CSci 5304 Homework 1 (draft) Due Fri 9/17/2021

This homework is intended to check your preparation for this class. Some questions are exploratory, to see whether your previous expertise can be applied. It will also test that you have have figured out the software environment needed for this class:

- Check that you have installed or access to Matlab, and can write simple programs therein. For most situations in this class, it will be OK to use Octave in place of Matlab.
- Check that you can access the Canvas environment and submit assignments.
- Use the ordinary builtin double precision floating point unless otherwise specified.
- Collect all your written answers plus the output of the programs into a single PDF file. Collect all your code (if any) into a separate zip file using 'zip' or 'pkzip'. Submit both files through Canvas. Do not use any other archiver such as rar, tar, bzip, 7z, etc. Do not submit word files.

Problem 1:

Solve the linear system of equations

$$\begin{aligned} 9.1x_1 + 6.6x_2 &= 2.0 \\ 4.6x_1 + 3.3x_2 &= 1.0 \end{aligned} \qquad \begin{array}{ccccc} A & & \mathbf{x} & = & \mathbf{b} \\ \begin{pmatrix} 9.1 & 6.6 \\ 4.6 & 3.3 \end{pmatrix} & & \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} & = & \begin{pmatrix} 2.0 \\ 1.0 \end{pmatrix} \end{array}$$

using the restrictions as outlined below. Use the gaussian elimination algorithm in which the first step is to eliminate x_1 from the second equation by adding in a multiple of the first equation.

1. Use only 2 digit arithmetic. That is: round the result of every individual arithmetic operation to 2 significant decimal digits. Show all your intermediate results, including the equations or matrix forms obtained when you have eliminated one variable. Denote the solution computed this way as $\mathbf{x}_c$.
2. Using 3 digit arithmetic, compute the residual $\mathbf{r} = \mathbf{b} - A\mathbf{x}_c$. Here you should keep an extra digit of significance temporarily only during this computation of the residual.
3. (exploratory) Can you think of way to estimate the error in the answer $\mathbf{x}_c$ using the residual, but still keeping only 2 digits in every calculation.

This is a toy version of what happens in real life, where accuracy must be estimated using the same approximate floating point arithmetic used to obtain the original answers.

Problem 2: Suppose you would like to predict a value y based on a vector of m feature values $\mathbf{x} = (x_1, \ldots, x_m)^T$. The standard linear regression problem finds a set of weights $\mathbf{w} = (w_1, \ldots, w_m)^T$ such that the linear combination $\langle \mathbf{x}, \mathbf{w} \rangle = w_1 x_1 + \cdots + w_m x_m$ is as close to y as possible. How "close" is measured in terms of a given training set with n samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and associated target values $\{y_1, \ldots, y_n\}$. We try to minimize the sum of squares of the discrepancies between the predictions and the targets over the training set:

$$\min_{\mathbf{w}} \sum_i \left(\langle \mathbf{x}_i, \mathbf{w} \rangle - y_i \right)^2 = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2, \tag{1}$$

where

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{pmatrix} = \begin{pmatrix} [\mathbf{x}_1]_1 & \cdots & [\mathbf{x}_1]_m \\ \vdots & \vdots & \vdots \\ [\mathbf{x}_n]_1 & \cdots & [\mathbf{x}_n]_m \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

Here $[\mathbf{x}_i]_j \equiv x_{ij}$ is the j-th component in the feature vector $\mathbf{x}_i$ for the i-th sample in the training set.

Using the matrix notation, The standard linear regression (1) can be formulated as solving the following minimzation problem:

$$\underset{\mathbf{w}}{\text{minimize}} \quad \phi(\mathbf{w}) = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 = \langle \mathbf{X}\mathbf{w} - \mathbf{y}, \mathbf{X}\mathbf{w} - \mathbf{y} \rangle \tag{2}$$

where $\mathbf{X} \in \mathbb{R}^{n \times m}$ $(n \geq m)$ represents the feature matrix, $\mathbf{y} \in \mathbb{R}^{n \times 1}$ represents the response vector and $\mathbf{w} \in \mathbb{R}^{m \times 1}$ is the vector variable of the linear coefficients. Here the $i-j$-th element of $\mathbf{X}$, denoted $x_{ij} = [\mathbf{x}_i]_j$, is the j-th attribute value for the i-th data sample (observation) and y_i is the true response for the i-th data sample.

Suppose $\mathbf{w}^*$ is a putative solution to (2). This means $\phi(\mathbf{w}^* + \boldsymbol{\delta})$ must be greater than or equal to $\phi(\mathbf{w}^*)$ for <u>any</u> $\boldsymbol{\delta} \in \mathbb{R}^{m \times 1}$. Expanding the formula for ϕ yields

$$\begin{aligned} \phi(\mathbf{w}^* + \boldsymbol{\delta}) &= [\mathbf{X}(\mathbf{w}^* + \boldsymbol{\delta})]^T \mathbf{X}(\mathbf{w}^* + \boldsymbol{\delta}) - 2\,[\mathbf{X}(\mathbf{w}^* + \boldsymbol{\delta})]^T \mathbf{y} + \mathbf{y}^T\mathbf{y} \\ &= \phi(\mathbf{w}^*) + 2\boldsymbol{\delta}^T \mathbf{X}^T [\mathbf{X}\mathbf{w}^* - \mathbf{y}] + (\mathbf{X}\boldsymbol{\delta})^T \mathbf{X}\boldsymbol{\delta} \\ &\geq \phi(\mathbf{w}^*). \end{aligned} \tag{3}$$

Here $\mathbf{X}^T$ denotes the transpose of $\mathbf{X}$, and we have used $[\mathbf{X}\boldsymbol{\delta}]^T = \boldsymbol{\delta}^T \mathbf{X}^T$.

