EE263 Homework 1

Fall 2024

1. **The chain rule for vector-valued functions.** Let $f : \mathbb{R}^p \to \mathbb{R}^m$ and $g : \mathbb{R}^n \to \mathbb{R}^p$ be vector-valued functions, and consider the composite function $f \circ g : \mathbb{R}^n \to \mathbb{R}^m$.

   a) The chain rule for scalar-valued functions tells us that

   $$\frac{\partial f_i(g_1(x), \ldots, g_p(x))}{\partial x_j} = \sum_{k=1}^{p} \frac{\partial f_i}{\partial g_k(x)} \frac{\partial g_k(x)}{\partial x_j}.$$

   Use the chain rule for scalar-valued functions to show that

   $$D(f \circ g)(x) = (Df)(g(x))Dg(x).$$

   This result is the chain rule for vector-valued functions.

   b) Define $f : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}$ such that

   $$f(z) = \sqrt{z} \qquad \text{and} \qquad g(x) = \|x\|^2.$$

   Compute the derivatives of f and g, and use the chain rule for vector-valued functions to compute the derivative of $f \circ g$. Note that $(f \circ g)(x) = \|x\|$.

   c) For a given matrix $A \in \mathbb{R}^{m \times n}$, define the functions $f : \mathbb{R}^{m+n} \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^{m+n}$ such that

   $$f\left(\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}\right) = z_1^\mathsf{T} z_2 \qquad \text{and} \qquad g(x) = \begin{bmatrix} x \\ Ax \end{bmatrix}.$$

   Compute the derivatives of f and g, and use the chain rule for vector-valued functions to compute the derivative of $f \circ g$. Note that $(f \circ g)(x) = x^\mathsf{T} Ax$.

2. **Matrix representation of linear systems.** Consider the (discrete-time) linear dynamical system

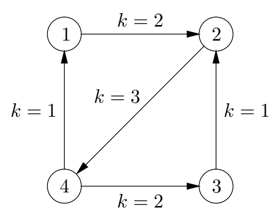
   $$x(t+1) = A(t)x(t) + B(t)u(t), \quad y(t) = C(t)x(t) + D(t)u(t).$$

   Find a matrix G such that

   $$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N) \end{bmatrix} = G \begin{bmatrix} x(0) \\ u(0) \\ \vdots \\ u(N) \end{bmatrix}.$$

   The matrix G shows how the output at $t = 0, \ldots, N$ depends on the initial state $x(0)$ and the sequence of inputs $u(0), \ldots, u(N)$.

3. **Communication over a wireless network with time-slots.** We consider a network with n nodes, labeled $1, \ldots, n$. A directed graph shows which nodes can send messages (directly) to which other nodes; specifically, an edge from node j to node i means that node j can transmit a message directly to node i. Each edge is assigned to one of K *time-slots*, which are labeled $1, \ldots, K$. At time period $t = 1$, only the edges assigned to time-slot 1 can transmit a message; at time period $t = 2$, only the edges assigned to time-slot 2 can transmit a message, and so on. After time period $t = K$, the pattern repeats. At time period $t = K + 1$, the edges assigned to time-slot 1 are again active; at $t = K + 2$, the edges assigned to time-slot 2 are active, etc. This cycle repeats indefinitely: when $t = mK + k$, where m is an integer, and $k \in \{1, \ldots, K\}$, transmissions can occur only over edges assigned to time-slot k. Although it doesn't matter for the problem, we mention some reasons why the possible transmissions are assigned to time-slots. Two possible transmissions are assigned to different time-slots if they would interfere with each other, or if they would violate some limit (such as on the total power available at a node) if the transmissions occurred simultaneously. A message or packet can be sent from one node to another by a sequence of transmissions from node to node. At time period t, the message can be sent across any edge that is active at period t. It is also possible to store a message at a node during any time period, presumably for transmission during a later period. If a message is sent from node j to node i in period t, then in period $t + 1$ the message is at node i, and can be stored there, or transmitted across any edge emanating from node i and active at time period $t + 1$. To make sure the terminology is clear, we consider the very simple example shown below, with $n = 4$ nodes, and $K = 3$ time-slots.



In this example, we can send a message that starts in node 1 to node 3 as follows:

- During period $t = 1$ (time-slot $k = 1$), store it at node 1.
- During period $t = 2$ (time-slot $k = 2$), transmit it to node 2.
- During period $t = 3$ (time-slot $k = 3$), transmit it to node 4.
- During period $t = 4$ (time-slot $k = 1$), store it at node 4.
- During period $t = 5$ (time-slot $k = 2$), transmit it to node 3.

