Homework 3

EE263 Stanford University, Fall 2017

Due: Wednesday 10/18/17 11:59pm

1. **Orthogonal complement of a subspace.** If $\mathcal{V}$ is a subspace of $\mathbb{R}^n$ we define $\mathcal{V}^\perp$ as the set of vectors orthogonal to every element in $\mathcal{V}$, *i.e.*,

$$\mathcal{V}^\perp = \{ \; x \mid \langle x, y \rangle = 0, \; \forall y \in \mathcal{V} \; \}.$$

   a) Verify that $\mathcal{V}^\perp$ is a subspace of $\mathbb{R}^n$.

   b) Suppose $\mathcal{V}$ is described as the span of some vectors $v_1, v_2, \ldots, v_r$. Express $\mathcal{V}$ and $\mathcal{V}^\perp$ in terms of the matrix $V = \begin{bmatrix} v_1 & v_2 & \cdots & v_r \end{bmatrix} \in \mathbb{R}^{n \times r}$ using common terms (range, nullspace, transpose, etc.)

   c) Show that every $x \in \mathbb{R}^n$ can be expressed uniquely as $x = v + v^\perp$ where $v \in \mathcal{V}$, $v^\perp \in \mathcal{V}^\perp$. *Hint:* let v be the projection of x on $\mathcal{V}$.

   d) Show that $\dim \mathcal{V}^\perp + \dim \mathcal{V} = n$.

   e) Show that $\mathcal{V} \subseteq \mathcal{U}$ implies $\mathcal{U}^\perp \subseteq \mathcal{V}^\perp$.

2. **Interpolation with rational functions..** Consider a function $f : \mathbb{R} \to \mathbb{R}$ of the form

$$f(x) = \frac{a_0 + a_1 x + \cdots + a_m x^m}{1 + b_1 x + \cdots + b_m x^m},$$

where $a_0, \ldots, a_m$ and $b_1, \ldots, b_m$ are parameters, with either $a_m \neq 0$ or $b_m \neq 0$. Such a function is called a rational function of degree m. We are given data points $x_1, \ldots, x_N \in \mathbb{R}$, and $y_1, \ldots, y_N \in \mathbb{R}$, where $y_i = f(x_i)$.

   a) Explain how to find a rational function of smallest degree that is consistent with the data: that is, explain how to find the smallest value of m, and corresponding values of $a_0, \ldots, a_m$, and $b_1, \ldots, b_m$ such that $f(x_i) = y_i$ for $i = 1, \ldots, N$.

   b) Carry out your method on the data in `rational_interpolation_data.m`. Report your value of m, and the corresponding coefficients $a_0, \ldots, a_m$, and $b_1, \ldots, b_m$. Plot the data and the rational function $f(x)$. Verify that $y_i = f(x_i)$ for $i = 1, \ldots, N$ (possibly with small numerical errors).

3. Checking some range and nullspace conditions. Explain how to determine whether or not the following statements hold:

a) $\text{range}(A) = \text{range}(B)$.

b) $\text{range}(A) \perp \text{range}(B)$.

c) $\text{range}(A) \cap \text{range}(B) = \{0\}$.

d) $\text{range}(C) \subseteq \text{null}(B)$.

The matrices have dimensions $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times p}$, $C \in \mathbb{R}^{p\times m}$.

Your answer can involve standard matrix operations on the matrices above, such as addition, multiplication, transposition, concatenation (*i.e.*, building block matrices), and inversion, as well as a function $\text{rank}(X)$, that gives the rank of a matrix X, and $\det(X)$, which gives the determinant of a (square) matrix X.

For example, you might assert that (a) holds if and only if $\text{rank}([A\ B]) = m$. (This is not correct; it's just an example of what your answer might look like.)

You do not need to give a proof or long justification that your conditions are correct; a short one or two sentence explanation for each statement is fine. Points will be deducted from correct answers that are substantially longer than they need to be, or are confusing (to us).

4. Orthogonal matrices.

a) Show that if U and V are orthogonal, then so is UV.

b) Show that if U is orthogonal, then so is U^{-1}.

c) Suppose that $U \in \mathbb{R}^{2\times 2}$ is orthogonal. Show that U is either a rotation or a reflection. Make clear how you decide whether a given orthogonal U is a rotation or reflection.

5. Identifying a system from input/output data. We consider the standard setup:

$$y = Ax + v,$$

where $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$ is the input vector, $y \in \mathbb{R}^m$ is the output vector, and $v \in \mathbb{R}^m$ is the noise or disturbance. We consider here the problem of estimating the matrix A, given some input/output data. Specifically, we are given the following:

$$x^{(1)}, \ldots, x^{(N)} \in \mathbb{R}^n, \qquad y^{(1)}, \ldots, y^{(N)} \in \mathbb{R}^m.$$

These represent N samples or observations of the input and output, respectively, possibly corrupted by noise. In other words, we have

$$y^{(k)} = Ax^{(k)} + v^{(k)}, \quad k = 1, \ldots, N,$$

where $v^{(k)}$ are assumed to be small. The problem is to estimate the (coefficients of the) matrix A, based on the given input/output data. You will use a least-squares criterion to form an estimate $\hat{A}$ of A. Specifically, you will choose as your estimate $\hat{A}$ the matrix that minimizes the quantity

$$J = \sum_{k=1}^{N} \|Ax^{(k)} - y^{(k)}\|^2$$

over A.