1. We seek the conditions that $\mathbf{w}^*$ must satisfy in order to guarantee the inquality in (3) for <u>any</u> vector $\boldsymbol{\delta}$. Since the term $(\mathbf{X}\boldsymbol{\delta})^T \mathbf{X}\boldsymbol{\delta}$ is never negative, it suffices to find the conditions on $\mathbf{w}^*$ such that $2\boldsymbol{\delta}^T \mathbf{X}^T [\mathbf{X}\mathbf{w}^* - \mathbf{y}]$ is never negative regardless of your choice for $\boldsymbol{\delta}$. Can you find such conditions?

2. Solve this least squares regression problem for the following toy problem with 4 samples, each with 3 features and one target (i.e., find $\mathbf{w}^*$):

$$\mathbf{X} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} 1 \\ 4 \\ 3 \\ 2 \end{pmatrix}$$

Problem 3: Consider the ordinary differential equation

$$y' = f(t, y) = 100t - 100y + 101, \qquad y(0) = 0.0 \tag{4}$$

Implement the following algorithm (Euler's method) in Matlab. At each time step t you know the value $y(t)$ and hence its derivative from the differential equation $y'(t) = f(t, y(t))$. So follow the slope for a small time-step h to obtain $y(t + h) \approx y(t) + h \cdot y'(t)$.

```
function table = solve_euler(f,t0,y0,h,t_final,step)
   % input:
   %   f = fcn
   %   t0 = initial time
   %   y0 = initial value
   %   h = stepsize
   %   t_final = final_time (so number of steps = (t_final-t0)/h)
   %   step = recording interval:  save values at times which are multiples of step
   % output:  table of values at the times specified by "step".

   t = t0;
   y = y0;
   table = []
   while t < t_final
       if ...  fill in* ...  %[(t-t0) is a multiple of step]:
           table = [table ; [t, y]]; % append [t,y] to table
       y = y + h * f(t,y);
       t = t + h;
   table = [table ; [t, y]]; % append [t,y] to table and return.
   end
```

Show the results of calling this function with

```
fcn = @(t,y) 100*t - 100*y + 101
table=solve_euler(fcn,0,0,.01,1,0.1)
table=solve_euler(fcn,0,0,.02,1,0.1)
table=solve_euler(fcn,0,0,.025,1,0.1).
```

In each case, the result should be a table with 11 rows of the form shown here listing the computed values at time intervals of 0.1 (do not show any of the other intermediate time steps that may be computed). $\Longrightarrow$

t	computed y		
	$h = .01$	$h = .02$	$h = .025$
0.0	0.0000	0.0000	0.0000
0.1	...	...	...
0.2	...	...	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$
1.0	...	...	...

*Fill in some code to check if `t-t0` is a multiple of `step`. You can use `round`, but allow for round-off.

Problem 4:

Consider two vectors $\mathbf{u}, \mathbf{v}, \in \mathbb{R}^n$, such that $\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1$, and a scalar $\alpha \neq 0$. Assume $n \geq 2$. As column vectors, $\mathbf{u}^T\mathbf{v}$ is their inner product and $\mathbf{u}\mathbf{v}^T$ is their outer product.

(a) What is $\det(\alpha \mathbf{u}\mathbf{v}^T)$?
(b) What is $\mathrm{Tr}(\mathbf{u}\mathbf{v}^T)$?
(c) what is $\det(I + \mathbf{u}\mathbf{v}^T)$?
(d) What is $\mathrm{Tr}(I + \mathbf{u}\mathbf{v}^T)$?
(e) What real unit vector achieves the maximum in $\max_{\|\mathbf{x}\|_2=1} \|\mathbf{u}\mathbf{v}^T\mathbf{x}\|_2$? What is the resulting maximum value? A "unit vector" here is any vector of length 1: $\mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\|_2 = 1$.
(f) What are all the eigenvalues of $\mathbf{u}\mathbf{v}^T$ (with multiplicities)?
(g) Find conditions on the unit vector $\mathbf{u}, \mathbf{v}$ such that all the eigenvalues of $\mathbf{u}\mathbf{v}^T$ are 0, or show no such vectors exist. If such vectors exist, what is the multiplicity of eigenvalue 0 (algebraic & geometric)?

Hints: The trace is the sum of the diagonal entries and also the sum of the eigenvalues. The determinant is the product of the eigenvalues (among other properties). An eigenvalue's multiplicity is that as a root of the characteristic polynomial (algebraic) or number of linearly independent eigenvectors (geometric). The "2-norm" of a real vector is $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^T\mathbf{x}} = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$.

Now assume the dimension $n = 2$.

(h) Let $\mathbf{u} = (u_1,\, u_2)^T$ be a real unit vector. Find a unit vector $\mathbf{v} = (v_1,\, v_2)^T$ such that $\mathbf{u}^T\mathbf{v} = 0$. What is the inverse of the 2×2 matrix

$$A = (\mathbf{u},\, \mathbf{v}) = \begin{pmatrix} u_1 & v_1 \\ u_2 & v_2 \end{pmatrix}?$$

You can try the specific numerical values $\mathbf{u} = (0.6,\, 0.8)^T$.

Problem 5:

Let A be a real unitary matrix with $\det(A) = 1$. Let B be another unitary real matrix with $\det(B) = -1$. What is the numerical value of the determinant of $(A + B)$? [Hint: multiply by A^T on the left and B^T on the right and then get numerical value for the determinant by using properties of determinants (products, transpose...)]

end of document