DSC 140A - Homework 03

Due: Wednesday, January 31

Write your solutions to the following problems by either typing them up or handwriting them on another piece of paper. Unless otherwise noted by the problem's instructions, show your work or provide some justification for your answer. Homeworks are due via Gradescope at 11:59 PM.

Problem 1.

Suppose that $f : \mathbb{R}^d \to \mathbb{R}$ is convex and $g : \mathbb{R} \to \mathbb{R}$ is convex and non-decreasing. That is, if $a > b$, then $g(a) \geq g(b)$.

Show that the composition of these functions, $h(\vec{x}) = g(f(\vec{x}))$, is also convex.

Hint: you'll want to go back to the definition to show this is true. A similar problem was solved in discussion.

Solution: Using the definition of convexity, we have that a function f is convex if every $a, b \in \mathbb{R}$ and $t \in [0, 1]$, the following is true

$$(1-t)f(a) + tf(b) \geq f((1-t)a + tb)$$

So given that $h(\vec{x}) = g(f(\vec{x}))$, to show that h is convex, we have to show that the following is true

$$(1-t)h(\vec{x}_1) + th(\vec{x}_2) \geq h((1-t)\vec{x}_1 + t\vec{x}_2)$$

First using the fact that $h(\vec{x}) = g(f(\vec{x}))$, and that g is convex and non-decreasing, we have that

$$\begin{aligned}(1-t)h(\vec{x}_1) + th(\vec{x}_2) &= (1-t)g(f(\vec{x}_1)) + tg(f(\vec{x}_2)) \\ &\geq g((1-t)f(\vec{x}_1) + tf(\vec{x}_2))\end{aligned}$$

Now we can use the fact that f is convex to show that

$$g((1-t)f(\vec{x}_1) + tf(\vec{x}_2)) \geq g(f((1-t)\vec{x}_1 + t\vec{x}_2))$$

No we can combine the two inequalities to show that

$$\begin{aligned}(1-t)h(\vec{x}_1) + th(\vec{x}_2) &\geq g(f((1-t)\vec{x}_1 + t\vec{x}_2)) \\ &\geq h((1-t)\vec{x}_1 + t\vec{x}_2)\end{aligned}$$

Thus we have shown that h is convex.

Problem 2.

Recall that the *hinge loss* is

$$L_{\text{hinge}}(\vec{w}, \vec{x}, y) = \max\{0, 1 - y\,\vec{w} \cdot \text{Aug}(\vec{x})\}$$

The Soft-SVM problem aims to minimize the regularized empirical risk:

$$R(\vec{w}) = \frac{C}{n}\sum_{i=1}^{n} L_{\text{hinge}}(\vec{w}, \vec{x}^{(i)}, y_i) + \|\vec{w}\|^2$$

Show that $R(\vec{w})$ is a convex function of $\vec{w}$.

Hint: you will probably *not* want to use the formal definition of convexity here. Instead, you'll want to show that R is composed of simpler functions which themselves are convex.

Solution: We can show that $R(\vec{w})$ is convex by showing that the functions that compose it are convex. I.e we need to show that the hinge loss function is convex and that the function $f(\vec{w}) = \|\vec{w}\|^2$ is convex.

Hinge Loss

To show that the Hinge Loss function is convex, we can look at the function more closely,

$$L_{\text{hinge}}(\vec{w}, \vec{x}, y) = \max\{0, 1 - y\,\vec{w} \cdot \text{Aug}(\vec{x})\}$$

We can further decompose the function as we see that the function is the maximum of two functions. Hence if we can show that both of these functions are convex, then we can show that the hinge loss function is convex. The first function is the zero function, which is trivially convex. The second function is $1 - y\,\vec{w} \cdot \text{Aug}(\vec{x})$. Which from lecture we know is an affine function, and hence convex. Thus we have shown that the hinge loss function is convex.

