Foundations

Stanford CS221 Spring 2019-2020

Owner CA: Prerna Dhareshwar

Version: 1

General Instructions

This (and every) assignment has a written part and a programming part.

The full assignment with our supporting code and scripts can be downloaded as foundations.zip.

a. This icon means a written answer is expected in `foundations.pdf`.

b. This icon means you should write code in `submission.py`.

All written answers must be **in order** and **clearly and correctly labeled** to receive credit.

You should modify the code in `submission.py` between

```
# BEGIN_YOUR_CODE
```

and

```
# END_YOUR_CODE
```

but you can add other helper functions outside this block if you want. Do not make changes to files other than `submission.py`.

Your code will be evaluated on two types of test cases, **basic** and **hidden**, which you can see in `grader.py`. Basic tests, which are fully provided to you, do not stress your code with large inputs or tricky corner cases. Hidden tests are more complex and do stress your code. The inputs of hidden tests are provided in `grader.py`, but the correct outputs are not. To run the tests, you will need to have `graderUtil.py` in the same directory as your code and `grader.py`. Then, you can run all the tests by typing

```
python grader.py
```

This will tell you only whether you passed the basic tests. On the hidden tests, the script will alert you if your code takes too long or crashes, but does not say whether you got the correct output. You can also run a single test (e.g., `3a-0-basic`) by typing

```
python grader.py 3a-0-basic
```

We strongly encourage you to read and understand the test cases, create your own test cases, and not just blindly run `grader.py`.

Welcome to your first CS221 assignment! The goal of this assignment is to sharpen your math and programming skills needed for this class. If you meet the prerequisites, you should find these problems relatively innocuous. Some of these problems will occur again as subproblems of later homeworks, so make sure you know how to do them.

Before you get started, please read the Assignments section on the course website thoroughly.

Problem 1: Optimization and probability

In this class, we will cast a lot of AI problems as optimization problems, that is, finding the best solution in a rigorous mathematical sense. At the same time, we must be adroit at coping with uncertainty in the world, and for that, we appeal to tools from probability. This problem will be covering topics of differentiation and basic probability/expectation, which will appear again throughout the course, so it is recommended to brush up on this material using resources from other courses or on the internet.

a. [2 points] Let $x_1, \ldots, x_n$ be real numbers representing positions on a number line. Let $w_1, \ldots, w_n$ be positive real numbers representing the importance of each of these positions. Consider the quadratic function: $f(\theta) = \frac{1}{2} \sum_{i=1}^{n} w_i(\theta - x_i)^2$. What value of θ minimizes $f(\theta)$? Show that the optimum you find is indeed a minimum. What problematic issues could arise if some of the w_i's are negative?
Note: You can think about this problem as trying to find the point θ that's not too far away from the x_i's. Over time, hopefully you'll appreciate how nice quadratic functions are to minimize.

b. [3 points] In this class, there will be a lot of sums and maxes. Let's see what happens if we switch the order. Let $f(\mathbf{x}) = \sum_{i=1}^{d} \max_{s \in \{1,-1\}} s x_i$ and $g(\mathbf{x}) = \max_{s \in \{1,-1\}} \sum_{i=1}^{d} s x_i$, where $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ is a real vector. Does $f(\mathbf{x}) \le g(\mathbf{x})$, $f(\mathbf{x}) = g(\mathbf{x})$, or $f(\mathbf{x}) \ge g(\mathbf{x})$ hold for all $\mathbf{x}$? Prove it.
Hint: You may find it helpful to refactor the expressions so that they are maximizing the same quantity over different sized sets.

c. [3 points] Suppose you repeatedly roll a fair six-sided die until you roll a 1 (and then you stop). Every time you roll a 2, you lose a points, and every time you roll a 6, you win b points. You do not win or lose any points if you roll a 3, 4, or a 5. What is the expected number of points (as a function of a and b) you will have when you stop?
Hint: it is recommended to think of defining a recurrence.

d. [3 points] Suppose the probability of a coin turning up heads is $0 < p < 1$, and that we flip it 7 times and get $\{\mathrm{H}, \mathrm{H}, \mathrm{T}, \mathrm{H}, \mathrm{T}, \mathrm{T}, \mathrm{H}\}$. We know the probability (likelihood) of obtaining this sequence is $L(p) = pp(1-p)p(1-p)(1-p)p = p^4(1-p)^3$. What value of p maximizes $L(p)$? What is an intuitive interpretation of this value of p?
Hint: Consider taking the derivative of $\log L(p)$. You can also directly take the derivative of $L(p)$, but it is cleaner and more natural to differentiate $\log L(p)$. You can verify for yourself that the value of p which maximizes $\log L(p)$ must also maximize $L(p)$ (you are not required to prove this in your solution).

