CS1675: Homework 1

Due: 9/6/2018, 11:59pm

This assignment is worth 40 points.

Part I: Matlab and linear algebra (36 pts)

Write your code in a script called `hw1.m`, except where noted. Each item is worth 3 points.

Since one optimization that Matlab uses is to vectorize code rather than use loops, pay close attention to whether you are allowed to use loops. If this is not specified, you can (but don't have to) use loops. Remember to use the Matlab documentation whenever needed.

Basic operations:

1. Generate a 1000000x1 (one million by one) vector of random numbers from a Gaussian (normal) distribution with mean of 0 and standard deviation of 5. Use Matlab's `randn` function.
2. Add 1 to every value in the previous list, by using a loop. To determine how many times to loop, use Matlab's `size` function. Time this operation and print the number in the code. (Use Matlab's documentation to find out how to time operations.)
3. Now add 1 to every value in the original random vector, without using a loop. Time this operation, print the time and write it down. Use a different way to print the number than the method you used above. (See ways to print numbers at the beginning of the Matlab tutorial script.)
4. Plot the exponential function `2.^x`, for non-negative *even* values of x smaller than 100, without using loops.
5. [4 pts] Create two matrices `A` and `B` which, when added together, result in a 10x10 matrix `C` containing all numbers from 1 to 100. In other words, when I add `A` and `B` and convert the result to vector form, I should get a vector containing all numbers from 1 to 100. In code, `C = A + B; assert(all(C(:) == (1:100)') == 1);` Each matrix should only be created with a single command (no loops).

Linear algebra:

6. Generate two random matrices `A` and `B`, and compute their product by hand, using loops. The size of the first matrix should be [5, 3] and of the second [3, 5]. Check your code by comparing the loop-computed product with the product that you get from Matlab's `A*B`. You can use loops.
7. Show three ways to compute an inner product between two vectors.
8. Compute and print the L1-norm and L2-norm for each of the following two vectors: x1 = [0.5 0 1.5] and x2 = [1 1 0]. Verify your answer against the Matlab function `norm`.
9. Use Matlab to solve the system of linear equations: 2x + y + 3z = 1; 2x + 6y + 8z = 3; 6x + 8y + 18z = 5.

Functions:

10. Write a function `function [B] = normalize_rows(A)` which uses a single command (one line and no loops) to make the sum in each row of the matrix 1. You may need to use `repmat`, depending on your Matlab version. Note that the sum of the entries in *each* row should be 1, in the

matrix output by your function.

11. Create a function `function [val] = fib(n)` that returns the n-th number (n >= 1) in the Fibonacci sequence 1, 1, 2, 3, 5, 8, 13...

Machine learning setup:

12. We have the following training set: a red circle, a blue triangle, a blue circle, a green triangle, and a red square. Represent each of these samples as a feature vector, using the same representation scheme/strategy. Write a comment to explain the meaning of each dimension in your representation, i.e. what feature it corresponds to, and what the possible values of that feature are.

Part II: Short answers (4 pts)

In a file `answers.txt`, propose two new problems that can be solved with machine learning (ones that we have not discussed in class), and describe how you would go about solving each. For each problem, discuss:

- What features (x) would you use?
- What would the labels (y) be?
- How would you collect data?
- Why might the problem turn out to be challenging?

Submission: Please include the following files:

- A Matlab script `hw1.m`
- Matlab functions (in corresponding files with the same names): `normalize_rows` and `fib`
- The short answers file `answers.txt`