times n_i}$. You can assume that the matrices A_i are skinny, *i.e.*, $m \geq n_i$ for $i = 1, \ldots, k$. (You can also assume that $n_i > 0$ and $A_i \neq 0$, for $i = 1, \ldots, k$.) Since the k transmitters all share the same m-dimensional vector space, we call this *vector space multiple access*. Each receiver knows the received signal y, and the matrices $A_1, \ldots, A_k$.

We say that the ith signal is *decodable* if the ith receiver can determine the value of x_i, no matter what values $x_1, \ldots, x_k$ have. Roughly speaking, this means that receiver i can process the received signal so as to perfectly recover the ith transmitted signal, while rejecting any interference from the other signals $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k$. Whether or not the ith signal is decodable depends, of course, on the matrices $A_1, \ldots, A_k$.

Here are four statements about decodability:

a) Each of the signals $x_1, \ldots, x_k$ is decodable.

b) The signal x_1 is decodable.

c) The signals $x_2, \ldots, x_k$ are decodable, but x_1 isn't.

d) The signals $x_2, \ldots, x_k$ are decodable when x_1 is 0.

For each of these statements, you are to give the exact (*i.e.*, necessary and sufficient) conditions under which the statement holds, in terms of $A_1, \ldots, A_k$ and $n_1, \ldots, n_k$. Each answer, however, must have a very specific form: it must consist of a conjunction of one or more of the following properties:

I. $\text{rank}(A_1) < n_1$.

II. $\text{rank}([A_2 \ \cdots \ A_k]) = n_2 + \cdots + n_k$.

III. $\text{rank}([A_1 \ \cdots \ A_k]) = n_1 + \text{rank}([A_2 \ \cdots \ A_k])$.

IV. $\text{rank}([A_1 \ \cdots \ A_k]) = \text{rank}(A_1) + \text{rank}([A_2 \ \cdots \ A_k])$.

As examples, possible answers (for each statement) could be "I" or "I and II", or "I and II and IV". For some statements, there may be more than one correct answer; we will accept any correct one.

You can also give the response "My attorney has advised me not to respond to this question at this time." This response will receive partial credit.

For (just) this problem, we want only your answers. We do not want, and will not read, any further explanation or elaboration, or any other type of answers.