e. [4 points] Let's practice taking gradients, which is a key operation for being able to optimize continuous functions. For $\mathbf{w} \in \mathbb{R}^d$ (represented as a column vector) and constants $\mathbf{a}_i, \mathbf{b}_j \in \mathbb{R}^d$ (also represented as column vectors) and $\lambda \in \mathbb{R}$, define the scalar-valued function

$$f(\mathbf{w}) = \sum_{i=1}^{n} \sum_{j=1}^{n} (\mathbf{a}_i^\top \mathbf{w} - \mathbf{b}_j^\top \mathbf{w})^2 + \lambda \|\mathbf{w}\|_2^2,$$

where the vector is $\mathbf{w} = (w_1, \ldots, w_d)^\top$ and $\|\mathbf{w}\|_2 = \sqrt{\sum_{k=1}^{d} w_k^2}$ is known as the L_2 norm. Compute the gradient $\nabla f(\mathbf{w})$.
Recall: the gradient is a d-dimensional vector of the partial derivatives with respect to each w_i:

$$\nabla f(\mathbf{w}) = \left(\frac{\partial f(\mathbf{w})}{\partial w_1}, \ldots \frac{\partial f(\mathbf{w})}{\partial w_d} \right)^\top .$$

If you're not comfortable with vector calculus, first warm up by working out this problem using scalars in place of vectors and derivatives in place of gradients. Not everything for scalars goes through for vectors, but the two should at least be consistent with each other (when $d = 1$). Do not write out summation over dimensions, because that gets tedious.

Problem 2: Complexity

When designing algorithms, it's useful to be able to do quick back of the envelope calculations to see how much time or space an algorithm needs. Hopefully, you'll start to get more intuition for this by being exposed to different types of problems.

a. [2 points] Suppose we have an image of a human face consisting of $n \times n$ pixels. In our simplified setting, a face consists of two eyes, two ears, one nose, and one mouth, each represented as an arbitrary axis-aligned rectangle (i.e. the axes of the rectangle are aligned with the axes of the image). As we'd like to handle Picasso portraits too, there are no constraints on the location or size of the rectangles. For example, it is possible for all four corners of a single rectangle to be the same point (which is a rectangle of size 0) or for all 6 rectangles to be on top of each other. How many possible faces (choice of its 6 component rectangles) are there? In general, we only care about asymptotic complexity, so give your answer in the form of $O(n^c)$ or $O(c^n)$ for some integer c.

b. [3 points] Suppose we have an $n \times n$ grid. We start in the upper-left corner (position $(1, 1)$), and we would like to reach the lower-right corner (position (n, n)) by taking single steps down or to the right. Let a function $c(i, j)$ represent the cost of touching position (i, j), and assume it takes constant time to compute. Note that $c(i, j)$ can be negative. Give an algorithm for computing the minimum cost in the most efficient way. What is the runtime (just give the big-O)?

c. [3 points] Suppose we have a staircase with n steps (we start on the ground, so we need n total steps to reach the top). We can take as many steps forward at a time, but we will never step backwards. How many ways are there to reach the top? Give your answer as a function of n.
For example: if $n = 3$, then the answer is 4. The four options are the following: (1) take one step, take one step, take one step (2) take two steps, take one step (3) take one step, take two steps (4) take three steps.

d. [4 points] Consider the scalar-valued function $f(\mathbf{w})$ from Problem 1e. Devise a strategy that first does preprocessing in $O(nd^2)$ time, and then for any given vector $\mathbf{w}$, takes $O(d^2)$ time instead to compute $f(\mathbf{w})$.
Note: Preprocessing refers to any computations that can be done independent of $\mathbf{w}$.

Hint: Refactor the algebraic expression; this is a classic trick used in machine learning. Again, you may find it helpful to work out the scalar case first.

Problem 3: Programming

In this problem, you will implement a bunch of short functions. The main purpose of this exercise is to familiarize yourself with Python, but as a bonus, the functions that you will implement will come in handy in subsequent homeworks.

Do not import any outside libraries (e.g. numpy). Only standard python libraries and/or the libraries imported in the starter code are allowed.

If you're new to Python, the following provide pointers to various tutorials and examples for the language:

- Python for Programmers
- Example programs of increasing complexity

a. [3 points] Implement `findAlphabeticallyLastWord` in `submission.py`.

b. [3 points] Implement `euclideanDistance` in `submission.py`.

c. [6 points] Implement `mutateSentences` in `submission.py`.

d. [4 points] Implement `sparseVectorDotProduct` in `submission.py`.

e. [4 points] Implement `incrementSparseVector` in `submission.py`.

f. [4 points] Implement `findSingletonWords` in `submission.py`.

g. [5 points] Implement `computeLongestPalindromeLength` in `submission.py`.

Problem 4: Societal impact

AI can no longer be viewed as a neutral technology, for its impact on society is increasing [1, 2]. While most of this course is the technology, we would like you to pause for a moment to reflect on how this technology can influence people's lives.

a. [1 point] Describe one positive use of AI to improve society that you are most excited about. What would be a concrete grand challenge or milestone in this area?

b. [1 point] Describe one negative use or risk of AI that you are most worried about. How would you go about preventing this?

Problem 5: Background Survey

As a big class, CS221 has a wide range of people coming from different backgrounds. Please fill the form linked here so we can get a better sense of who you are and what you know so far. This will be graded for completion, so your answers will not affect your grade whatsoever.

Submission

Submission is done on Gradescope.

Written: When submitting the written parts, make sure to select **all** the pages that contain part of your answer for that problem, or else you will not get credit. To double check after submission, you can click on each problem link on the right side and it should show the pages that are selected for that problem. In order to receive the typeset bonus, you must select the first page of your submission.

Programming: After you submit, the autograder will take a few minutes to run. Check back after it runs to make sure that your submission succeeded. If your autograder crashes, you will receive a 0 on the programming part of the assignment.

More details can be found in the Submission section on the course website.