Homework 1

Due: Thursday, Feb 3, 2022 at 12:00pm **(Noon)**

Written Assignment

Introduction: Solidifying Background

The purpose of this portion is to fortify your background in probability and statistics, linear algebra, and algorithmic analysis. The topics explored here will be used many times throughout this course.

You may be able to find answers to these problems by searching the problem text. Please search instead for the concepts being applied; the goal is not to solve these specific problems, but to be comfortable with the principles that will be applied later in the course.

Problem 1: Bayes' Rule

(8 points)

Bayes' Rule, or Bayes' Theorem is an oft-used identity coming from probability theory. If we have two events of interest, A and B, we might want to ask what the probability of B is, given that we know A happened.

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

Note that this is the same as

$$P(B|A) = \frac{P(A \cap B)}{P(A)}.$$

Later in this course, the parts of this formula may be relabeled:

$$\text{Posterior} = \frac{\text{Likelihood} * \text{Prior}}{\text{Evidence}}$$

This rule will be explicitly used in Bayesian algorithms, but it is also a principle that will *implicitly* underlie almost all of our machine learning algorithms. This problem consists of four parts, each worth 3 points (1 point if the answer is correct and an additional 2 points for showing correct work). As a hint, none of the four parts have the same answer.

For the purposes of this question, assume that desserts have equal probability of being a cake or ice cream and uniform probability of being any of the following 7 flavors: chocolate, vanilla, strawberry, coconut, cookies & cream, fudge, and raspberry.

Fun ice cream fact: Vanilla is the superior ice cream flavor.

a. Suppose Steve has two desserts. What is the probability that both desserts are cakes?

b. Suppose Paul has two desserts, the first of which is ice cream. What is the probability that both desserts are ice cream?

c. Suppose Chace has two desserts and at least one is a cake. What is the probability that both desserts are cakes?

d. Suppose Andrew has two desserts and at least one is chocolate flavored ice cream. What is the probability that both desserts are ice cream?

Problem 2: Matrix Rank, Eigenvalues, and Eigenvectors

(12 points)

a. Consider a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$. Let $\text{rank}(\mathbf{A}) < n$. Prove that $\mathbf{A}$ has an eigenvalue equal to 0. Do not use the Invertible Matrix Theorem (e.g. https://mathworld.wolfram.com/InvertibleMatrixTheorem.html) when solving this problem.
Hint: recall that the rank of a matrix is equal to the number of linearly independent columns of the matrix.

b. Let $\mathbf{B} \in \mathbb{R}^{n \times n}$ be a symmetric matrix satisfying $x^T \mathbf{B} x > 0$ for all $x \in \mathbb{R}^n$ such that $x \neq \mathbf{0}$. Remember that the eigenvalues of a symmetric matrix are real. Prove that all of the eigenvalues of $\mathbf{B}$ are strictly greater than 0. Do not use any theorems for positive definite matrices. (4 points)

c. Let $\mathbf{C} \in \mathbb{R}^{n \times n}$ be a symmetric matrix. Assume that $\mathbf{C}$ has n unique eigenvalues. Prove that all of the eigenvectors of $\mathbf{C}$ are orthogonal. *Hint: begin your proof by considering the equations*

$$\mathbf{C}v_1 = \lambda_1 v_1$$
$$\mathbf{C}v_2 = \lambda_2 v_2$$

where (λ_1, v_1) and (λ_2, v_2) are two different eigenpairs of $\mathbf{C}$. Then, multiply the second equation by v_1^T.

Problem 3: Runtime Complexity

(14 points)

The Fibonacci sequence is defined as $F(n) = F(n-1) + F(n-2)$ for $n \geq 2$ where $F(0) = 0$ and $F(1) = 1$. We can make use of the fact (likely first noted by Edsger Dijkstra) that

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = \begin{pmatrix} F(n+1) & F(n) \\ F(n) & F(n-1) \end{pmatrix}$$

for any positive integer n. So, to compute $F(n)$ we can compute $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1}$ and return the upper left number in the matrix. Describe an algorithm to compute $F(n)$ for arbitrary n that uses *fewer than* $O(n)$ additions and multiplications (don't worry about the size of the integers), and prove that it satisfies this complexity bound. That is, prove a sublinear complexity bound in terms of n on the number of additions and multiplications this algorithm performs. Feel free to search for such an algorithm, but please generate the proof of the complexity bound yourself.

Problem 4: Numpy, Scipy, and Matplotlib

Introduction

The purpose of this section is to introduce you to some tools that you will find useful and/or necessary in order to complete future homeworks. By the end of this assignment, you will have used numpy to perform efficient computations, loaded standard datasets using SciPy, and used matplotlib to visualize several performance metrics you will be using this semester. This homework also serves as an environment/install test and will get you familiar with the hand-in process for physical documents.

Set up

Python 3.9, numpy, scipy, and matplotlib are considered necessary in order to do the homeworks in the remainder of this course. Fortunately, all of these can be installed on your own machine without admin privileges in a single package, called Anaconda. You can follow the directions at www.anaconda.com/download. Alternatively, you can install these packages using a Python virtual environment without Anaconda. Be sure to use Python version 3.9. For your convenience, we also have a course-wide virtual environment set up on the department machines at `/course/cs1420/cs142_env`. It can be activated from your own folder by running: source `/course/cs1420/cs142_env/bin/activate`. After this, you can run your program with the necessary environment/packages.