Function $f(\vec{w}) = \|\vec{w}\|^2$

To show that the function $f(\vec{w}) = \|\vec{w}\|^2$ is convex, we can use the Hessian matrix to show that the function is convex. As the function is defined as

$$f(\vec{w}) = \|\vec{w}\|^2 = w_1^2 + w_2^2 + \cdots + w_d^2$$

we have the first order derivatives as

$$\frac{\partial f}{\partial w_i} = 2w_i = \begin{bmatrix} 2w_1 \\ 2w_2 \\ \vdots \\ 2w_d \end{bmatrix}$$

Then for the second order derivatives we have all the second order partials as

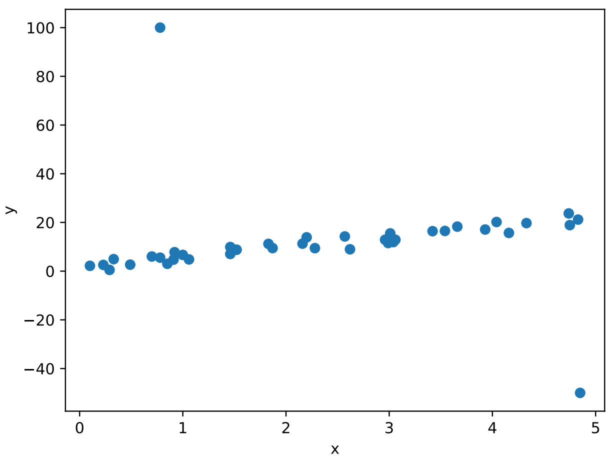
$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial w_1^2} = 2 & 0 & \dots & 0 \\ 0 & \frac{\partial^2 f}{\partial w_2^2} = 2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial^2 f}{\partial w_d^2} = 2 \end{pmatrix}$$

This is because the second order partials of the function are all zero except when $i = j$, in which case the second order partial is 2. Which implies that the function is convex.

As we have shown that both the hinge loss function and the function $f(\vec{w}) = \|\vec{w}\|^2$ are convex, looking at the $R(\vec{w})$ function again, we can see that it is it is composed of two convex functions summations above and a $\frac{C}{n}$, as C and n are just constants, including them, the function is still convex. Now looking at the summations, in lecture we know that the sum of 2 convex functions is convex, which will hold for a summation over n convex functions. Thus we have shown that $R(\vec{w})$ is convex.

Problem 3.

In the last homework, you fit a line to the data set below by minimizing the mean squared error:



Notably, the data contains outliers which may affect our regression.

Remember that you can download the data set at the link below:

`https://f000.backblazeb2.com/file/jeldridge-data/002-regression_outlier/data.csv`

a) Recall the absolute loss:

$$L_{\text{abs}}(H(x), y) = |H(x) - y|.$$

The absolute loss is not as sensitive to outliers as compared to the square loss. However, it is not differentiable, so unlike the mean squared error there is no closed form solution for the minimizer of the risk.

Implement subgradient descent and use it to fit a function of the form $w_0 + w_1 x$ by minimizing the risk with respect to the absolute loss. Report w_0 and w_1, and provide your code.

Hint: the subgradient of the absolute loss was considered in discussion.

Code

```python
import numpy as np
import matplotlib.pyplot as plt


def gradient_descent(gradient, x, y, w0, w1, learning_rate=0.0001,
                     threshold=1e-4, max_iter=1000):
    count = 0
    while True:
        grad = gradient(x, y, w0, w1)
        w0_new = w0 - learning_rate * grad[0]
        w1_new = w1 - learning_rate * grad[1]
        if abs(w1_new - w1) < threshold and abs(w0_new - w0) < threshold:
            break
        w0, w1 = w0_new, w1_new
        if count > max_iter:
            print(count)

            break
        count += 1
    return w1, w0
```

```
def d_abs_loss(x, y, w_0, w_1):
    y_pred = w_0 + w_1 * x
    grad_w_0 = np.sign(y_pred - y).mean()
    grad_w_1 = (np.sign(y_pred - y) * x).mean()
    return grad_w_0, grad_w_1


data = np.loadtxt('data.csv', delimiter=',')
x = data[:, 0]
y = data[:, 1]

# init weights randomly
w_0 = 0
w_1 = 0

new_w_1, new_w_0 = gradient_descent(d_abs_loss, x, y, w_1, w_0, max_iter=1000000)

print("Final weights:")
print("w_0 =", new_w_0)
print("w_1 =", new_w_1)
```

Solution: The code above produces the following output:

$$w_0 = 1.7746000000000444 \; w_1 = 4.015744499999941$$

b) Plot both of the fitted lines on top of a scatter plot of the data; that is, plot the least squares regression line you found in Homework 02 and the line fit by minimizing the risk of the absolute loss.

