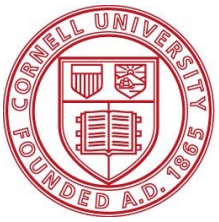

Homework # 10

Math 4310 Fall 2020

Due Friday evening, 12/4/20 (at midnight).

Please submit your completed homework via gradescope on canvas. I encourage you to work with your classmates on this homework (except for the journal entries!) When you submit your work, please **list your collaborators**. (Your grade will not be affected.) Even if you work in a group, you should write up your solutions **yourself**! You should include all computational details, and proofs should be carefully written with full details. As always, please write **neatly and legibly** (feel free to use LaTeX to write up your solutions, if you wish!).

Journal entry. There is no journal entry this week.

Exercises.

1. Let V be the vector space (over $\mathbb{C}$) of all continuous functions $f(x) = g(x) + ih(x)$ from the interval $[0, 2\pi]$ to the complex numbers. One example of a function in V is $e^{ix} = \cos x + i \sin x$. In this problem you may use the fact that the product and sum of two continuous functions is continuous. You may also use the fact that a continous function which is always nonnegative, but takes on a positive value somewhere, has non-zero integral.

   Integration works for these functions pretty much like over the real numbers, for example, if $m \neq 0$ is an integer, then

$$\int_0^{2\pi} e^{imx}\, dx = \frac{1}{im} e^{imx}|_0^{2\pi} = -\frac{i}{m}(1-1) = 0.$$

   (a) Let $\mathcal{A}_r = \{e^{imx} \mid -r \leq m \leq r, m \text{ an integer}\}$. For a positive integer r, let V_r be the subspace of V spanned by $\mathcal{A}_r$. Show that V_r is also the span (over $\mathbb{C}$) of $\mathcal{B}_r$:

$$\mathcal{B}_r := \{\cos(mx) \mid 0 \leq m \leq r\} \cup \{\sin(mx) \mid 1 \leq m \leq r\}.$$

   (b) Define $\langle f, g\rangle = \int_0^{2\pi} f(x)\overline{g(x)}\, dx$. Prove that this is an inner product on V.

   (c) Find an orthonormal basis for V_r.

   (d) Given a continuous function $f(x) \in V$, let $a_m := \int_0^{2\pi} f(x)e^{imx}\, dx$. In terms of these numbers, find the closest function (in the norm corresponding to the above inner product) in V_r to $f(x)$.

   (e) Given a continuous function over $\mathbb{R}$, $f(x)$, let $c_m := \int_0^{2\pi} f(x)\cos mx\, dx$, and let $s_m := \int_0^{2\pi} f(x)\sin mx\, dx$. How are the a_m and c_m, s_m related? Write the closest function in terms of this new basis.

2. In this problem, we consider the QR factorization of a matrix A (with entries in $\mathbb{R}$), with linearly independent columns. Suppose that $A = \begin{bmatrix}\mathbf{a}_1 \dots \mathbf{a}_n\end{bmatrix}$ is an $m \times n$ matrix over $\mathbb{R}$ with linearly independent columns, and that $Q = \begin{bmatrix}\mathbf{u}_1 \dots \mathbf{u}_n\end{bmatrix}$ is obtained from A by using the Gram-Schmidt process.

(a) Show that there is a $n \times n$ invertible matrix S such that $AS = Q$.

(b) Show that there is exactly one $n \times n$ matrix R, such that $A = QR$.

(c) Show that R is upper triangular and invertible. What are the entries of R?

(d) Find $A = QR$, for

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

(e) Recall that the least squares solution of $A\mathbf{x} = \mathbf{y}$ is the vector $\mathbf{x}$ such that $\|A\mathbf{x} - \mathbf{y}\|$ is minimal, and that this is given by the unique exact solution to the normal equation:

$$A^T A\mathbf{x} = A^T\mathbf{y}.$$

Assume that $A = QR$ is a QR factorization of A. Rewrite the normal equations using Q and R, simplifying as much as possible. Explain how to get the solution $\mathbf{x}$ by only solving upper or lower triangular systems, not general systems (once the Q and R have been found!).

3. **Householder (reflection) matrices**. Let $V = \mathbb{R}^n$, equipped with the standard inner product. Given a nonzero vector v and unit vector $u = \frac{v}{\|v\|}$, let $H \in \mathbb{R}^{n\times n}$ be the square matrix defined by

$$H = I - 2\frac{vv^T}{v^Tv} = I - 2uu^T.$$

Notice that the displayed fraction is a matrix divided by a scalar, so H is an $n \times n$ matrix.

(a) Show that H is symmetric, and H is an orthogonal $n \times n$ matrix. What is the inverse of H?

