DSC 40A - Homework 2

Due: Monday, October 11, 2021 at 11:59pm PT

Write your solutions to the following problems by either typing them up or handwriting them on another piece of paper. Homeworks are due to Gradescope by 11:59pm PT on the due date. You can use a slip day to extend the deadline by 24 hours. Make sure to correctly assign pages to Gradescope when submitting.

Homework will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should **always explain and justify** your conclusions, using sound reasoning. Your goal should be to convince the reader of your assertions. If a question does not require explanation, it will be explicitly stated.

Homeworks should be written up and turned in by each student individually. You may talk to other students in the class about the problems and discuss solution strategies, but you should not share any written communication and you should not check answers with classmates. You can tell someone how to do a homework problem, but you cannot show them how to do it.

For each problem you submit, you should **cite your sources** by including a list of names of other students with whom you discussed the problem. Instructors do not need to be cited.

This homework will be graded out of 48 points. The point value of each problem or sub-problem is indicated by the number of avocados shown.

Note: Problems 4 and 5 refer to a supplemental Jupyter Notebook, which can be found **at this link**.

Problem 1. Adding a Data Point

🥑🥑🥑🥑🥑 Suppose we have a dataset $y_1 \leq y_2 \leq \cdots \leq y_n$ and want to minimize absolute loss on it for the prediction h. As we've seen before, the corresponding empirical risk is mean absolute error,

$$R_{\text{abs}}(h) = \frac{1}{n} \sum_{i=1}^{n} |y_i - h|$$

Suppose that $R_{\text{abs}}(\alpha) = R_{\text{abs}}(\alpha+3) = M$, where M is the minimum value of $R_{\text{abs}}(h)$ and α is some constant. Suppose we add to the dataset a new data point y_{n+1} whose value is $\alpha + 1$. For this new larger dataset, what is the minimum of $R_{\text{abs}}(h)$ and at what value of h is it achieved? Your answers to both parts should only involve the variables n, M, α, and constants.

Problem 2. Quadratus Lumborum

In class, we used calculus to prove that the mean minimizes mean squared error, $R_{sq}(h) = \frac{1}{n}\sum_{i=1}^{n}(h - y_i)^2$. In this problem, we'll use another technique to prove this same fact.

a) 🥑🥑🥑🥑 Recall, the mean squared error $R_{sq}(h)$ is a quadratic function of h. Any quadratic function of h can be written in the form $Ah^2 + Bh + C$, where A, B, and C are constants that do not depend on h.

Determine the values of A, B, and C such that $R_{sq}(h) = Ah^2 + Bh + C$. Your answers for A, B, and C should only be in terms of the variables $n, y_1, y_2, ..., y_n$, and any constants.

Hint: Start by expanding $R_{sq}(h)$.

b) 🥑🥑 Let's simplify your expressions for A, B, and C.

The variance of our dataset is defined as $\sigma_y^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$, where $\bar{y}$ is the mean of our dataset (i.e. $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. An alternative way of writing variance is as follows:

$$\sigma_y^2 = (\frac{1}{n}\sum_{i=1}^{n} y_i^2) - \bar{y}^2$$

Using this fact, simplify your expressions for A, B, and C so that they only contain the variables n, σ_y, $\bar{y}$, and constants.

c) 🥑 In earlier math classes, you may have learned that "completing the square" involves rewriting a quadratic of the form

$$Ah^2 + Bh + C$$

in the form

$$A(h + \frac{B}{2A})^2 + C - \frac{B^2}{4A}$$

Such a quadratic is minimized when $h = -\frac{B}{2A}$, and its minimum value is $C - \frac{B^2}{4A}$.

Evaluate the above expressions ($-\frac{B}{2A}$ and $C - \frac{B^2}{4A}$). Conclude that the mean minimizes mean squared error and that the variance is the minimum possible value of mean squared error.

Problem 3. What Do You Mean?

