USC CSCI 467 Intro to Machine Learning	**Assignment 0** Due: August 31, 2023, 11:59pm	Fall 2023 Instructor: Robin Jia

This assignment is primarily designed to remind you of various prerequisites that will be important for this class. If you look at these problems and find that you don't remember how to do everything, that is expected. If you look at these problems and find the concepts to be unfamiliar to you, there may be prerequisites that you should consider taking before you enroll in this class. This assignment has 4 questions, for a total of 50 points.

When submitting on Gradescope, note that **you must make a submission both for the written portion and programming portion.** For the programming portion, upload the file `q4_numpy.py` with your completed solutions. (There is an "autograder" which will not actually grade your code but it will run it, and should return 0 if it encountered an error and 100 otherwise.) **Please still include the output of your code in the PDF report when requested in the problems.**

Question 1: Probability (12 points)

(a) (Bayes Rule) A company develops a test for COVID. If a patient has COVID, the test returns positive with probability .7. If a patient does not have COVID, the test returns positive with probability .01. Suppose that 2% of the population is currently infected with COVID. Compute:

i. (3 points) $P(\text{Patient has covid} \mid \text{Test is positive})$

ii. (3 points) $P(\text{Patient has covid} \mid \text{Test is negative})$

(b) (Random variables and expected value) Two classmates, Alice and Bob, are about to take an exam for the same class. Let A be the random variable denoting Alice's exam score (between 0 and 100), and let B be the random variable denoting Bob's exam score.

i. (3 points) Suppose you knew that $P(A \geq 90) = 0.6$ and $P(B \geq 90) = 0.3$. Do you have enough information to compute $P(A \geq 90 \text{ and } B \geq 90)$? If so, compute it. Either way, justify your answer in 1-2 sentences.

ii. (3 points) Suppose you knew that $\mathbb{E}[A] = 92$ and $\mathbb{E}[B] = 84$. Do you have enough information to compute $\mathbb{E}[A + B]$? If so, compute it. Either way, justify your answer in 1-2 sentences.

Question 2: Single-variable Calculus (14 points)

You find a mysterious coin on the ground and flip it five times. You observe the sequence Heads, Tails, Heads, Heads, Heads.

(a) (2 points) Let p be the (unknown) probability that the coin lands heads when flipped. Write an expression for the *likelihood function* $L(p)$, the probability of the observed data as a function of p.

(b) (4 points) Now, we will compute the value of p that maximizes $L(p)$. You should do this by taking the derivative of $L(p)$, setting it to 0, and solving for p. Remember to apply the second derivative test to check that you have found a local maximum and not a local minimum.

(c) (2 points) Provide an intuitive explanation for this optimal value of p.

(d) (4 points) There is a second way to compute the value of p that maximizes $L(p)$. We can first take the (natural) logarithm of $L(p)$, and then calculate the p that maximizes $\log L(p)$. Carry out this calculation by taking the derivative of $\log L(p)$, setting it to 0, and solving for p. Again, remember to apply the second derivative test.[1]

(e) (2 points) You should find that your answers to part (b) and part (d) are the same. In 1-2 sentences, explain why this is the case.

Question 3: Linear Algebra (12 points)

(a) (Vector geometry) Let $v = [1, -3, 5]$ and $w = [-2, 1, 2] \in \mathbb{R}^3$. Compute:

i. (2 points) The unit vector u in the same direction as w.

ii. (2 points) The (vector) component of v that is parallel to w.

iii. (2 points) The (vector) component of v that is perpendicular to w. Verify that this vector is in fact perpendicular to w.

(b) (6 points) A positive semi-definite matrix is a matrix $A \in \mathbb{R}^{d \times d}$ such that for all vectors $x \in \mathbb{R}^d$, $x^\top Ax \geq 0$. Prove that for any vector $u \in \mathbb{R}^d$, the matrix $uu^\top$ is positive semi-definite. (Hint: While you can show this by taking an eigendecomposition, there is a simpler proof.)

Question 4: Programming and Gradient Descent (12 points)

This problem will be a little different from the rest, as it is not about prerequisites but instead about gradients and doing linear algebra in numpy. Completing this problem will also ensure that you have a suitable python environment installed. You'll be ready to start this after Lecture 2.

numpy. numpy is a widely used python library for numerical linear algebra. We do not expect you to already be familiar with numpy; however, you should be comfortable writing python code. The advantage of using numpy, as opposed to writing linear algebra operations like matrix multiplication in raw python, is efficiency. Under the hood, numpy is implemented in C and makes use of fast linear algebra libraries.

Setup. We require python3 for this class. The assignments were developed with python 3.8.0 and numpy 1.23.4; other versions may work but we recommend using the same or similar versions if possible. You can install numpy by following these instructions: https://numpy.org/install/. Once you have installed numpy, download the starter code file called `hw0.zip`.

Problem. In this problem we will implement gradient descent in numpy to find the minimum of a function from $\mathbb{R}^2 \to \mathbb{R}$. In particular, let $u = \begin{pmatrix} 1 \\ 5 \end{pmatrix}$ and $v = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ (both are column vectors). Suppose we want to find a vector that has large dot product with u but also is close to v. We can formulate this as an optimization problem: define

$$f(x) = -u^\top x + \|x - v\|^2.$$

[1] As we will see soon in class, we often choose to work with the log of the likelihood function because it can be easier to differentiate and often has better mathematical properties.

We can now achieve our goal by minimizing f. Note the negative sign in front of $u^\top x$; because we want to maximize $u^\top x$, we want to minimize its negation. Here, $\|w\|$ denotes the Euclidean norm.

(a) (2 points) In `q4_numpy.py`, implement the function `f(x)`. Your solution should **not** use any for loops or list comprehensions. You may find the numpy function `numpy.sum()`, the `.dot()` method, and the `**` operators useful.

(b) (4 points) Now, compute the mathematical expression for $\nabla_x f(x)$.

(c) (2 points) In `q4_numpy.py`, implement the function `grad_f(x)`, which should return the expression you wrote in the previous part. Again, your solution should **not** use any for loops or list comprehensions.

(d) (4 points) Finally, in `q4_numpy.py`, implement the function `find_optimum()`. You should run gradient descent with the specified default learning rate and number of iterations. (You will need to use a single for loop for this.) Start at the initial value of $x = [0, 0]$. (You can use the `numpy.zeros()` method for this.) **Report the final value of x and $f(x)$ that you find.**