(b) Show that H is a reflection: $H(v) = -v$, and if $w \in v^\perp$, then $H(w) = w$.

4. (Rank one update) Suppose that A is a square invertible matrix of size $n \times n$ over $\mathbb{R}$. Let $\mathbf{u}$ and $\mathbf{v}$ be $n \times 1$ column vectors. Let $B = A + \mathbf{u}\mathbf{v}^T$, this is called **a rank one update to the matrix** A. (This is since $\mathbf{u}\mathbf{v}^T$ is a rank one $n \times n$ matrix).

(a) Find the real number a such that

$$(A + \mathbf{u}\mathbf{v}^T)^{-1} = A^{-1} - aA^{-1}\mathbf{u}\mathbf{v}^TA^{-1}.$$

(actually, sometimes this a cannot be found, when does this happen? in this case the matrix B might not be invertible).

(b) Show that one can solve linear systems of the form $B\mathbf{y} = \mathbf{b}$ as follows: First solve $A\mathbf{x} = \mathbf{b}$ and $A\mathbf{z} = \mathbf{u}$. Compute $D = 1 + \mathbf{v}^T\mathbf{z}$ (this is a number). Assume that $D \neq 0$. Use this data to solve $B\mathbf{y} = \mathbf{b}$.

5. In supervised machine learning, a neural network consists of a bunch of "neurons". The whole thing is a function: you give it an input, say a vector of real numbers $\mathbf{t} = (t_1, \ldots, t_n)$, and out pops a result y. Each neuron has a number of parameters, and when you "learn" the network, you give it a lot of input vectors, and the output that is desired . Using all this data, an algorithm will tweak all of the parameters in the neurons to minimize the error: i.e. try to match the data you gave it.

   In this problem, we consider the very special (yet still important) case of a single neuron. A single neuron in a neural network is a function which has n real number inputs, which we represent as the $1 \times n$ matrix

   $$\mathbf{t} = \begin{bmatrix} t_1 & t_2 & \ldots & t_n \end{bmatrix}.$$

   Internally, the neuron (function) has n "weights" x_i, which we represent as a column vector

   $$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}.$$

   The neuron, when given input $\mathbf{t}$, outputs

   $$\mathbf{t}\,\mathbf{x} = \sum_{i=1}^{n} x_i t_i,$$

   the dot product of these two vectors. (Actually, it usually outputs something slightly more complicated, but we won't consider that here).

   Suppose that we have training data: the input $\mathbf{t}_i = \begin{bmatrix} t_{i1} & \ldots & t_{in} \end{bmatrix}$, with output y_i, for $i = 1..m$, with m generally much larger than n. If we let A be the $m \times n$ matrix with rows $\mathbf{t}_1, \ldots, \mathbf{t}_m$, and we let $\mathbf{y}$ be the $m \times 1$ column vector with entries y_i, then the optimal $\mathbf{x}$ which minimizes the error of the training data is solved by the normal equations $A^T A\mathbf{x} = A^T\mathbf{y}$. We assume that the columns of A are linearly independent.

   We will now investigate what happens when we have already found the weights $\mathbf{x} = \mathbf{x}_{old}$ which best fit the training data, but we get new training data. For the purposes of this problem, we will assume we have one new data point:

   $$\mathbf{r} = \mathbf{t}_{m+1},$$

   a $1 \times n$ matrix, and it's output y_{m+1}. Our goal is to update $\mathbf{x}_{old}$ to get $\mathbf{x}_{new}$, without starting from scratch and redoing the entire computation.

   (a) The new matrix is $A' = \begin{bmatrix} A \\ \mathbf{r} \end{bmatrix}$, and the normal equations are:

   $$(A')^T A'\mathbf{x} = (A')^T \begin{bmatrix} \mathbf{y} \\ y_{m+1} \end{bmatrix}.$$

   Simplify this equation, obtaining a system of n equations in n unknowns:

   $$B\mathbf{x}_{new} = \mathbf{y}',$$

   where B is a rank one update of $A^T A$.

(b) Let $\mathbf{z}$ be the solution to $A^T A\mathbf{z} = \mathbf{r}^T$. Use the previous problem to determine a solution for $\mathbf{x}_{new}$ of the form:

$$\mathbf{x}_{new} = \mathbf{x}_{old} + b\,\mathbf{z}.$$

What is the real number b?

(c) (optional, open-ended): Is this worth it? Does this take less computation than the full solving? (recall: multiplication of $n \times n$ matrices takes n^3 operations, as does solving a general linear system of n equations and n unkowns. It is possible to do better, by the way. Solving upper or lower triangular systems requires on the order of n^2 operations).