Stencil Code

Stencil code is available at github classroom, accessible at this link.

Part 1: Matplotlib

(8 points)

The code for this homework assignment is contained in the file `hw1.py`. The file has been included with this handout.

`hw1.py` contains two functions you need to fill out, `graph_iris_data` and `graph_series_data`. In order to fill out `graph_iris_data`. Note that the `plt.show()` call should be the last call, so add all of your graph customization below the TODO, but above the `show()` call!

1. call `plt.scatter(x, y, c=None)`, giving it the following arguments:
   (a) the `x` argument will be `xs`
   (b) the `y` argument will be `ys`
   (c) in order to give the plotted points color, we will specify the optional c argument. Thus, we will pass a third argument, `c=iris.target`
2. Look into the matplotlib.pyplot documentation and learn how to add titles and axis labels to plots. Add a title to the plot of the form "Made by: [Banner ID]".
3. Examine the column names of the iris data to find appropriate x and y labels for the plot, and use pyplot commands to label the axes of the two iris plots.

We will use a similar process for `graph_series_data`.

1. call `plt.plot(x, y, format)`, giving it the following arguments:
   (a) the `x` argument will be `xs`
   (b) the `y` argument will be `y1s`
   (c) the `format` argument will be `'.r'`
2. call `plt.plot(x, y, format)` again, giving it the following arguments:
   (a) the `x` argument will be `xs`
   (b) the `y` argument will be `y2s`
   (c) the `format` argument will be `'-b'`
3. Add a title to the plot of the form "Made by: [Banner ID]".

4. Use pyplot commands to add a legend to the series data plot, where each series is labeled with its function. Look here to start!

Afterwards, print the two graphs and turn these in with the rest of the assignment. Each graph is worth 4 points.

Part 2: Numpy

(8 points)

Answer the following questions using numpy functions. Note that when importing numpy, it is often abbreviated to np (i.e., `import numpy as np`), so when calling numpy functions, you can use np.[function]. Further, note that you may NOT use the np array constructor to solve these questions (i.e., `np.array(...)`). Some functions you may want to consider to approach this problem are `np.arange`, `np.zeros`, `np.ones`, `np.eye`, `np.sum`, `np.hstack`, `np.vstack`, `np.transpose`, `np.matmul`, `np.inner`, `np.where` and `np.dot`.

1. Using numpy, how would you create a 1D array containing the values 2 through 6 inclusive?
2. Using numpy, how would you create a 4x4 matrix where all the values are 1?
3. Using numpy, how would you create a 6x6 identity matrix?
4. Using numpy, how would you sum the values of each column of matrix A?
5. Using numpy, given the matrices `A` and `B`, how would you find the matrix `C`, where $\texttt{C} = \texttt{A}^T\texttt{B}$?
6. Using numpy, using either np.vstack or np.hstack, how would you create a 4x2 matrix where all the values in the first column are 0's and all the values in the second column are 1's?
7. Given a matrix of floats A, using numpy, how would you return an array of the same shape, where all values > 3.0 are set to 1, and the rest to 0?

Note that with anaconda set up, you have numpy downloaded, so you can check your answers by running python from the command line!

Problem 5: Critical Response Question

(5 points)

Read the short excerpt below from an article written by Ben Green where he talks of his views on the field of social good in Computer Science:

'Incrementalist "good" can lead to long-term harm. Although efforts to promote "social good" can be productive, computer science has thus far not developed a rigorous methodology for considering the relationship between algorithmic interventions and long-term social impact. The field takes for granted that, even if machine learning cannot provide perfect solutions to social problems, it can nonetheless contribute to "good" by making many aspects of society better. In fact, some computer scientists emphasize these immediate improvements over long-term considerations: arguing, for example, that "we should not let the perfect be the enemy of the good." This position assumes that because we all agree that crime, poverty, discrimination, and so on are problems, we should applaud any attempts to alleviate those issues. This orientation to producing technical reforms treats the "perfect" as an unrealistic utopia that, on account of its impossibility of being realized, is not worth articulating or debating.'

What are your initial thoughts on reading this passage? Do you agree or disagree with the author? Is there an effective way to question and repair unethical practices in the field of Computer Science without coming in the way of the overall good tech does for society? Please elaborate your answer.

Grading Breakdown

The grading breakdown for the assignment is as follows:

Problem 1	15%
Problem 2	22%
Problem 3	25%
Problem 4	29%
Problem 5	9%
Total	100%

Handing In

Important: You will need to fill out the Collaboration Policy Form available on the course website in order to hand in this assignment. Once you complete the form, you should receive an invitation to the Gradescope course within the next few hours. You will turn in your final handin via Gradescope. If you have questions about using Gradescope, please ask on Edstem. For this assignment, you should have written answers for Questions 1, 2, 3, 5, and the Numpy part of Question 4. For the Matplotlib portion of Question 4, you should turn in your two graphs.

Anonymous Grading

You need to be graded anonymously, so please do not write your name anywhere on your handin.

Obligatory Note on Academic Integrity

Plagiarism — don't do it.

As outlined in the Brown Academic Code, attempting to pass off another's work as your own can result in failing the assignment, failing this course, or even dismissal or expulsion from Brown. More than that, you will be missing out on the goal of your education, which is the cultivation of your own mind, thoughts, and abilities. Please review this course's collaboration policy and if you have any questions, please contact a member of the course staff.