As discussed in class (and in Problem 2), if we choose squared loss to be our loss function, i.e. $L_{sq}(h, y) = (y - h)^2$, the prediction h^* that minimizes empirical risk is the mean, i.e. $h^* = \text{Mean}(y_1, y_2, ..., y_n)$.

In this problem, we will look at a new loss function, the "relative squared loss function" $L_{rsq}(h, y)$:

$$L_{rsq}(h, y) = \frac{(y - h)^2}{y}$$

Throughout this problem, assume that each of $y_1, y_2, ..., y_n$ is positive.

a) 🥑🥑 Determine $\frac{\partial}{\partial h} L_{rsq}$, that is, the partial derivative of the relative squared loss function with respect to h.

b) 🥑🥑🥑🥑 What value of h minimizes empirical risk for the relative squared loss function, i.e. $R_{rsq}(h) = \frac{1}{n}\sum_{i=1}^{n}\frac{(y_i - h)^2}{y_i}$? Your answer should only be in terms of the variables $n, y_1, y_2, ..., y_n$, and any constants.

Hint: You will need to use your answer from the previous part.

c) 🥑🥑🥑 Let $H(y_1, y_2, ..., y_n)$ be your result from the previous part. (That is, for a particular dataset $y_1, y_2, ..., y_n$, $H(y_1, y_2, ..., y_n)$ is the value of h that minimizes empirical risk for relative squared loss on that dataset.)

What is the value of $\lim_{y_4 \to \infty} H(1, 2, 4, y_4)$? Your answer should involve the function H and/or one or more constants.

Hint: To notice the pattern, evaluate $H(1, 2, 4, 100)$, $H(1, 2, 4, 10000)$, and $H(1, 2, 4, 1000000)$.

d) 🥑🥑 What is the value of $\lim_{y_4 \to 0} H(1, 2, 4, y_4)$? Again, your answer should involve the function H and/or one or more constants.

e) 🥑🥑 Based on the results of part c) and part d), when is the prediction $H(y_1, y_2, ..., y_n)$ robust to outliers? When is it not robust to outliers?

Problem 4. Garage Sale: Everything d Dollars!, Continued

For a fixed value x_i, consider the garage sale loss function,

$$L(d) = L(d; x_i) = \begin{cases} x_i - d, \text{if } d \leq x_i \\ x_i, \text{else} \end{cases} . \tag{1}$$

Recall that if x_i represents the dollar value of an item at a garage sale, $L(d; x_i)$ represents the amount of potential earnings lost if we price the item at d dollars. As before, we assume that for each item, if the price of the item is more than its value, then the item will not sell, and otherwise, it will sell for the asking price.

For a set of n items with dollar values $x_1 \leq x_2 \leq \cdots \leq x_n$, define the total loss as

$$T(d) = \sum_{i=1}^{n} L(d; x_i).$$

Our goal is to find the asking price d that minimizes this total loss $T(d)$. (Note that total loss is similar to empirical risk, but does not have a factor of $\frac{1}{n}$ in front.)

a) 🥑🥑 Can we use gradient descent to minimize $T(d)$? Explain why or why not.

b) 🥑🥑🥑🥑 Notice that $L(d; x_i)$ is made up of linear functions, which means $T(d)$ is also made up of linear functions. Suppose $n = 3$ and $0 < x_1 < x_2 < x_3$. For d in each of the intervals below, what is the slope of the graph of $T(d)$?

1. $d < x_1$
2. $x_1 < d < x_2$
3. $x_2 < d < x_3$
4. $d > x_3$

Hint: Refer back to Lecture 2, where we figured out the slope of $R_{abs}(h)$, and apply a similar strategy.

c) 🥑🥑 Give a formula for the slope of $T(d)$ at any value of d (besides $d = x_1, d = x_2, \ldots, d = x_n$, where the slope is not defined). This formula should work for any number of data points n and any set of values $x_1 \leq x_2 \leq \cdots \leq x_n$, even if some of them are the same.

