MATH 4931 - MSSC 5931 Homework 2

1. **Quadratic extrapolation of a time series.** We are given a series z up to time t. Using a quadratic model, we want to extrapolate, or predict, $z(t+1)$ based on the three previous elements of the series, $z(t)$, $z(t-1)$, and $z(t-2)$. We'll denote the predicted value of $z(t+1)$ by $\hat{z}(t+1)$. More precisely, you will find $\hat{z}(t+1)$ as follows.

   a) Find the quadratic function $f(\tau) = a_2\tau^2 + a_1\tau + a_0$ which satisfies $f(t) = z(t)$, $f(t-1) = z(t-1)$, and $f(t-2) = z(t-2)$. Then the extrapolated value is given by $\hat{z}(t+1) = f(t+1)$. Show that

   $$\hat{z}(t+1) = c \begin{bmatrix} z(t) \\ z(t-1) \\ z(t-2) \end{bmatrix},$$

   where $c \in \mathbb{R}^{1\times 3}$, and does not depend on t. In other words, the quadratic extrapolator is a linear function. Find c explicitly.

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

   ```
   t <- seq(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 $\hat{z}(t)$ for $t = 4, \ldots, 1000$. Find the relative root-mean-square (RMS) error, which is given by

   $$\left(\frac{(1/997)\sum_{j=4}^{1000} (\hat{z}(j) - z(j))^2}{(1/997)\sum_{j=4}^{1000} z(j)^2} \right)^{1/2}.$$

2. **Price elasticity of demand.** The demand for n different goods is a function of their prices:

   $$q = f(p),$$

   where p is the price vector, q is the demand vector, and $f : \mathbb{R}^n \to \mathbb{R}^n$ is the demand function. The current price and demand are denoted p^* and q^*, respectively. Now suppose there is a small price change δp, so $p = p^* + \delta p$. This induces a change in demand, to $q \approx q^* + \delta q$, where

   $$\delta q \approx Df(p^*)\delta p,$$

   where Df is the derivative or Jacobian of f, with entries

   $$Df(p^*)_{ij} = \frac{\partial f_i}{\partial p_j}(p^*).$$

   This is usually rewritten in term of the *elasticity matrix* E, with entries

   $$E_{ij} = \frac{\partial f_i}{\partial p_j}(p^*)\frac{1/q_i^*}{1/p_j^*},$$

so E_{ij} gives the relative change in demand for good i per relative change in price j. Defining the vector y of relative demand changes, and the vector x of relative price changes,

$$y_i = \frac{\delta q_i}{q_i^*}, \qquad x_j = \frac{\delta p_j}{p_j^*},$$

we have the linear model $y = Ex$.

Here are the questions:

a) What is a reasonable assumption about the diagonal elements E_{ii} of the elasticity matrix?

b) Goods i and j are called *substitutes* if they provide a similar service or other satisfaction (*e.g.*, train tickets and bus tickets, cake and pie, etc.). They are called *complements* if they tend to be used together (*e.g.*, automobiles and gasoline, left and right shoes, etc.). For each of these two generic situations, what can you say about E_{ij} and E_{ji}?

c) Suppose the price elasticity of demand matrix for two goods is

$$E = \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix}.$$

Describe the nullspace of E, and give an interpretation (in one or two sentences). What kind of goods could have such an elasticity matrix?

3. Halfspace. Suppose $a, b \in \mathbb{R}^n$ are two given points. Show that the set of points in $\mathbb{R}^n$ that are closer to a than b is a halfspace, *i.e.*:

$$\{x \mid \|x - a\| \leq \|x - b\| \ \} = \{ \ x \mid c^\mathsf{T} x \leq d\}$$

for appropriate $c \in \mathbb{R}^n$ and $d \in \mathbb{R}$. Give c and d explicitly, and draw a picture showing a, b, c, and the halfspace.

4. Temperatures in a multi-core processor. We are concerned with the temperature of a processor at two critical locations. These temperatures, denoted $T = (T_1, T_2)$ (in degrees C), are affine functions of the power dissipated by three processor cores, denoted $P = (P_1, P_2, P_3)$ (in W). We make 4 measurements. In the first, all cores are idling, and dissipate 10W. In the next three measurements, one of the processors is set to full power, 100W, and the other two are idling. In each experiment we measure and note the temperatures at the two critical locations.

P_1	P_2	P_3	T_1	T_2
10W	10W	10W	27°	29°
100W	10W	10W	45°	37°
10W	100W	10W	41°	49°
10W	10W	100W	35°	55°

Suppose we operate all cores at the same power, p. How large can we make p, without T_1 or T_2 exceeding 70°?

You must fully explain your reasoning and method, in addition to providing the numerical solution.

5. Projection matrices. A matrix $P \in \mathbb{R}^{n \times n}$ is called a projection matrix if $P = P^\mathsf{T}$, and $P^2 = P$. (These properties are sometimes called symmetry and idempotency, respectively.)

a) Show that if P is a projection matrix, then $I - P$ is also a projection matrix.

b) Suppose $U \in \mathbb{R}^{n \times k}$ has orthonormal columns. Show that UU^T is a projection matrix. (The converse is also true: every projection matrix can be written as UU^T for some matrix U with orthonormal columns; you do not need to prove this.)

c) Suppose $A \in \mathbb{R}^{n \times k}$ is skinny, and full rank. Show that $A(A^\mathsf{T}A)^{-1}A^\mathsf{T}$ is a projection matrix.

d) Given $S \subset \mathbb{R}^n$, and $x \in \mathbb{R}^n$, the point $\hat{x} \in S$ that is closest to x is called the projection of x onto S. Show that if P is a projection matrix, then $\hat{x} = Px$ is the projection of x onto $\text{range}(P)$. (This is the origin of the term "projection matrix.")

6. Single sensor failure detection and identification. We have $y = Ax$, where $A \in \mathbb{R}^{m \times n}$ is known, and $x \in \mathbb{R}^n$ is to be found. Unfortunately, up to one sensor may have failed (but you don't know which one has failed, or even whether any has failed). You are given $\tilde{y}$ and not y, where $\tilde{y}$ is the same as y in all entries except, possibly, one (say, the kth entry). If all sensors are operating correctly, we have $y = \tilde{y}$. If the kth sensor fails, we have $\tilde{y}_i = y_i$ for all $i \neq k$.

The file `one_bad_sensor.RData`, available on the course web site, defines A and $\tilde{y}$ (as `A` and `ytilde`). Determine which sensor has failed (or if no sensors have failed). You must explain your method, and submit your code.

For this exercise, you can use the R code `qr(cbind(F,g))$rank == qr(F)$rank` to check if $g \in \text{range}(F)$. (We will see later a much better way to check if $g \in \text{range}(F)$.)