All hw should be uploaded to canvas as a *pdf*. Make sure that if you scan your handwritten notes that they are legible and appropriately oriented. If you use an online resource to solve any problem, please appropriately cite that source.

Problem 1. (Least Angle Property of Least Squares.) Suppose the $m \times n$ matrix A has linearly independent columns, and $b \in \mathbb{R}^m$ is an m-dimensional vector. Let $\hat{x} = A^\dagger b = (A^\top A)^{-1} A^\top b$ denote the least squares approximate solution of $Ax = b$.

a. Show that for any $x \in \mathbb{R}^n$, the following equality holds:

$$(Ax)^\top b = (Ax)^\top (A\hat{x}) \quad \text{(i.e. } \langle Ax, b\rangle = \langle Ax, A\hat{x}\rangle\text{)}$$

Hint: Use the following facts.

- $(Ax)^\top b = x^\top (A^\top b)$ (i.e., the the transpose of a scalar is equal to itself)
- $(A^\top A)\hat{x} = A^\top b$ (i.e., $(A^\top A)A^\dagger b = (A^\top A)(A^\top A)^{-1} A^\top b = A^\top b$)

b. Show that when $A\hat{x}$ and b are both not identically zero, we have that

$$\frac{(A\hat{x})^\top b}{\|A\hat{x}\| \|b\|} = \frac{\|A\hat{x}\|}{\|b\|}$$

Note that the left hand side is the cosine of the angle between $A\hat{x}$ and b. Hint: apply part (a) with $x = \hat{x}$.

c. The choice $x = \hat{x}$ minimizes the distance between Ax and b. Under the assumption that Ax and b are not identically zero, show that $x = \hat{x}$ also minimizes the angle between Ax and b.

Problem 2. (Least Squares and QR Factorization.) Suppose $A \in \mathbb{R}^{m \times n}$ is a matrix with linearly independent columns and QR factorization $A = QR$, and $b \in \mathbb{R}^m$. We saw in **Mod2-Lec1** that given a least squares approximate solution $\hat{x}$, the vector $A\hat{x}$ is the linear combination of the columns of A that is closest to the vector b, i.e., it is the projection of b onto the set of linear combinations of the columns of A.

a. The matrix $QQ^\top$ is called the projection matrix. Show that $A\hat{x} = QQ^\top b$.

b. Show that $\|A\hat{x} - b\|^2 = \|b\|^2 - \|Q^\top b\|^2$.

Problem 3. (Weighted Least Squares.) As we saw in **Mod2-Lec1**, the least squares objective can be written in terms of the sum of residuals using the row formulation. That is, the objective is

$$\|Ax - b\|^2 = \sum_{i=1}^{m} (\tilde{a}_i^\top x - b_i)^2,$$

where $\tilde{a}_i^\top$ are the rows of A, and the vector $x \in \mathbb{R}^n$ is the decision variable. In the weighted least squares problem, we minimize the objective

$$\sum_{i=1}^{m} w_i (\tilde{a}_i^\top x - b_i)^2$$

where w_i are the given positive weights. The weights allow us to assign different weights to the different components of the residual vector. (The objective of the weighted least squares problem is the square of the weighted norm, $\|Ax - b\|_w^2$, as defined in exercise 3.28 VMLS.)

a. Show that the weighted least squares objective can be expressed as $\|D(Ax-b)\|^2$ for an appropriate diagonal matrix D. This allows us to solve the weighted least squares problem as a standard least squares problem, by minimizing $\|Bx-d\|^2$, where $B = DA$ and $d = Db$.

b. Show that when A has linearly independent columns, so does the matrix B.

c. The least squares approximate solution is given by $\hat{x} = (A^\top A)^{-1}A^\top b$. Write out a similar formula for the solution of the weighted least squares problem. **Hint**: use the matrix $W = \text{diag}(w)$ in your formula.

Problem 4. (Iterative Method for Least Squares.) Suppose that A has linearly independent columns, so $\hat{x} = A^\dagger b$ minimizes $\|Ax-b\|^2$. In this exercise we explore an iterative method, due to the mathematician Lewis Richardson, that can be used to compute $\hat{x}$. We define $x^{(1)} = 0$ and for $k = 1, 2, \ldots,$

$$x^{(k+1)} = x^{(k)} - \mu A^\top (Ax^{(k)} - b)$$

where $\mu > 0$, and the superscripts indicate the iteration number. This defines a sequence of vectors that converge to $\hat{x}$ provided μ is not too large; the choice $\mu = 1/\|A\|^2$, for example, always works. The iteration is terminated when $A^\top(Ax^{(k)} - b)$ is small enough, which means the least squares optimality conditions are almost satisfied. To implement the method we only need to multiply vectors by A and by $A^\top$. Iterative methods are often used for very large scale least squares problems.

a. Show that if $x^{(k+1)} = x^{(k)}$, we have $x^{(k)} = \hat{x}$.

b. Express the vector sequence $x^{(k)}$ as a linear dynamical system with constant dynamics matrix and offset, i.e., in the form $x^{(k+1)} = Fx^{(k)} + g$.

c. Generate a random 20×10 matrix A and vector $b \in \mathbb{R}^{20}$, and compute $\hat{x} = A^\dagger b$. Run the Richardson algorithm with $\mu = 1/\|A\|^2$ for 500 iterations, and plot $\|x^{(k)} - \hat{x}\|^2$ to verify that $x^{(k)}$ is converging to $\hat{x}$. Just include your error plots. Plot on both a linear and log-scale. To get the log-scale you can use `plt.yscale('log')`. Include a snap shot of your code in your homework pdf. You can print your python notebook to a pdf.

