DSC 40A - Group Work Session 2

1 Empirical Risk Minimization

In class, we've seen how to minimize the empirical risk associated with certain natural loss functions, such as the absolute loss and the squared loss. There are a variety of other possible loss functions we could use instead. This problem explores empirical risk minimization with an alternate choice of loss function.

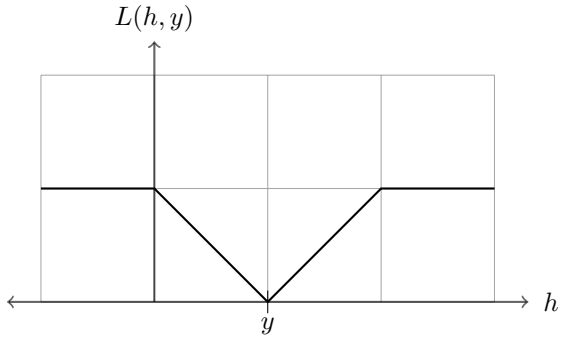
Problem 1.

In this problem, consider the loss function

$$L(h, y) = \begin{cases} 1, & |y - h| > 1 \\ |y - h|, & |y - h| \leq 1 \end{cases}.$$

a) Consider y to be a fixed number. Plot $L(h, y)$ as a function of h.

Solution:



b) Suppose that we have the following data:

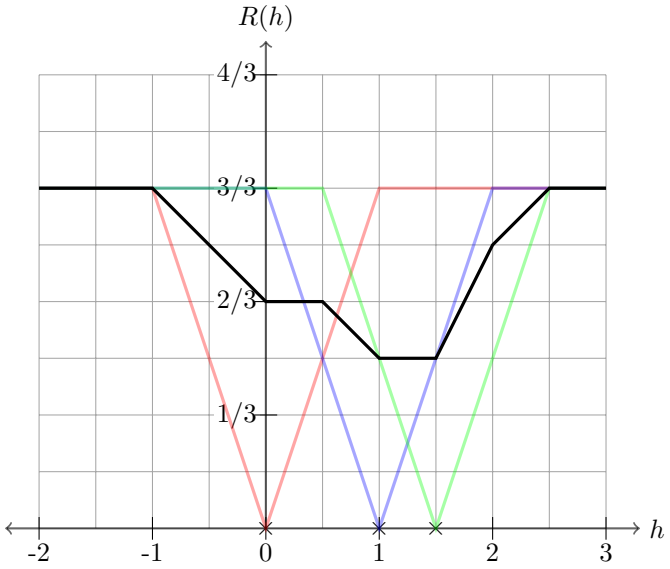
$$y_1 = 0$$
$$y_2 = 1$$
$$y_3 = 1.5$$

Plot the empirical risk

$$R(h) = \frac{1}{n} \sum_{i=1}^{n} L(h, y_i)$$

on the domain $[-2, 3]$. It might help to use the grid on the next page; note that the vertical axis tick marks occur in increments of $1/3$ while the horizontal axis tick marks are in increments of 1.

Hint: $R(h)$ is made up of several line segments. What is the slope of each line segment?



Note: You should be able to do this by hand without using technology. After you've done this, click this link to check your work using Desmos, an online graphing calculator.

Solution: The loss function, $L(h, y)$, is a piecewise linear function, and so the risk, $R(h)$, is a constant $(1/3)$ times sum of piecewise linear functions. Recall that the slope of the sum of piecewise linear functions at a particular point is the sum of the slopes of the pieces at that point.

For instance, consider $h = -1/2$. $R(h)$ is the sum of the red, blue, and green functions plotted above. At this point, the slope of the red function is -1, while the slopes of the other two functions are zero. So the slope of the risk, R, at this point is $\frac{1}{3}(-1 + 0 + 0) = -1/3$.

We can break the function into pieces where the slope does change: these "break points" will occur at the data points, and at points which are 1 to the left or 1 to the right of a data point. We calculate the slope in each section. Starting from the left, we can draw R as a piecewise linear function using these slopes.

c) Suppose that we are interested in finding the typical price of an avocado using this loss function. To do so, we have gathered a data set of n avocado prices, $y_1, \ldots, y_n$, and we found the price h^* which minimized the empirical risk (a.k.a, average loss), $R(h) = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} L(h, y_i)$.