a) Explain how to do this. If you need to make an assumption about the input/output data to make your method work, state it clearly. You may want to use the matrices $X \in \mathbb{R}^{n \times N}$ and $Y \in \mathbb{R}^{m \times N}$ given by

$$X = \left[\, x^{(1)} \quad \cdots \quad x^{(N)} \,\right], \qquad Y = \left[\, y^{(1)} \quad \cdots \quad y^{(N)} \,\right]$$

in your solution.

b) On the course web site you will find some input/output data for an instance of this problem in the file `sysid_data.json`. Executing this Julia file will assign values to m, n, and N, and create two matrices that contain the input and output data, respectively. The $n \times N$ matrix variable `X` contains the input data $x^{(1)}, \ldots, x^{(N)}$ (*i.e.*, the first column of `X` contains $x^{(1)}$, etc.). Similarly, the $m \times N$ matrix `Y` contains the output data $y^{(1)}, \ldots, y^{(N)}$. You must give your final estimate $\hat{A}$, your source code, and also give an explanation of what you did.

6. Simple fitting. You are given some data $x_1, \ldots, x_N \in \mathbb{R}$ and $y_1, \ldots, y_N \in \mathbb{R}$. These data are available in `simplefitdata.m` on the course web site.

a) Find the best affine fit, *i.e.*, $y_i \approx ax_i + b$, where 'best' means minimizing $\sum_{i=1}^{N}(y_i - (ax_i + b))^2$. (This is often called the 'best linear fit'.) Set this up and solve it as a least-squares problem. Plot the data and the fit in the same figure. Give us a and b, and submit the code you used to find a and b.

b) Repeat for the best least-squares cubic fit, *i.e.*, $y_i \approx ax_i^3 + bx_i^2 + cx_i + d$.

7. Fitting a model for hourly temperature. You are given a set of temperature measurements (in degrees C), $y_t \in \mathbb{R}$, $t = 1, \ldots, N$, taken hourly over one week (so $N = 168$). An expert says that over this week, an appropriate model for the hourly temperature is a trend (*i.e.*, a linear function of t) plus a diurnal component (*i.e.*, a 24-periodic component):

$$\hat{y}_t = at + p_t,$$

where $a \in \mathbb{R}$ and $p \in \mathbb{R}^N$ satisfies $p_{t+24} = p_t$, for $t = 1, \ldots, N - 24$. We can interpret a (which has units of degrees C per hour) as the warming or cooling trend (for $a > 0$ or $a < 0$, respectively) over the week.

a) Explain how to find $a \in \mathbb{R}$ and $p \in \mathbb{R}^N$ (which is 24-periodic) that minimize the RMS value of $y - \hat{y}$.

b) Carry out the procedure described in part (a) on the data set found in `tempfit_data.m`.

   Give the value of the trend parameter a that you find. Plot the model $\hat{y}$ and the measured temperatures y on the same plot. (The matlab code to do this is in the data file, but commented out.)

c) *Temperature prediction.* Use the model found in part (b) to predict the temperature for the next 24-hour period (*i.e.*, from $t = 169$ to $t = 192$). The file `tempdata.m` also contains a 24 long vector `ytom` with tomorrow's temperatures. Plot tomorrow's temperature and your prediction of it, based on the model found in part (b), on the same plot. What is the RMS value of your prediction error for tomorrow's temperatures?

8. Empirical algorithm complexity. The runtime T of an algorithm depends on its input data, which is characterized by three key parameters: k, m, and n. (These are typically integers that give the dimensions of the problem data.) A simple and standard model that shows how T scales with k, m, and n has the form

$$\hat{T} = \alpha k^{\beta} m^{\gamma} n^{\delta},$$

where α, β, γ, $\delta \in \mathbb{R}$ are constants that characterize the approximate runtime model. If, for example, $\delta \approx 3$, we say that the algorithm has (approximately) cubic complexity in n. (In general, the exponents β, γ, and δ need not be integers, or close to integers.)

Now suppose you are given measured runtimes for N executions of the algorithm, with different sets of input data. For each data record, you are given T_i (the runtime), and the parameters k_i, m_i, and n_i. It's possible (and often occurs) that two data records have identical values of k, m, and n, but different values of T. This means the algorithm was run on two different data sets that had the same dimensions; the corresponding runtimes can be (and often are) a little different.

We wish to find values of α, β, γ, and δ for which our model (approximately) fits our measurements. We define the fitting cost as

$$J = (1/N) \sum_{i=1}^{N} \left(\log(\hat{T}_i / T_i) \right)^2,$$

where $\hat{T}_i = \alpha k_i^{\beta} m_i^{\gamma} n_i^{\delta}$ is the runtime predicted by our model, using the given parameter values. This fitting cost can be (loosely) interpreted in terms of relative or percentage fit. If $(\log(\hat{T}_i/T_i))^2 \le \epsilon$, then $\hat{T}_i$ lies between $T_i / \exp\sqrt{\epsilon}$ and $T_i \exp\sqrt{\epsilon}$.

Your task is to find constants α, β, γ, δ that minimize J.

a) Explain how to do this. If your method always finds the values that give the true global minimum value of J, say so. If your algorithm cannot guarantee finding the true global minimum, say so. If your method requires some matrix (or matrices) to be full rank, say so.

b) Carry out your method on the data found in `empac_data.m`. Give the values of α, β, γ, and δ you find, and the corresponding value of J.