Machine Learning from Data HW6

Shane O'Brien

October 2017

Exercise 3.4

a

$w^{*T}x_n + \epsilon_n$ gives us the y value for a single x, but we want to look at the whole matrix. So we look at $Xw^* + \epsilon$

$$\hat{y} = Hy$$

$$\hat{y} = H(\mathbf{X}w^* + \epsilon)$$

$$\hat{y} = H\mathbf{X}w^* + H\epsilon$$

$$\hat{y} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}w^* + H\epsilon$$

$$\hat{y} = \mathbf{X}Iw^* + H\epsilon$$

$$\hat{y} = \mathbf{X}w^* + H\epsilon$$

b

We know this already:

$$\hat{y} = \mathbf{X}w^* + H\epsilon$$

and

$$y = \mathbf{X}w^* + \epsilon$$

We can use these to do this math and get our answer:

$$\hat{y} - y$$

$$= \mathbf{X}w^* + H\epsilon - (\mathbf{X}w^* + \epsilon)$$

$$= H\epsilon - \epsilon$$

$$= (H - I)\epsilon$$

c

$$\mathbf{E}_{in}(w_{lin}) = \frac{1}{N}\Sigma_{n=1}^{N}(\hat{y} - y)^2$$

$$\mathbf{E}_{in}(w_{lin}) = \frac{1}{N}(\hat{y}-y)^T(\hat{y}-y)$$

$$\mathbf{E}_{in}(w_{lin}) = \frac{1}{N}(((H-I)\epsilon)^T(H-I)\epsilon)$$

$$\mathbf{E}_{in}(w_{lin}) = \frac{1}{N}(\epsilon^T(H-I)^T(H-I)\epsilon)$$

Now we use the property from question 3.3:

$$(I-H) = -1(H-I)$$

and we manipulate our equation to get $(I-H)$ instead of $(H-I)$

$$\mathbf{E}_{in}(w_{lin}) = \frac{1}{N}(\epsilon^T(I-H)^2\epsilon(-1)^2)$$

$$\mathbf{E}_{in}(w_{lin}) = \frac{1}{N}(\epsilon^T(I-H)\epsilon)$$

$$\mathbf{E}_{in}(w_{lin}) = \frac{1}{N}\epsilon^T\epsilon - \frac{1}{N}\epsilon^T H\epsilon$$

d

$$\mathbf{E}_{in}(w_{lin}) = \frac{1}{N}\epsilon^T\epsilon - \frac{1}{N}\epsilon^T H\epsilon$$

$$\mathbf{E}_D[\mathbf{E}_{in}(w_{lin})] = \mathbf{E}_D[\frac{1}{N}\epsilon^T\epsilon - \frac{1}{N}\epsilon^T H\epsilon]$$

$$\mathbf{E}_D[\mathbf{E}_{in}(w_{lin})] = \frac{1}{N}\mathbf{E}_D[\epsilon^T\epsilon] - \frac{1}{N}\mathbf{E}_D[\epsilon^T H\epsilon]$$

So now we need to evaluate:

$$\frac{1}{N}\mathbf{E}_D[\epsilon^T\epsilon] - \frac{1}{N}\mathbf{E}_D[\epsilon^T H\epsilon]$$

Left Side
We know, by the definition of variance, that

$$E[\epsilon^2] = \sigma^2$$

Since there are N elements in ϵ, this becomes:

$$\frac{1}{N}\mathbf{E}_D[\epsilon^T\epsilon] = \frac{1}{N}N\sigma^2$$

Right Side
We have this expression:

$$\frac{1}{N}\mathbf{E}_D[\epsilon^T H\epsilon]$$

We will break this down by looking at its dimensions. We know that the dimensions of $\epsilon^T H\epsilon$ are (1xN)(NxN)(Nx1). When we evaluate this, we get a final number, or a (1x1) matrix. That means that this matrix is essentially a very large sum, as shown here:

$$\sum_{i=0}^{N}\sum_{j=0}^{N}\epsilon_i\epsilon_j H_{i,j}$$

Now that we have this, we can bring in our property from problem 3.3. We can split this into two parts by taking away the diagonal elements in the matrices. This is shown here:

$$\sum_{i=0}^{N}\epsilon_i\epsilon_i H_{i,i} + \sum_{i=0}^{N}\sum_{j=0,j!=i}^{N}\epsilon_i\epsilon_j H_{i,j}$$

The left side of this expression can be evaluated by using the trace(H) quality. Since there are (d+1) elements, and we know $\mathbf{E}[\epsilon^2]$ is σ^2, this is equal to $(d+1)\sigma^2$

