Alex Lemon
EE263 (Introduction to Linear Dynamical Systems)
Homework 1
July 1, 2013

1 Representing linear functions as matrix multiplication

Suppose $f : \mathbb{R}^n \to \mathbb{R}^m$ is linear. Find a matrix $A \in \mathbb{R}^{m \times n}$ such that $f(x) = Ax$ for all $x \in \mathbb{R}^n$. Is the matrix A unique? Give a proof or counterexample.

2 Affine functions

A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is said to be affine if

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

for all $x, y \in \mathbb{R}^n$ and $\alpha, \beta \in \mathbb{R}$ such that $\alpha + \beta = 1$.

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

(b) Suppose $f : \mathbb{R}^n \to \mathbb{R}^m$ is an affine function. Show that there exist a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $b \in \mathbb{R}^m$ such that $f(x) = Ax + b$. Are A and b unique?

3 Gradients of common functions

Recall that the gradient of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ at a point $x \in \mathbb{R}^n$ is defined to be the vector

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}.$$

The first-order Taylor expansion of f near x is given by

$$\hat{f}_1(z) = f(x) + \nabla f(x)^\mathsf{T}(z - x).$$

This function is affine: that is, a linear function plus a constant offset. If z is near x, then $\hat{f}_1(z)$ is very near $f(z)$. Find the gradients of the following functions.

(a) $f(x) = a^\mathsf{T} x + b$, where $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$

(b) $f(x) = x^\mathsf{T} A x$, where $A \in \mathbb{R}^{n \times n}$

(c) $f(x) = x^\mathsf{T} A x$, where $A \in \mathbb{S}^n$

4 Some matrices from signal processing

We can think of a vector $x \in \mathbb{R}^n$ as a representation discrete-time scalar signal, where x_t denotes the value of the signal at time t, for $t = 1, \ldots, n$. For each of the following transformations, find a matrix A such that $y = Ax$.

(a) $2\times$ *up-conversion with linear interpolation.* The output signal is $y \in \mathbb{R}^{2n-1}$ such that

$$y_t = \begin{cases} x_{\frac{1}{2}(t+1)} & t \text{ is odd,} \\ \frac{1}{2}(x_{\frac{1}{2}t} + x_{\frac{1}{2}t+1}) & t \text{ is even.} \end{cases}$$

Roughly speaking, this operation doubles the sampling rate, using linear interpolation to compute the new samples.

(b) $2\times$ *down-sampling.* For this part of the problem, we assume that n is even. The output signal is $y \in \mathbb{R}^{\frac{1}{2}n}$ such that $y_t = x_{2t}$.

(c) $2\times$ *down-sampling with averaging.* For this part of the problem, we assume that n is even. The output signal is $y \in \mathbb{R}^{\frac{1}{2}n}$ such that $y_t = \frac{1}{2}(x_{2t-1} + x_{2t})$.

5 Input-output matrix of a discrete-time linear dynamical system

Consider a discrete-time linear dynamical system:

$$\begin{aligned} x(t+1) &= A(t)x(t) + B(t)u(t), \\ y(t) &= C(t)x(t) + D(t)u(t). \end{aligned}$$

Find a matrix G such that

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

The matrix G shows how the sequence of outputs, $y(0), \ldots, y(T)$, depends on the initial state, $x(0)$, and the sequence of inputs, $u(0), \ldots, u(T)$.

6 A mass subject to applied forces

Consider a unit mass subject to a time-varying force $f(t)$ for $0 \leq t \leq n$. Let the initial position and velocity of the mass both be zero. Suppose that the force has the form $f(t) = x_j$ for $j-1 \leq t < j$ and $j = 1, \ldots, n$. Let y_1 and y_2 denote respectively the position and velocity of the mass at time $t = n$.

(a) Find a matrix $A \in \mathbb{R}^{2\times n}$ such that $y = Ax$.

(b) For $n = 4$, find a sequence of input forces $x_1, \ldots, x_n$ that moves the mass to position 1 with velocity 0 at time n.

7 Counting sequences in a language or code

We consider a language or code with alphabet $\mathbb{N}_n = \{1, \ldots, n\}$. A sentence is a finite sequence of symbols, $(k_1, \ldots, k_L)$, where $k_\ell \in \mathbb{N}_n$ for all $\ell = 1, \ldots, L$. A language or code consists of a set of sequences, which we will call the set of allowable sequences. A language is called Markov if the allowed sequences can be described by giving the allowable transitions between consecutive symbols: for each symbol, we give a set of symbols that are allowed to follow that symbol. As a simple example, consider a Markov language with $n = 3$

symbols. Suppose symbol 1 must be followed by 1 or 3; symbol 2 must be followed by 3; and symbol 3 must be followed by 1 or 2. The sentence $(1, 1, 3, 2, 3, 1, 3)$ is allowable (that is, in the language) because all the transitions in this sentence are allowed; the sentence $(1, 1, 3, 2, 3, 1, 2)$ is not allowable (that is, not in the language) because the final transition in this sentence is not allowed. We can describe the allowed transitions using a matrix $A \in \mathbb{R}^{n\times n}$ with

$$A_{ij} = \begin{cases} 1 & \text{symbol } i \text{ is allowed to follow symbol } j, \\ 0 & \text{symbol } i \text{ is not allowed to follow symbol } j. \end{cases}$$

(a) Give an interpretation of $(A^p)_{ij}$ (that is, the (i, j)-entry of A^p) for $p \in \mathbb{N}$.

(b) Consider a Markov language with $n = 5$ symbols, and the following transition rules:

- 1 must be followed by 2 or 3,
- 2 must be followed by 2 or 5,
- 3 must be followed by 1,
- 4 must be followed by 2, 4 or 5,
- 5 must be followed by 1 or 3.

Find the total number of sentences of length $L = 10$. Compare this to the total number of sequences of length L that can be formed from n symbols (that is, the total number of sentences of length L in a language with n symbols, and no restrictions on the allowable transitions.)