

---

## DSC 40A - Homework 1

Due: Tuesday, July 11 at 11:59pm

---

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 on the due date. You can use a slip day to extend the deadline by 24 hours.

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 50 points. The point value of each problem or sub-problem is indicated by the number of avocados shown.


### Problem 1. Means

Which of the following statements *must* be true? Remember to justify all answers.

- a)  At least half of the numbers in a data set must be larger than the median.
- b)  At least a few numbers in a dataset must be greater than the mean.
- c)  Exactly half of the numbers in a data set must be smaller than the mean.
- d)  Not all of the numbers in the data set can be smaller than the mean.

### Problem 2. Linear Functions

Consider the linear function  $f(x) = 3x + 7$ .

- a)  If  $a \leq b$ , show that  $f(a) \leq f(b)$ .

### Problem 3. Classification and Binary Cross Entropy

Binary Cross-Entropy is a type of loss function that is used specifically for binary classification problems. A loss function's job is to quantify how well your model is performing by comparing its predictions with the actual outcomes.

You're familiar with Mean Squared Error (MSE) and Mean Absolute Error (MAE), which are commonly used in regression tasks. These loss functions work well when the task is to predict a continuous value, such as predicting house prices or predicting a person's weight. In these cases, we're interested in how far off our predictions are from the actual values, and MSE or MAE provide a good measure of this.

However, in binary classification, our task is to predict whether an instance belongs to one of two classes. For example, we might want to predict whether an email is spam or not spam. In these cases, our model outputs probabilities that an instance belongs to one class or the other.

This is where Binary Cross-Entropy comes in. Instead of measuring the difference between the predicted and actual values, Binary Cross-Entropy measures the 'distance' between the probability distribution output by our model, and the actual distribution of the outcomes.

The actual distribution in this case is straightforward: for each instance, it's 1.0 for the correct class and 0.0 for the other class. The model's predicted distribution is whatever floating point values between 0 and 1 it outputs.

Binary Cross-Entropy loss will be low if the model's predictions are close to the actual outcomes (i.e., it assigns high probability to the correct class), and high if the model's predictions are far from the actual outcomes (i.e., it assigns low probability to the correct class).

Assuming a fixed prediction  $h$  as we have been using for the previous examples, the formula for binary cross entropy is as follows:

$$\text{Binary Cross-Entropy} = -\frac{1}{N} \sum_{i=1}^N [y_i \cdot \log(h) + (1 - y_i) \cdot \log(1 - h)]$$

Note that this loss is undefined at  $h=0$  or  $h=1$ , so we will add a constraint for the purposes of this assignment that the minimum value for  $h$  is 0.01 and the maximum value is 0.99.

- a) 🥑🥑🥑🥑 Let  $Y = [1, 0, 0, 0]$  (the true labels) and  $h = 0.99$ . Calculate binary cross entropy.
- b) 🥑🥑 Continuing with  $Y = [1, 0, 0, 0]$  (the true labels), now let  $h = 0.01$ . Calculate binary cross entropy.
- c) 🥑🥑🥑🥑 Describe the intuition of BCE in relation to  $h = .99$  vs  $h = .01$  when we have  $Y = [1, 0, 0, 0]$

#### Problem 4. Linear Transformations

Suppose we are given a data set  $\{d_1, d_2, \dots, d_n\}$  and know its mean, variance, and standard deviation to be  $mean_d$ ,  $var_d$ , and  $std_d$ . Consider another data set  $\{t_1, t_2, \dots, t_n\}$ , where  $t_i$  is a linear transformation of  $d_i$ :

$$t_i = f(d_i) = a \cdot d_i + b$$

for each  $i = 1, 2, \dots, n$ . Here,  $a$  and  $b$  are arbitrary constants. Let  $mean_t$ ,  $var_t$ , and  $std_t$  be the mean, variance, and standard deviation of the transformed data.

- a) 🥑🥑 Express  $mean_t$  in terms of  $mean_d$ ,  $a$ , and  $b$  (you may not need all of these).
- b) 🥑🥑 Express  $var_t$  in terms of  $var_d$ ,  $a$ , and  $b$  (you may not need all of these).
- c) 🥑🥑 Express  $std_t$  in terms of  $std_d$ ,  $a$ , and  $b$  (you may not need all of these).

### Problem 5. Spread

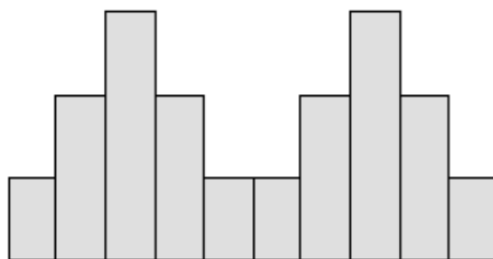
In class, we defined the *mean absolute deviation from the median* as a measure of the spread of a data set. This measure takes the absolute deviations, or differences, of each value in the data set from the median, and computes the mean of these absolute deviations. We can think of this one measure of spread as a member of a family of analogously defined measures of spread:

- mean absolute deviation from the median
- median absolute deviation from the median
- mean absolute deviation from the mean
- median absolute deviation from the mean

While all four of these measures capture the notion of spread, they do so in different ways, and so they may have different values for the same data set.

