

# Midterm Questions!

Morgan Baker

10/13/16

## 1 Question 1

This was the question from the past homework. The question asks us to take a dataset of 7291 points located in a csv file, and asks us to sort them using three different approaches. The first approach uses the pocket algorithm. I figured out the error in the old code. I kept getting instance variables mixed with local variables. However, it works now, and the output for the code can be found at Figure 1. The Eout for this method came to be .012069674941708956 after 916 iterations. Linear Regression calculated it's Eout at .01522424907420107, which is worse than the pocket algorithm. The worst is with the thrird algorithm, or the best of both worlds algorithm from before. It isn't necessarily the best of both worlds though, because the pocket algorithm got to the same Eout, but took about 350 more iterations, as shown in Figure 2. This means that in this case, the pocket algorithm by itself seems like the best bet.

## 2 Question 2

The second question is something that I had wanted to do for homework 2, but instead threw it in a graphing calculator. The quesiton asks us to make the equation from problem 2.12 to run iteratively. This is done by creating the Equation.py class. This class takes the equation

$$N \geq \frac{8}{\epsilon^2} \ln\left(\frac{4((2N)^{10}+1)}{\delta}\right)$$

and applies it to the problem. It then loops until N is equal to itself with the accuracy of about 15 decimals. It then takes the data and plots it, as shown in Figure 3. This was a fun problem to work on.

```
8956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956]
916
0.01522424907420107
[ 1916.]
```

Figure 1: The Eout of the Pocket Algorithm and Linear Regression separately.

```
1708956, 0.012069674941708956]  
1261  
0.01522424907420107
```

Figure 2: The Eout and iterations of the Pocket Algorithm that started with the weights from Linear Regression.

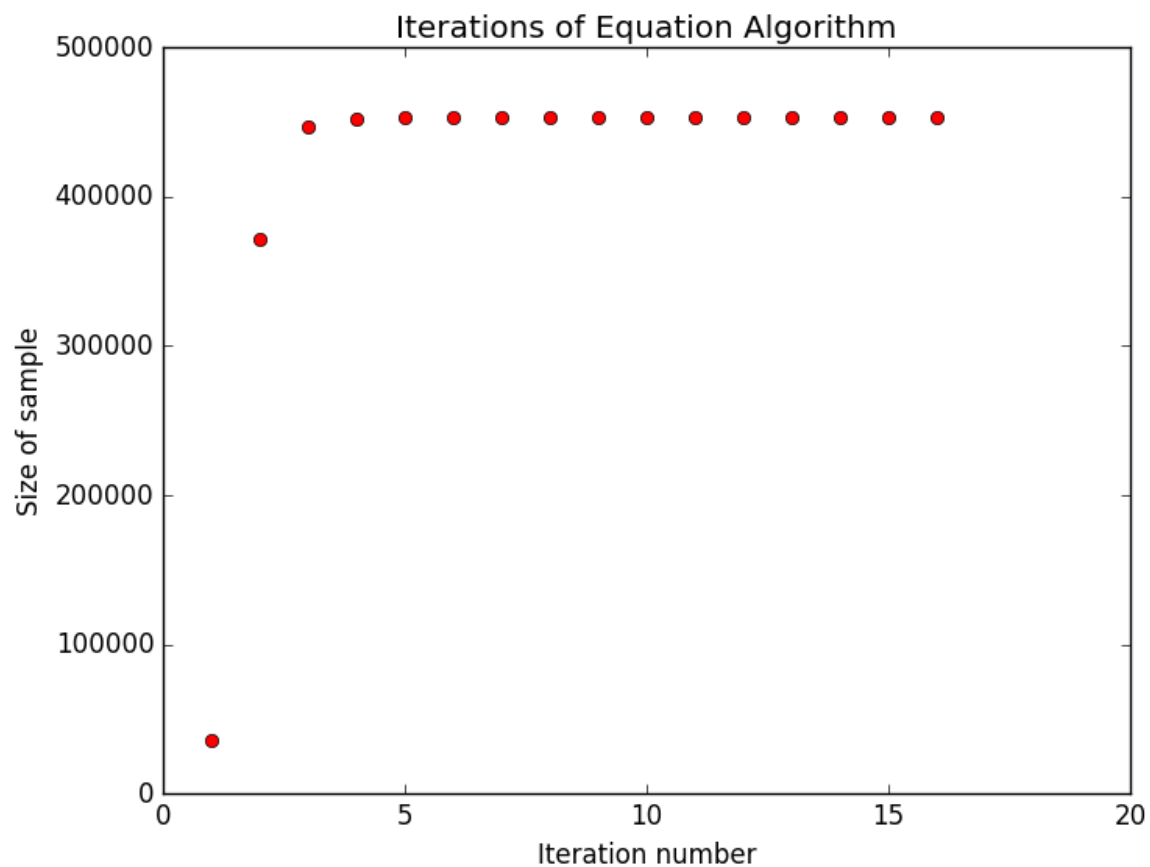


Figure 3: The plot of samples to the number of iterations.

### 3 Question 3

I wasn't really sure what to do for this, since I'm not necessarily the best artist. I can tell you that when we were first talking about Perceptrons, I saw them as another type of Transformer. "Oh no, the Perceptrons are coming", "Don't worry class, I can help with this" runs perceptron learning algorithm... Yeah this is weird.