Unfortunately, a flat tax of c dollars has been imposed on avocados since we performed our analysis, increasing every price in our data set by c.

Is it true that $h^* + c$ is a minimizer of R when we use the new prices, $(y_1 + c), (y_2 + c), \ldots, (y_n + c)$? Explain why or why not by explaining how the graph of R changes.

Solution: Yes, it is. The same number of points are greater than distance one away from h^* after the shift, and the points within distance 1 from h^* are the same distance away. Hence the graph of R is simply shifted over by c units, and the minimizers are shifted, too.

d) Given avocado prices $\{1/4, 1/2, 3/4, 7/8, 9/8\}$, find a minimizer of R. Provide some justification for your answer.

Hint: you don't need to plot R or do any calculation to find the answer.

Solution: Because all of the data points are within distance 1 of another, the risk function for $y_1 \leq h \leq y_n$ is simply:

$$R(h) = \frac{1}{5} \sum_{i=1}^{5} |y_i - h|.$$

This is the mean absolute error, and it is minimized by the median: $3/4$.

2 Gradient Descent

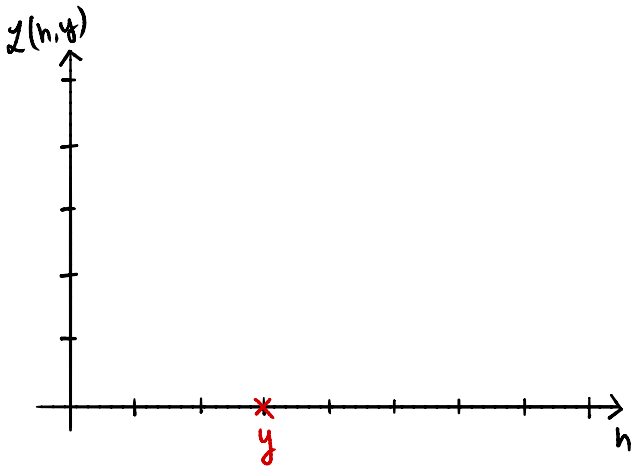
Gradient descent is an algorithm used to minimize differentiable functions. In this class, we will primarily use it to minimize empirical risk, though its use is much more broad. In this problem, we'll explore a new loss function and see how our initial prediction impacts how much our prediction changes with one iteration.

Problem 2.

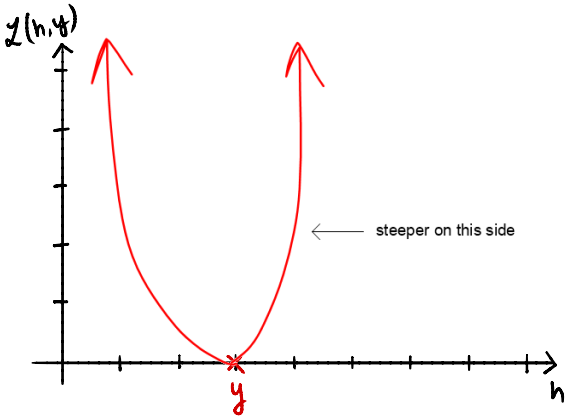
Consider a new loss function,

$$L(h, y) = \begin{cases} (h-y)^2, & h < y \\ (h-y)^3 & h \geq y \end{cases}.$$

a) Fix an arbitrary value of y. On the axes below, draw the graph of $L(h, y)$ as a function of h.



Solution:



b) For any data set $y_1, \ldots, y_n$, the empirical risk, $R(h) = \frac{1}{n}\sum_{i=1}^{n} L(h, y_i)$ will be differentiable and convex (also known as concave up). This means gradient descent is guaranteed to be able to find its minimum value. It might take many iterations or only a few. We won't do the whole algorithm, just one iteration for practice.

Suppose our data set is $\{2, 3, 6, 10\}$. Perform one iteration of gradient descent by hand on the empirical risk function $R(h)$ for this data set, starting with an initial prediction of $h_0 = 3$ and using a step size of $\alpha = \frac{1}{10}$. Calculate h_1, the prediction after the first iteration.