You can check that at each period, the transmission used is active, *i.e.*, assigned to the associated time-slot. The sequence of transmissions (and storing) described above gets the message from node 1 to node 3 in 5 periods. Finally, the problem. We consider a specific network with $n = 20$ nodes, and $K = 3$ time-slots, with edges and time-slot assignments given in `ts_data.json`. The labeled graph that specifies the possible transmissions and the associated time-slot assignments are given in a matrix $A \in \mathbb{R}^{n \times n}$, as follows:

$$A_{ij} = \begin{cases} k & \text{if transmission from node } j \text{ to node } i \text{ is allowed, and assigned to time-slot } k \\ 0 & \text{if transmission from node } j \text{ to node } i \text{ is never allowed} \\ 0 & i = j. \end{cases}$$

Note that we set $A_{ii} = 0$ for convenience. This choice has no significance; you can store a message at any node in any period. To illustrate this encoding of the graph, consider the simple example described above. For this example, we have

$$A_{\text{example}} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 \end{bmatrix}.$$

The problems below concern the network described in the data file and *not* the simple example given above.

a) *Minimum-time point-to-point routing.* Find the fastest way to get a message that starts at node 5, to node 18. Give your solution as a prescription ordered in time from $t = 1$ to $t = T$ (the last transmission), as in the example above. At each time period, give the transmission (as in 'transmit from node 7 to node 9') or state that the message is to be stored (as in 'store at node 13'). Be sure that transmissions only occur during the associated time-slots. You only need to give *one* prescription for getting the message from node 5 to node 18 in minimum time.

b) *Minimum time flooding.* In this part of the problem, we assume that once the message reaches a node, a copy is kept there, even when the message is transmitted to another node. Thus, the message is available at the node to be transmitted along any active edge emanating from that node, at any future period. Moreover, we allow multi-cast: if during a time period there are multiple active edges emanating from a node that has (a copy of) the message, then transmission can occur during that time period across *all* (or any subset) of the active edges. In this part of the problem, we are interested in getting a message that starts at a particular node, to all others, and we attach no cost to storage or transmission, so there is no harm is assuming that at each time period, every node that has the message forwards it to all nodes it is able to transmit to. What is the minimum time it takes before all nodes have a message that starts at node 7?

For both parts of the problem, you must give the specific solution, as well as a description of your approach and method.

4. Affine functions. A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is called *affine* if for any $x,\ y \in \mathbb{R}^n$ and any $\alpha,\ \beta \in \mathbb{R}$ with $\alpha + \beta = 1$, we have

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y).$$

(Without the restriction $\alpha + \beta = 1$, this would be the definition of linearity.)

a) Suppose that $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Show that the function $f(x) = Ax + b$ is affine.

b) Now the converse: Show that any affine function f can be represented as $f(x) = Ax + b$, for some $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. (This representation is unique: for a given affine function f there is only one A and one b for which $f(x) = Ax + b$ for all x.)

Hint. Show that the function $g(x) = f(x) - f(0)$ is linear.

You can think of an affine function as a linear function, plus an offset. In some contexts, affine functions are (mistakenly, or informally) called linear, even though in general they are not. (Example: $y = mx + b$ is described as 'linear' in US high schools.)

5. **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 Julia code to generate a time series z:

   ```
   t = collect(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

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

6. **Population dynamics.** An ecosystem consists of n species that interact (say, by eating other species, eating each other's food sources, eating each other's predators, and so on). We let $x(t) \in \mathbb{R}^n$ be the vector of deviations of the species populations (say, in thousands) from some equilibrium values (which don't matter here), in time period (say, month) t. In this model, time will take on the discrete values $t = 0, 1, 2, \ldots$. Thus $x_3(4) < 0$ means that the population of species 3 in time period 4 is below its equilibrium level. (It does not mean the population of species 3 is negative in time period 4.)

   The population (deviations) follows a discrete-time linear dynamical system, which means that $x(t+1)$ is determined by $x(t)$. That is, we can compute the entire sequence $x(0), x(1), x(2), \ldots$ from $x(0)$ by applying the iteration

   $$x(t+1) = Ax(t).$$

   We refer to $x(0)$ as the *initial population perturbation.*

   The questions below pertain to the specific case with $n = 10$ species, with matrix A given in `pop_dyn_data.json`.

   a) Suppose the initial perturbation is $x(0) = e_4$ (meaning, we inject one thousand new creatures of species 4 into the ecosystem at $t = 0$). How long will it take to affect the other species populations? In other words, report a vector s, where s_i is the smallest t for which $x_i(t) \neq 0$. (We have $s_4 = 0$).

b) *Population control.* We can choose any initial perturbation that satisfies $|x_i(0)| \leq 1$ for each $i = 1, \ldots, 10$. (We achieve this by introducing additional creatures and/or hunting and fishing.) What initial perturbation $x(0)$ would you choose in order to maximize the population of species 1 at time $t = 10$? Explain your reasoning. Give the initial perturbation, and using your selected initial perturbation, give $x_1(10)$ and plot $x_1(t)$ versus t for $t = 0, \ldots, 40$.