- a) 🥑🥑🥑🥑 For the data set whose histogram is shown below, draw a histogram showing the rough shape of the distribution of the absolute deviations from the mean. Which of these two measures is greater, or are they about the same?

- mean absolute deviation from the mean
- median absolute deviation from the mean

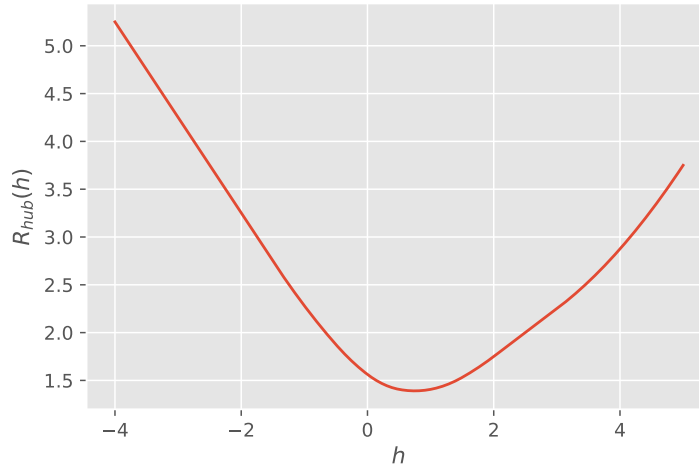


### Problem 6. Huber Loss

The *Huber loss* is a mixture between the square loss and the absolute loss. It is defined piecewise as follows:

$$L_{\text{hub}}(h, y) = \begin{cases} |h - y|, & |h - y| > 1 \\ \frac{1}{2}(h - y)^2 + \frac{1}{2}, & |h - y| \leq 1 \end{cases}$$

- a) 🥑🥑 Fix an arbitrary value of  $y$ . Draw the graph of  $L_{\text{hub}}(h, y)$  as a function of  $h$ . You should notice that  $L_{\text{hub}}(h, y)$  is minimized at  $y$ .
- b) 🥑🥑🥑 What is the derivative of  $L_{\text{hub}}$  with respect to  $h$ ? Your answer should also be a piecewise function.
- c) 🥑🥑🥑🥑 Suppose our data set is  $\{-\frac{1}{2}, \frac{1}{2}, 1, 4\}$ . The plot of the empirical risk,  $R_{\text{hub}}(h) = \frac{1}{n} \sum_{i=1}^n L_{\text{hub}}(h, y)$  is shown below:



It is not possible to directly solve for the value of  $h$  which minimizes this function. Instead, run gradient descent by hand using an initial prediction of  $h_0 = 5$  and a step size of  $\alpha = 2$ . Run the algorithm until it converges (it shouldn't take too many iterations). Please show your calculations, and to help the graders track your progress, include a boxed summary with the value of  $h$  at each iteration, such as below:

$h_0 = 5$
$h_1 = \dots$
$h_2 = \dots$
$\vdots$

**Problem 7. Gradient Descent for Linear Regression**

Soon in the course we are going to learn about linear regression, which is commonly known as finding the “line of best fit”. Interestingly enough, the metric used to define “best fit” is a function you are quite familiar with already, the mean squared error:

$$R_{sq}(h) = \frac{1}{n} \sum_{i=1}^n (y_i - h)^2.$$

So far in this class, our prediction  $h$  has been a single real number. When performing linear regression, however, our prediction is allowed to vary with each  $x_i$ , according to some linear function  $f(x_i) = mx_i + b$ , where  $m$  and  $b$  are real numbers. That means, when talking about how well a linear function “fits” the data, we are measuring the fit by the mean squared error:

$$R_{sq}(f) = R_{sq}(m, b) = \frac{1}{n} \sum_{i=1}^n (y_i - (mx_i + b))^2$$

and linear regression is just finding the values of  $m$  and  $b$  that minimize this mean squared error.

While this may look intimidating, we actually have already learned an excellent tool that will be able to help us solve this problem: gradient descent.

Gradient descent is guaranteed to find the global minimum of a function if the function is *convex* (also called concave up) and *differentiable*. In order to verify that gradient descent will indeed find the global minimum for us, we need to prove that the mean squared error is both convex and differentiable. Remember from calculus that a function  $g : \mathbb{R} \rightarrow \mathbb{R}$  is convex (concave up) if

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

- a) 🥑🥑🥑 Use the above definition of convexity to show that

$$R_{sq}(h) = \frac{1}{n} \sum_{i=1}^n (y_i - h)^2$$

is a convex function of  $h$ .

- b) 🥑🥑🥑🥑 In order to run gradient descent on  $R_{sq}(m, b)$ , we first need to find its gradient. Thinking of  $R_{sq}(m, b)$  as a function of two variables, compute the partial derivatives  $\frac{\partial}{\partial m} R_{sq}(m, b)$  and  $\frac{\partial}{\partial b} R_{sq}(m, b)$ .