DSC 140A - Homework 02

Due: Wednesday, January 25

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 p.m.

Problem 1.

Let A be a symmetric $n \times n$ matrix, and let $\vec{x} \in \mathbb{R}^n$. Let $f(\vec{x}) = \vec{x}^T A \vec{x}$. Show that

$$\frac{df}{d\vec{x}} = 2A\vec{x}.$$

Hint: this problem was started in discussion.

Solution: To compute the gradient, we follow the general strategy outlined in lecture of:

1. Expand until we see the components of $\vec{x}$.
2. Compute the gradient vector of partial derivatives: $(\partial f/\partial x_1, \partial f/\partial x_2, \ldots)^T$
3. Try to rewrite the gradient in vector form.

Let the entries of A be

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}.$$

In discussion, it was shown that

$$\vec{x}^T A \vec{x} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j.$$

It may help to visualize all of these terms in a table:

	$j=1$	$j=2$	$\cdots$	$j=n$
$i=1$	$a_{11}x_1^2$	$a_{12}x_1x_2$	$\cdots$	$a_{1n}x_1x_n$
$i=2$	$a_{21}x_2x_1$	$a_{22}x_2^2$	$\cdots$	$a_{2n}x_2x_n$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$i=n$	$a_{n1}x_nx_1$	$a_{n2}x_nx_2$	$\cdots$	$a_{nn}x_n^2$

This table contains all of the terms in the double summation. If we added up all of the terms in this table, we would get exactly $\vec{x}^T A \vec{x}$.

Next, we compute the gradient vector, starting with $\partial f/\partial x_1$. We have:

$$\begin{aligned} \partial f/\partial x_1 &= \frac{\partial}{\partial x_1} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial}{\partial x_1} a_{ij} x_i x_j \end{aligned}$$

If neither i or j are equal to one, the derivative will be zero. Which are not zero, and what are they? This is where the table comes in handy. The partial of each entry with respect to x_1 is:

	$j=1$	$j=2$	$\cdots$	$j=n$
$i=1$	$2a_{11}x_1$	$a_{12}x_2$	$\cdots$	$a_{1n}x_n$
$i=2$	$a_{21}x_2$	0	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$i=n$	$a_{n1}x_n$	0	$\cdots$	0

Since A is symmetric, $a_{12} = a_{21}$, $a_{13} = a_{31}$, etc. Gathering like terms, we find:

$$\begin{aligned} &= 2a_{11}x_1 + 2a_{12}x_2 + \ldots + 2a_{1n}x_n \\ &= 2\sum_{j=1}^{n} a_{1j}x_j \end{aligned}$$

We can guess that, in general, $\partial f/\partial x_i = 2\sum_{j=1}^{n} a_{ij}x_j$. Therefore, the gradient vector is:

$$\frac{df}{d\vec{x}} = \begin{pmatrix} 2\sum_{j=1} a_{1j}x_j \\ 2\sum_{j=1} a_{2j}x_j \\ \vdots \\ 2\sum_{j=1} a_{nj}x_j \end{pmatrix}$$

To show that this equals $2A\vec{x}$, we can work backwards by expanding $2A\vec{x}$. We have:

$$\begin{aligned} 2A\vec{x} &= 2\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \\ &= 2\begin{pmatrix} \sum_{j=1}^{n} a_{1j}x_j \\ \sum_{j=1}^{n} a_{2j}x_j \\ \vdots \\ \sum_{j=1}^{n} a_{nj}x_j \end{pmatrix} \\ &= \begin{pmatrix} 2\sum_{j=1}^{n} a_{1j}x_j \\ 2\sum_{j=1}^{n} a_{2j}x_j \\ \vdots \\ 2\sum_{j=1}^{n} a_{nj}x_j \end{pmatrix} \\ &= \frac{df}{d\vec{x}} \end{aligned}$$

Problem 2.