You should see that the two lines are quite different. Briefly explain why this is the case.

Code

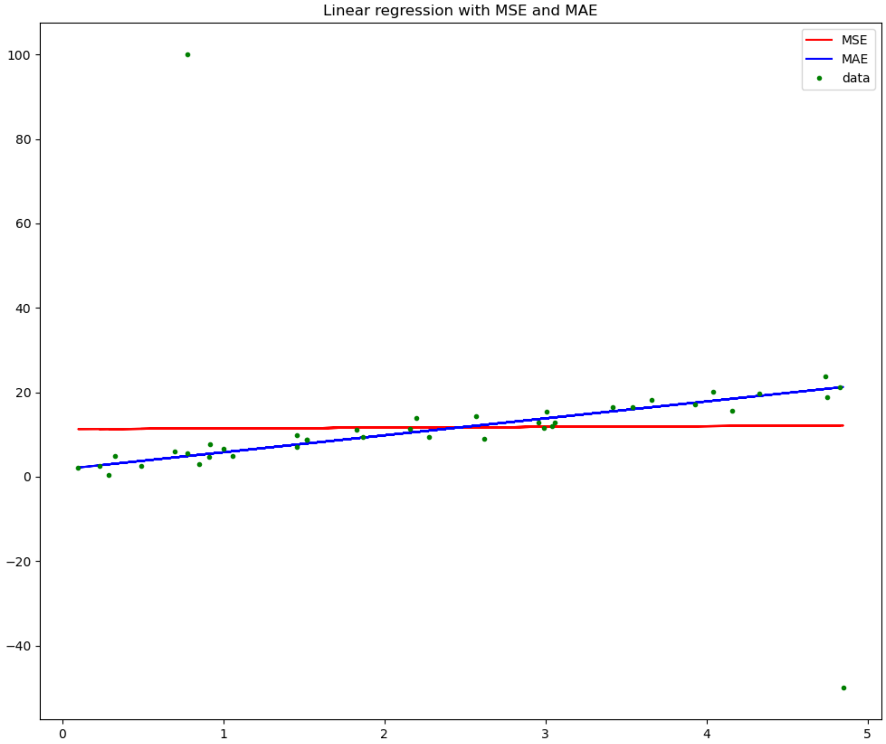
```
# w0 and w1 from Homework 2
w_0_mse = 11.259
w_1_mse = 0.180

w_0_abs = 1.7746000000000444
w_1_abs = 4.015744499999941


y_pred_mse = w_0_mse + w_1_mse * x
y_pred_abs = w_0_abs + w_1_abs * x


plt.figure(figsize=(12, 10))
plt.plot(x, y_pred_mse, color='red', label='MSE')
plt.plot(x, y_pred_abs, color='blue', label='MAE')
plt.plot(x, y, marker='.', linestyle='none', color='green', label='data')
plt.title('Linear regression with MSE and MAE')
plt.legend()
plt.show()
```

Solution: The code above produces the following plot:



There is a a huge difference between the two lines, as the line fit by the MSE loss, which is the red line is very clearly being affected heavily by the outlier, which causes the line to be more straight and miss the true trend of the data. The line fit by the absolute loss, which is the blue line, is not affected by the outlier as much, and hence it was able to capture the trend of the data better, which is evident as the line runs almost straight through the middle of the data points.

This is the case because the MSE loss is more sensitive to outliers than the absolute loss, due to the fact that the MSE loss squares the difference between the predicted value and the actual value, which causes loss values to be much larger than the absolute loss, which in a extremely case like the data given above, the due to the large difference between the points, the MSE loss accumalated extremely large losses from the outliers, which caused the risk minimizer to be heavily affected by the outliers, and hence the line fit by the MSE loss was heavily affected by the outlier. On the other hand, the absolute loss is not as sensitive to outliers, as the absolute loss does not square the difference between the predicted value and the actual value, and hence the loss values are not as large as the MSE loss, and hence the risk minimizer is not as affected by the outliers, and hence the line fit by the absolute loss was not as affected by the outlier.