Problem 5. (Fitting a rational function with a polynomial..) We saw in the Jupyter Notebook **Mod2-Lec2** an example of fitting a randomly generated polynomial with polynomial basis functions. In this problem, we will repeat this exercise for a rational function. Include all your plots in the submitted pdf of your solution.

Let $x_1, \ldots, x_{11}$ be 11 points uniformly spaced in the interval $[-1, 1]$—i.e., $x_i = -1.0 + 0.1(i-1)$ for $i = 1, \ldots, 11$). Moreover, let

$$y_i = \frac{(1+x_i)}{(1+5x_i^2)}, \quad i = 1, \ldots, 11.$$

a. Find the least squares fit of polynomials of degree $0, 1, \ldots, 8$ to these points. Plot the fitting polynomials, and the true function

$$y = \frac{(1+x)}{1+5x^2},$$

over the interval $[-1.1, 1.1]$ using 100 points. Note that the interval for the plot, $[-1.1, 1.1]$ extends outside the range of the data used to fit the polynomials—i.e., $[-1, 1]$; this gives us an idea of how well the polynomial fits can extrapolate.

b. Generate a test data set by choosing $u_1, \ldots, u_{10}$ uniformly spaced over $[-1.1, 1.1]$ and generating

$$v_i = \frac{(1+u_i)}{(1+5u_i^2)} \quad \text{for each } u_i$$

Plot the `RMS` error of the polynomial fits found above on this test data set. On the same plot, show the `RMS` error of the polynomial fits on the training data set. Suggest a reasonable value for the degree of the polynomial fit, based on the `RMS` fits on the training and test data. Remark.

There is no practical reason to fit a rational function with a polynomial. This exercise is only meant to illustrate the ideas of fitting with different basis functions, over-fit, and validation with a test data set.

Problem 6. (Estimating a matrix.) Suppose that the vector $x \in \mathbb{R}^n$ and the vector $y \in \mathbb{R}^m$ are thought to be approximately related by a linear function, i.e., $y \approx Ax$, where $A \in \mathbb{R}^{m\times n}$. We do not know the matrix A, but we do have observed data,

$$x^{(1)}, \ldots, x^{(N)} \quad \text{and} \quad y^{(1)}, \ldots, y^{(N)}.$$

We can estimate or guess the matrix A by choosing it to minimize

$$\sum_{i=1}^{N} \|Ax^{(i)} - y^{(i)}\|^2 = \|AX - Y\|^2$$

where

$$X = \begin{bmatrix} | & \cdots & | \\ x^{(1)} & \cdots & x^{(N)} \\ | & \cdots & | \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} | & \cdots & | \\ y^{(1)} & \cdots & y^{(N)} \\ | & \cdots & | \end{bmatrix}$$

We denote this least squares estimate as $\hat{A}$. Show that $\hat{A} = YX^\dagger$ (where $X^\dagger = X^\top(XX^\top)^{-1}$, assuming the rows of X are linearly independent. **Hint**: Use the fact $\|AX - Y\|^2 = \|X^\top A^\top - Y^\top\|^2$, which turns the problem into a matrix least squares problem; see page 233 (VMLS). Then, define $W = A^\top$ and rewrite $\|X^\top A^\top - Y^\top\|^2$ as a sum in terms of the columns of W. Find the optimal solution for each of the terms in the sum.

Problem 7. (Multi-Class Classification via Matrix Least Squares.) We just saw an example of matrix least squares in **Problem 6**. Consider the least squares multi-class classifier described in §14.3 VMLS, with a regression model

$$\tilde{f}_k(x) = x^\top \beta_k$$

for the one-versus-others classifiers. We assume that the offset term is included using a constant feature. Show that the coefficient vectors $\beta_1, \ldots, \beta_K$ such that $\beta_i \in \mathbb{R}^n$ for each $i = 1, \ldots, K$ can be found by solving the matrix least squares problem of minimizing $\|X^\top\beta - Y\|$, where $\beta \in \mathbb{R}^{n\times K}$ is the matrix with columns $\beta_1, \ldots, \beta_K$, and $Y \in \mathbb{R}^{N\times K}$. That is,

a. Define the matrix Y—i.e., what is the i-th row of Y?

b. Write $\|X^\top\beta - Y\|^2$ as a sum of norms defined in terms of the columns of β and columns of Y. This is exactly the least squares problem for the Boolean class-k-versus-not-k classifier.

c. Use this form of the objective to show that the solution to the matrix least squares problem is $\hat{\beta} = (X^\top)^\dagger Y$ (**Hint**: cf. page 233 VMLS).

Problem 8. (Polynomial Classifier.) Using Python, generate 200 points $x^{(1)}, \dots, x^{(200)}$, uniformly spaced on the interval $[-1, 1]$, and take

$$y^{(i)} = \begin{cases} +1, & -0.5 \leq x^{(i)} < 0.1 \text{ or } 0.5 \leq x^{(i)} \\ -1, & \text{otherwise} \end{cases}$$

for $i = 1, \dots, 200$. Complete the following steps.

a. Fit polynomial least squares classifiers of degrees $0, \dots, 8$ to this training data set. **Hint**: Modify the code provided in `Mod2-Lec2.ipynb`, Example 1.

b. For each degree, plot the true y values, the polynomial $\tilde{f}(x)$ and the classifier $\hat{f}(x) = \text{sign}(\tilde{f}(x))$. **Hint**: you simply need to compute the sign of the estimate from part (a).

c. Evaluate the error rate on the training data set—i.e., generate a plot of the error rate. Does the error rate decrease when you increase the degree? **Hint**: Use the code from `Mod2-Lec3.ipynb` for computing the error rate. To use this error code, you will need to convert your predicted $\hat{y}$ and original y to boolean valued vectors.