Consider a linear prediction function H used for binary classification, and assume that when the output of H is positive we predict for class +1, and when it's negative we predict for class -1. This means that the **decision boundary** is where $H(\vec{x}) = 0$.

In lecture, we saw that a linear prediction function has the form:

$$\begin{aligned} H(\vec{x}) &= w_0 + w_1x_1 + \ldots + w_dx_d \\ &= \vec{w} \cdot \text{Aug}(\vec{x}) \end{aligned}$$

where $\vec{w} = (w_0, w_1, \ldots, w_d)^T$. In this problem, it will also be useful to define the vector $\vec{w}' = (w_1, \ldots, w_d)^T$, which is the same as $\vec{w}$ except that it does not include w_0. Note that $\vec{x} \cdot \vec{w}' = w_1 x_1 + \ldots w_d x_d$. With this definition, we can write $H(\vec{x})$ in a slightly different way:

$$H(\vec{x}) = w_0 + \vec{w}' \cdot \vec{x}$$

Remember this formula, as it will be useful several times below!

Over the course of this problem, we'll answer the question: how is the magnitude of $H(\vec{x})$ related to the distance between $\vec{x}$ and the decision boundary?

Note: for this problem it may be useful to review the properties of the dot product and vector algebra. In particular, remember that $\|u\|$ denotes the norm (length) of a vector, and that $\vec{u} \cdot \vec{u} = \|u\|^2$. By dividing a vector by its norm, as in $\vec{u}/\|\vec{u}\|$, we obtain a *unit vector* with unit length – a unit vector is useful for specifying a direction. To find the component of a vector $\vec{u}$ that points in the same direction as another vector $\vec{v}$, we write $\vec{u} \cdot \vec{v}/\|\vec{v}\|$. Also remember that $\vec{u} \cdot \vec{v} = \vec{v} \cdot \vec{u}$, and that $\vec{u} \cdot (\vec{v} + \vec{w}) = \vec{u} \cdot \vec{v} + \vec{u} \cdot \vec{w}$.

a) Show that for any point $\vec{z}$ on the decision boundary, $\vec{w}' \cdot \vec{z} = -w_0$.

Hint: use that fact that $H(\vec{z}) = 0$.

Solution: For any point $\vec{z}$, we have:

$$H(\vec{z}) = \vec{w}' \cdot \vec{z} + w_0$$

Since $\vec{z}$ is on the decision boundary, though, $H(\vec{z}) = 0$, so the above is equal to zero, and $\vec{w}' \cdot \vec{z} = -w_0$.

b) Argue that $\vec{w}'$ is orthogonal to the decision boundary.

Hint: take two arbitrary points $\vec{x}^{(1)}$ and $\vec{x}^{(2)}$ that are assumed to be on the decision boundary. Then, since we know that the boundary is linear (it is a line, plane, etc.), the difference of these vectors, $\vec{\delta} = \vec{x}^{(1)} - \vec{x}^{(2)}$ is parallel to the boundary. To show that $\vec{w}'$ is orthogonal to the boundary, it suffices to show that $\vec{w}' \cdot \vec{\delta} = 0$. Make sure you somehow use the fact that $\vec{x}^{(1)}$ and $\vec{x}^{(2)}$ are on the decision boundary.

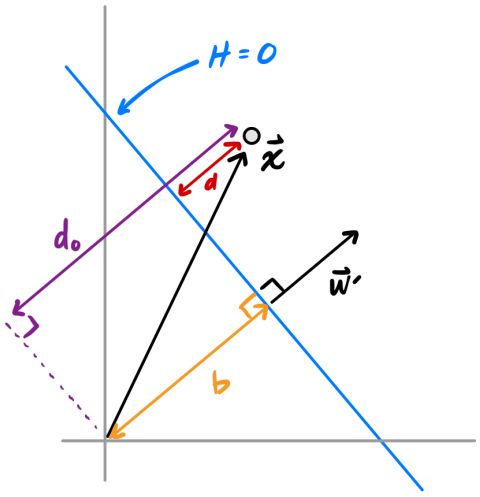
Solution:

$$\begin{aligned} \vec{w}' \cdot \vec{\delta} &= \vec{w}' \cdot (\vec{x}^{(1)} - \vec{x}^{(2)}) \\ &= \vec{w}' \cdot \vec{x}^{(1)} - \vec{w}' \cdot \vec{x}^{(2)} \end{aligned}$$