We are almost there. We combine all of this to get

$$\mathbf{E}_D[\mathbf{E}_{in}(w_{lin})] = \frac{1}{N}N\sigma^2 - \frac{1}{N}(d+1)\sigma^2$$

$$\mathbf{E}_D[\mathbf{E}_{in}(w_{lin})] = \sigma^2 - \frac{1}{N}(d+1)\sigma^2$$

e

Let's first get what we were given, and evaluate it with some matrix algebra:

$$\hat{y} - y_{test} = H\epsilon - \epsilon^{'}$$

We can use our previous problem to lead us to:

$$\mathbf{E}_{test}(w_{lin}) = \frac{1}{N}(H\epsilon - \epsilon^{'})^T(H\epsilon - \epsilon^{'})$$

$$\mathbf{E}_{test}(w_{lin}) = \frac{1}{N}(\epsilon^T H^T H\epsilon - \epsilon^T H^T\epsilon^{'} - \epsilon^{'T}H\epsilon + \epsilon^{'T}H\epsilon^{'})$$

Then, we know from the book that $H^2 = H$

$$\mathbf{E}_{test}(w_{lin}) = \frac{1}{N}(\epsilon^T H\epsilon - \epsilon^T H^T\epsilon^{'} - \epsilon^{'T}H\epsilon + \epsilon^{'T}H\epsilon^{'})$$

Now, we get the expected value of this with respect to D and ϵ

$$\mathbf{E}_{D,\epsilon}[E_{test}(w_{lin})] = E_{D,\epsilon}[(\epsilon^T H\epsilon - \epsilon^T H^T\epsilon^{'} - \epsilon^{'T}H\epsilon + \epsilon^{'T}H\epsilon^{'})]$$

$$\mathbf{E}_{D,\epsilon}[E_{test}(w_{lin})] = \frac{1}{N}E[\epsilon^T H\epsilon] - \frac{1}{N}E[\epsilon^T H^T\epsilon^{'}] - \frac{1}{N}E[\epsilon^{'T}H\epsilon] + \frac{1}{N}E[\epsilon^{'T}\epsilon^{'}]$$

Now we have four terms that we need to evaluate individually.

$$\frac{1}{N}E[\epsilon^T H\epsilon]$$

$$= \frac{1}{N}(N\sigma^2)$$

$$= \sigma^2$$

The next two terms are evaluated the same as each other. We can evaluate them into

$$\frac{1}{N}\sum_{i=0}^{N=1} E[\epsilon_i^{'}]E[\epsilon_j]H_{i,j}$$

and

$$\frac{1}{N}\sum_{i=0}^{N=1} E[\epsilon_i]E[\epsilon_j^{'}]H_{i,j}$$

We can separate the expected value of each ϵ because they are different this time. These expected values are 0, so both of these expressions are 0.

For this last expression, we get it like this by using a lot of techniques we used earlier in the homework:

$$\frac{1}{N}E[\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\epsilon_i\epsilon_j H_{i,j}^T]$$

$$\frac{1}{N}(\sum_{i=0}^{N-1} E[\epsilon_j^2]E[H_{i,i}^T] + \sum_{i,j\geq 0; i!=j}^{N-1} E[\epsilon_i]E[\epsilon^j]E[H_{i,j}^T])$$

We know from previous problems that trace(H) = d+1. Since we are looking at H^T, the trace is the same. This is equal to:

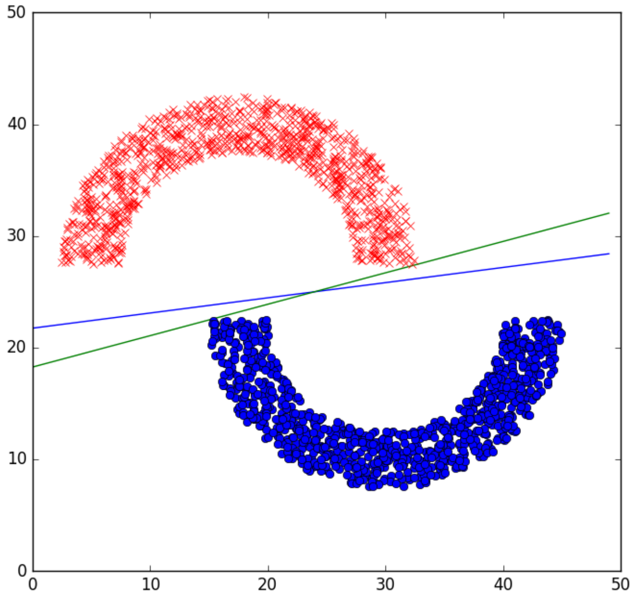
$$\frac{1}{N}((d+1)\sigma^2 + 0)$$

$$\frac{1}{N}((d+1)\sigma^2$$

Now, we combine all of this to get our final answer:

$$\sigma^2(1 + \frac{d+1}{N})$$

Problem 3.1



a and b

In the above image, the green line is the PLA, and the blue line is w_{lin}, the regression algorithm.