Solution:

First we must calculate the derivative of the risk. Notice that since

$$L(h, y) = \begin{cases} (h-y)^2, & h < y \\ (h-y)^3 & h \geq y \end{cases},$$

then by taking the derivative of each part as a function of h, we have

$$L'(h, y) = \begin{cases} 2(h-y), & h < y \\ 3(h-y)^2 & h \geq y \end{cases}.$$

Using the sum rule for derivatives, this gives:

$$\begin{aligned} R'(h) &= \frac{1}{n} \sum_{i=1}^{n} L'(h, y_i) \\ &= \frac{1}{n} \sum_{i=1}^{n} \begin{cases} 2(h-y), & h < y \\ 3(h-y)^2 & h \geq y \end{cases}. \end{aligned}$$

Now, we are ready to start gradient descent. We start the first iteration with $h_0 = 3$. To apply the gradient descent update rule, we first have to calculate the derivative of $R'(h)$ at $h_0 = 3$:

$$\begin{aligned} R'(3) &= \frac{1}{4} \sum_{i=1}^{4} L'(3, y_i) \\ &= \frac{1}{4} \left[L'(3,2) + L'(3,3) + L'(3,6) + L'(3,10) \right] \end{aligned}$$

We now evaluate each derivative using the piecewise formula for $L'(h, y)$ above.

$$\begin{aligned} &= \frac{1}{4} \left[3(3-2)^2 + 3(3-3)^2 + 2(3-6) + 2(3-10) \right] \\ &= \frac{-17}{4} \end{aligned}$$

Applying the update rule, we find:

$$\begin{aligned} h_1 &= h_0 - \alpha \cdot R'(h_0) \\ &= 3 - \frac{1}{10} \cdot \frac{-17}{4} \\ &= 3 + \frac{17}{40} \end{aligned}$$

c) For the same data set $\{2, 3, 6, 10\}$, suppose instead we did one iteration of gradient descent on $R(h)$ but starting at a different initial prediction, $h_0 = 7$. Using the same step size of $\alpha = \frac{1}{10}$, calculate h_1, the prediction after the first iteration, for this new starting point.

Solution: We start by calculating the derivative of $R'(h)$ at $h_0 = 7$:

$$
\begin{aligned}
R'(7) &= \frac{1}{4}\sum_{i=1}^{4} L'(7, y_i) \\
&= \frac{1}{4}\left[L'(7,2) + L'(7,3) + L'(7,6) + L'(7,10)\right]
\end{aligned}
$$

We now evaluate each derivative using the piecewise formula for $L'(h, y)$ above.

$$
\begin{aligned}
&= \frac{1}{4}\left[3(7-2)^2 + 3(7-3)^2 + 3(7-6)^2 + 2(7-10)\right] \\
&= 30
\end{aligned}
$$

Applying the update rule, we find:

$$
\begin{aligned}
h_1 &= h_0 - \alpha \cdot R'(h_0) \\
&= 7 - \frac{1}{10} \cdot 30 \\
&= 4
\end{aligned}
$$

d) Compare your answers to the previous parts and notice that the prediction moves only a little bit when we start at $h_0 = 3$, but it moves a lot when we start at $h_0 = 7$. Can you explain why that happens by looking at the loss function we're using?

Solution: Notice that the loss function $L(h, y)$ is much steeper to the right of y than to the left of y. This means when our prediction h is greater than a data value, the loss is significantly more than when our prediction is equidistant from the data value in the other direction. The risk, since it comes from averaging loss functions, also behaves like this, and increases much more steeply for larger h values than for smaller h values.

For our data set $\{2, 3, 6, 10\}$, the prediction $h_0 = 3$ is a bit low, but not that much lower than the minimum value of $R(h)$, so the slope $R'(3)$ is not too steep. When we use $h_0 = 7$, the risk is an average of mostly cubic functions since 7 is larger than most of the data points, and so the slope there is very steep. This corrects itself quickly by moving rapidly towards the minimum value. As we can see in the gradient descent update rule, when the derivative is larger, the algorithm takes larger steps.