Hint: Refer back to Lecture 2, where we figured out the slope of $R_{abs}(h)$, and apply a similar strategy.

d) 🥑🥑 Use your formula to explain why $T(d)$ must be minimized at one of $\{x_1, x_2, \ldots, x_n\}$.

e) 🥑🥑🥑🥑 In the supplemental Jupyter Notebook, which can be found at this link, complete the implementation of `maximize_money`, which takes in a list of x_is and returns the optimal selling price d^*. You can assume that the input list is sorted.

Turn in an explanation of the strategy you used to solve this problem, a screenshot of your code for the function, and the output of your function on each of the following inputs:

1. $[1, 2, 4, 5, 11]$
2. $[4, 10, 12, 16, 20, 21, 22, 30, 32]$

3. $[3, 3, 3, 5, 9, 10, 12, 15, 15]$

Problem 5. Exploring Gradient Descent

In this class, we'll primarily use gradient descent to minimize empirical risk. We'll see a lot of this in lecture and in upcoming assignments. However, gradient descent can also be used to minimize functions that have nothing to do with empirical risk.

Suppose want to minimize the polynomial function $g(u)$ using gradient descent, where

$$g(u) = u^4 - 24u^3 + 154u^2 - 207u - 12$$

a) 🥑 What is $g'(u)$, the derivative of g with respect to u?

b) 🥑 Gradient descent is guaranteed to find the global minimum of a function if the function is *convex* (also called concave up) and *differentiable*, if given an appropriate choice of step size. Remember from calculus that a function $f : \mathbb{R} \to \mathbb{R}$ is convex (concave up) if

$$f''(x) = \frac{\mathrm{d}^2 f}{\mathrm{d}x^2} \geq 0, \text{for all } x.$$

Prove that $g(u)$ is **not** convex.

Hint: Use the definition of convexity above and show that there is at least one value of x for which it does not hold.

c) 🥑🥑🥑 Since g is not convex, gradient descent is not guaranteed to converge to a global minimum. However, given an appropriate choice of step size, gradient descent will converge to a local minimum, which may or may not be a global minimum.

For non-convex functions, whether or not gradient descent converges to a global minimum depends on where we initialize gradient descent, i.e. where our "first guess" for the minimizing value is. Let's experiment with this idea.

In the supplemental Jupyter Notebook, which can be found at this link, complete the following tasks:

- Complete the implementation of `g(u)`.
- Complete the implementation of `dg(u)` (i.e. the derivative of `g(u)`) using your answer from part a).
- Run the cell provided to create a plot of `g(u)`. You don't need to do anything to create this plot.
- Based on the above plot, answer the following questions in your write-up:
  - How many local minimums does $g(u)$ have?
  - What are the values of u that correspond to local minimums? (Come up with approximate values based on the plot, your answers don't need to be exact.)
  - Which value of u from above corresponds to a global minimum, if any?

d) 🥑🥑🥑🥑 At the bottom of the supplementary notebook, linked above, you will see a call to the function `visualize_gradient_descent`. This function animates the execution of gradient descent on a single-variable function (in our case, `g`). It allows us to specify different initial guesses for the best u (through the argument `h_0`) and different step sizes/learning rates (through the argument `alpha`).

Your task in this subpart is to call `visualize_gradient_descent` with different pairs of initial guesses and step sizes and observe the resulting behavior. Start with an initial guess of 5 and step size of 0.01.

In your write-up, all you are required to do is include answers to the following questions:

- Suppose we choose -2 as an initial guess. Find two different step sizes `alpha` such that gradient descent converges at different local minimums for both step sizes. In your write-up, note the two step sizes and which local minimums they each converged to, and give a brief explanation as to why you think this happened.
- If you set `alpha` to 0.1, you will get an error that says `OverflowError: (34, 'Result too large')`. Why do you think this happens, given that 0.1 is a relatively small step size? (Refer to the scale of the function $g(u)$ in your answer.)