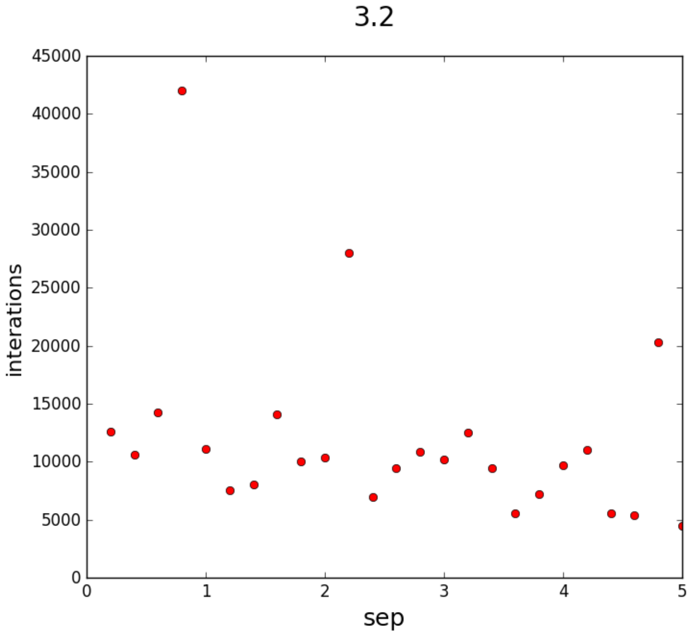
The PLA gave me a line of $x_2 = 0.15 * x_1 + 22.1$. It terminated after a lot of iterations: 11,545

The regression algorithm gave me a line that was $x_2 = 0.064 * x_1$.

Between the two, the regression algorithm was much more consistent in calculation time. The PLA (as shown in 3.2), is quite unpredictable. Also, the PLA seemed to give lines that JUST BARELY classified the data correctly. The regression algorithm would give the same line every time, as it is just a calculation.

I also noted speeds. For this particular calculation, the PLA had a chance to run for a very long time. The regression algorithm, however, was very quick.

Problem 3.2



The above image is the output when varying sep from 0.2 to 5.

The only clear trend seems to be a downward trend in iterations as sep increases. This is because as sep increases, there are more possible hypotheses that fit all the data points. However, the jumpy nature of this plot also makes sense, as PLA is by nature a very volatile algorithm.

The bound we proved in problem 1.3e is: $t < \frac{R^2||w^*||^2}{\rho}$.

This bound mostly stays the same. Both R and ρ increase as sep increases, because the distance that points are from the termination line increases. Both are a squared factor, so the bound result is approximately the same.

Problem 3.8

We will start with what we are given:

$$E_{out}(h) = \mathbf{E}[h(x) - y)^2]$$

$$= \mathbf{E}[h^2(x) - 2h(x)y + y^2$$

$$= \mathbf{E}[h^2(x)] - \mathbf{E}[2h(x)y] + \mathbf{E}[y^2]$$

Because you are looking for the optimal (minimum) hypotheses, $E[y] = E[y|x]$:

$$= h^2(x) - 2h(x)\mathbf{E}[y|x] + \mathbf{E}[y^2|x]$$

$$\frac{d\mathbf{E}_{out}}{dh(x)} = 2h(x) - 2\mathbf{E}[y|x] = 0$$

$$h(x) - \mathbf{E}[y|x] = 0$$

$$h(x) = \mathbf{E}[y|x]$$

Now, we will show $\mathbf{E}[\epsilon(x)] = 0$:

$$y = h^*(x) + \epsilon(x)$$

$$\mathbf{E}[y] = \mathbf{E}[h^*(x)] + \mathbf{E}[\epsilon(x)]$$

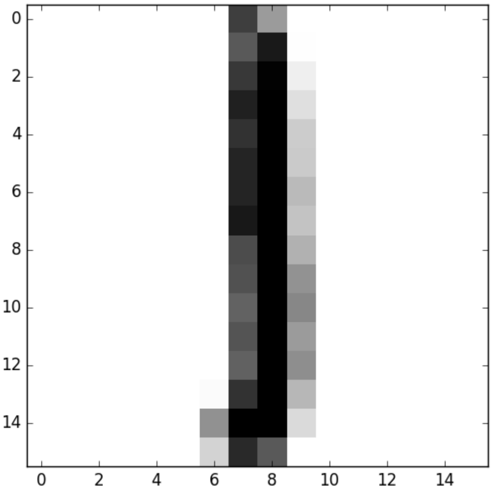
We know that $E[y] = h^*(x)$

$$\mathbf{E}[h^*(x)] = \mathbf{E}[h^*(x)] + \mathbf{E}[\epsilon(x)]$$

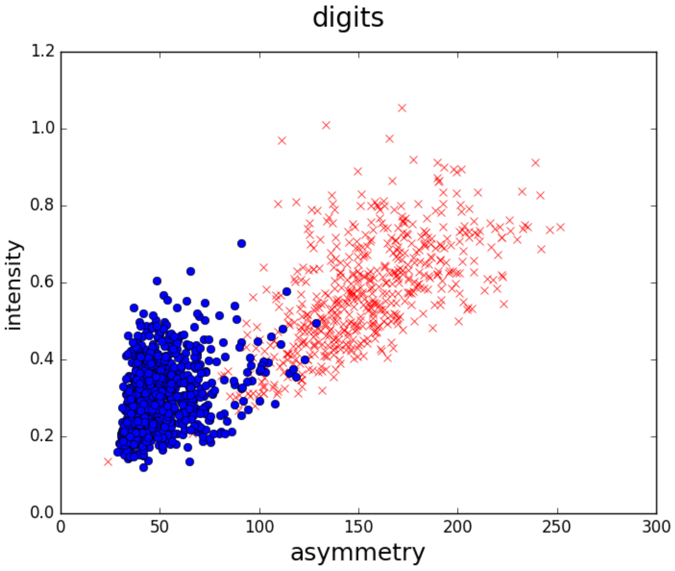
$$\mathbf{E}[\epsilon(x)] = 0$$

Extra Problem

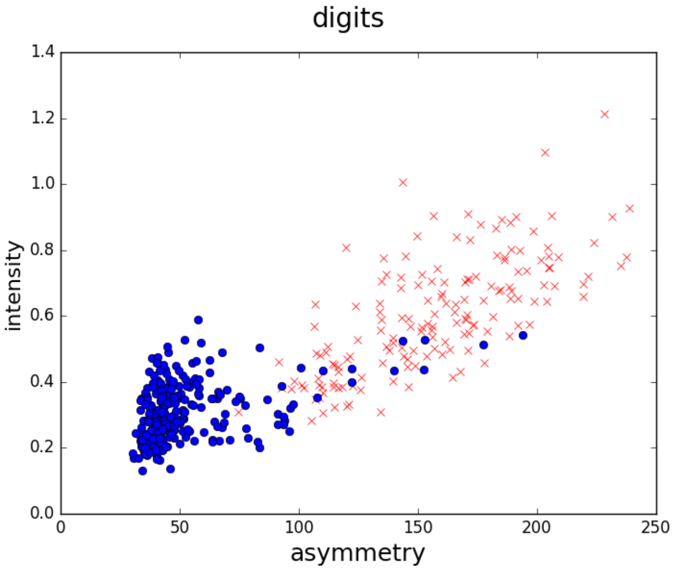
The image below is for part a. It is just a visual representation of the digit ”1”, as provided in the data.



The image below shows symmetry vs. intensity for the training data:



The image below shows symmetry vs. intensity for the test data:



Intensity
The intensity is defined by summing all the points and averaging the value. This is:

$$\frac{1}{256}\sum_{i=0}^{256}(D[i])$$

Where D[i] is the value of the digit at index i.

Symmetry

I achieved symmetry by comparing the left half of the image to the right half, and then comparing the top half of the image by the bottom half. I then add these two numbers to get total symmetry.

$$\sum_{i=0,j=0}^{i=7,j=16} P[16j+i]-P[16(j+1)-(i+1)]+\sum_{i=0,j=0}^{i=16,j=7} P[16j+i]-P[(15-j)*16-i]$$

The pixels that were different that it's opposite were penalized, essentially.

Summary

In the end, this method was rather effective. As seen in the photos, most of the 1's and 5's were separable. However, a few outliers still found their way into the wrong group.