Because $\vec{x}^{(1)}$ and $\vec{x}^{(2)}$ are on the boundary, we know from the previous part that $\vec{w}' \cdot \vec{x}^{(1)} = -w_0$ and $\vec{w}' \cdot \vec{x}^{(2)} = -w_0$. Therefore:

$$\begin{aligned} &= -w_0 - (-w_0) \\ &= 0 \end{aligned}$$

c) Now that we know that $\vec{w}'$ is orthogonal to the decision boundary, we can draw a better picture of the situation:



The blue line is the decision boundary – it is where $H = 0$. We have drawn an arbitary point $\vec{x}$, along with several distances:

- d: the (signed) distance from the decision boundary to $\vec{x}$
- b: the distance from the the origin to the decision boundary
- d_0: the length of the component of $\vec{x}$ that is orthogonal to the decision boundary.

We're most interested in knowing d. First, though, we need to find b.

Consider the vector $b\vec{w}'/\|\vec{w}'\|$; this is a vector from the origin to the decision boundary that is orthogonal to the boundary and with length b.

Since this vector is on the decision boundary, $H(b\vec{w}'/\|\vec{w}'\|) = 0$. Using this fact, show that $b = -\dfrac{w_0}{\|\vec{w}'\|}$.

Hint: first show that $H(b\vec{w}'/\|\vec{w}'\|) = b\|\vec{w}'\| + w_0$, then set this to zero and solve for b.

Solution: First, we compute:

$$\begin{aligned} H(\vec{z}) &= H\left(\frac{b\vec{w}'}{\|\vec{w}'\|}\right) \\ &= \frac{b\vec{w}'}{\|\vec{w}'\|} \cdot \vec{w}' + w_0 \\ &= \frac{b\|\vec{w}'\|^2}{\|\vec{w}'\|} + w_0 \\ &= b\|\vec{w}'\| + w_0 \end{aligned}$$

Since $\vec{z}$ is on the decision boundary, $H(\vec{z}) = 0$, and so $b\|\vec{w}'\| + w_0 = 0$, meaning that $b = -w_0/\|\vec{w}'\|$.

d) Recall that d_0 is the component of $\vec{x}$ that is orthogonal to the decision boundary; this is simply $\vec{x} \cdot \vec{w}'/\|\vec{w}'\|$. From the picture, $d_0 = d + b$. We know d_0 and b, and can therefore solve for d.

Use this to show that $|d| = |H(\vec{x})|/\|\vec{w}'\|$.

Solution:

$$\begin{aligned} d &= d_0 - b \\ &= \frac{\vec{x} \cdot \vec{w}'}{\|\vec{w}'\|} + \frac{w_0}{\|\vec{w}'\|} \\ &= \frac{\vec{x} \cdot \vec{w}' + w_0}{\|\vec{w}'\|} \\ &= \frac{H(\vec{x})}{\|\vec{w}'\|} \end{aligned}$$

We have shown that the distance between $\vec{x}$ and the decision boundary is proportional to the output of the prediction function, $H(\vec{x})$. This gives us a very useful interpretation of $|H(\vec{x})|$! For example, this means if $H(\vec{x}^{(1)}) > H(\vec{x}^{(2)}) > 0$, then $\vec{x}^{(1)}$ is further from the decision boundary than $\vec{x}^{(2)}$.

