Homework 0

Due: February 2, 2018 at 7:00PM

Written Questions

(30 points total + 3 extra credit)

Intro: 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, in addition to the practical skills developed in the programming section, will be used many times throughout this course.

Note 1: This section is not meant to take more than 2 hours. If you are not confident with any of the sections, or find yourself stuck, please make use of tutorials, Piazza, and TA hours.

Note 2: 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

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:

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

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 whales have equal probability of calving a male or female, and uniform probability of being born on any day of the week. Fun whale fact: No whale has ever been observed giving birth to twins.

Part 1

(3 points)
Suppose a whale has two offspring. What is the probability that both offspring are male?

Part 2

(3 points)
Suppose a whale has two offspring and the eldest is male. What is the probability that both offspring are male?

Part 3

(3 points)
Suppose a whale has two offspring and at least one is male. What is the probability that both offspring are male?

Part 4

(Extra Credit: 3 points)
Suppose a whale has two offspring and at least one is a male whale born on a Wednesday. What is the probability that both offspring are male?

Problem 2: Linear Algebra - Singular Value Decomposition

One goal of a singular value decomposition is to represent a large matrix as a product of smaller ones. See Figure **??**.

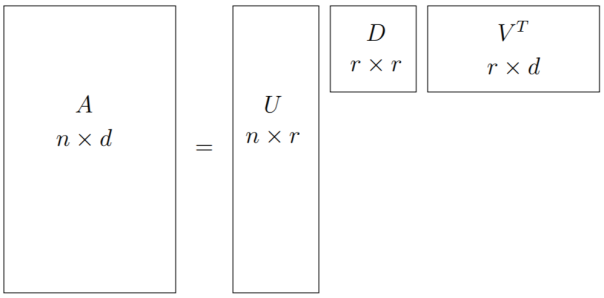

Figure 1: Singular Value Decomposition (SVD) represents a large matrix as a product of smaller ones. Note that the columns of U and V are orthonormal, and D is a real-valued diagonal matrix.

If we have an $n \times d$ matrix of n datapoints each in d dimensions, SVD finds a $d \times r$ matrix V that allows us to project the data to a smaller number of dimensions. This is the basis (pun intended) of Principal Component Analysis, as well as several other algorithms we will cover this semester. Note that V and U are orthonormal. An orthonormal matrix has the property that its columns are mutually orthogonal (the dot product of any pair of distinct columns is 0) and normalized (the dot product of any column with itself is 1). The combination of these properties means that the product of an orthonormal matrix's transpose with that matrix is an identity matrix.

$$V^T \times V = I$$
$$U^T \times U = I$$

This problem has 3 parts, each worth 3 points. Points will be awarded for clean derivations/proofs and for the compactness of the result (that is, the result should be expressed in simplest terms). Assume that expressions may freely include the transpose of any matrix, the identity matrix, and the zero matrix, in addition to any given terms.

Part 1

(3 points)
Let $\begin{bmatrix} A \\ A \end{bmatrix}$ be a $2n \times d$ matrix made by extending A with itself. Given a SVD of matrix A into $U \times D \times V^T$, where D is a diagonal matrix with dimension r, express SVD($\begin{bmatrix} A \\ A \end{bmatrix}$) in terms of U, D, and V. Provide an explanation for your answer.

Part 2

(3 points)
Given $A = U_A \times D_A \times V_A^T$, use V_A to reduce the dimensionality of A. That is, let $B = A \times V_A$. Express the SVD(B) in terms of U_A, D_A, and V_A. Provide a justification for your answer.

Part 3

(3 points)
Suppose we have $A = U_A \times D_A \times V_A^T$, and we let $B = A \times V_A$. In the same way, let $C = B \times V_B$. Write a SVD of C in terms of U_A, D_A, and V_A. Is there a maximum to the number of times the dimensionality of a dataset can be reduced using SVD?

Problem 3: Runtime Complexity

(12 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. 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.

Programming Assignment: Numpy, SciPy, matplotlib, and Notebooks

(20 points total)

Introduction

The purpose of this section is to introduce you to four 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 Jupyter notebooks to test and run code incrementally, 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.

Installing Anaconda

Python 3.6, numpy, scipy, and matplotlib are considered necessary in order to do the homeworks in the remainder of this course, while Jupyter notebooks are a useful scientific and development tool. 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. Be sure to use Python version 3.6. You will be able to launch a Jupyter notebook environment in your browser using the command line command `jupyter notebook`, or by launching the end-user executable of the same name.

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 `jupyter notebook` from the terminal to launch the notebook directory and environment.

Data

Open and execute the cells in the included `HW0.ipynb`. This will include further instructions for the completion of this assignment.

Grading Breakdown

As explained in `HW0.ipynb`, there are three components to the programming section. The first two are treated as a group and each is worth 5 points, 2 for handing in the correct part and 1 for each of three correct alterations. The final part is graded in the same way, but worth twice as much. These details will be made clear by examining the notebook file.

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.

As described in `HW0.ipynb`, the result of the programming component should be printed out and handed in along with a paper copy of your answers to the math component. These should be stapled together and labeled with the homework number, due date and your unique ID for this course (which you will get from the Collaboration Policy Form). In order to be graded anonymously, you should not write your name anywhere on your handin. The stapled packet should be left in the handin box for CSCI 1420 located in the CIT.

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.