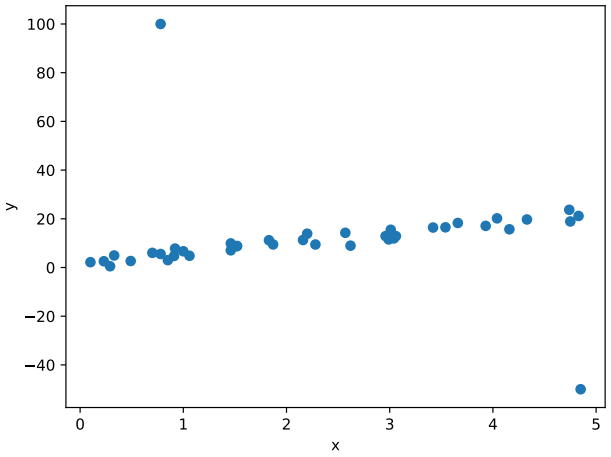
Problem 3.

The file at the link below contains a simple data set suitable for regression.

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

The first column contains the x values (the predictor variable) and the second column contais the y values (the target).

The plot below shows the data:



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

a) Fit a linear function of the form $w_0 + w_1 x$ by minimizing the mean squared error. Report w_0 and w_1, and include your code.

Hint: You may use whatever language or tool you'd like, but if you're using Python, take a look at `np.linalg.lstsq`.

Solution:

```
# load the data
X, y = np.loadtxt('./data.csv', delimiter=',').T

# make an augmented design matrix by adding a column of ones
```

```
# to X
X_aug = np.column_stack((
    np.ones_like(X), X
))

# fit by minimizing squared error
w_sq = np.linalg.lstsq(X_aug, y)[0]
```

We find, approximately: $w_0 \approx 7.9$ and $w_1 \approx 1.74$.

b) 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.

Solution: Recall that a subgradient of the absolute loss is:

$$g(\vec{w}, \vec{x}, y) = \text{sign}(\vec{x} \cdot \vec{w} - y)\vec{x}$$

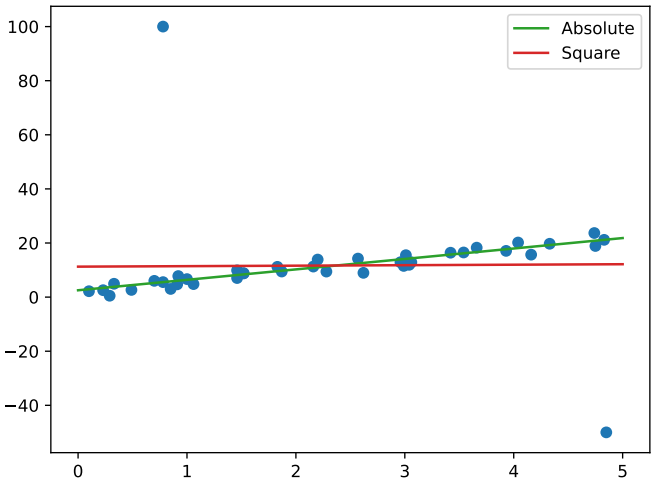
With this information:

```
def subgradient_descent(x, subgradient, learning_rate, tol=1e-5):
    i = 1
    while True:
        g = subgradient(x)
        eta = learning_rate(i)

        if np.linalg.norm(eta * g) < tol:
            break

        x = x - eta * g
        i += 1
    return x

def g_l(w, x, y):
    """Subgradient of the absolute loss w.r.t. w"""
    return np.sign(x @ w - y) * x

def subgradient(w):
    """Subgradient of the risk."""
    return np.mean([g_l(w, x, y) for x, y in zip(X, y)], axis=0)

def learning_rate(n):
    return 1 / np.sqrt(n)
```

Running this, we converge at weights of: $w_0 \approx 2.53$ and $w_1 \approx 3.86$.

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

Solution:



Notice how the slope of the least squares line has been affected by the outliers.