Machine Learning Programming Exercise (Lab 1)

Hamit Kavas

2019

1) Regression Task

I have chosen the *mg* dataset for this task, which has total of 1385 points of data. I have used 100 increments of samples; data accordingly increments itself by starting at 14 until reaching 1385. Samples have been taken randomly and then, in order to smooth the values, I have averaged 10 iterations of the regression.

1. Plot the approximation error (square loss) on the training set as a function of the number of samples N, i.e., data points in the training set.

Mean squared error as function of different samples is shown in Figure1. The red line uses entire data set for the prediction, while the blue line shows the error of the prediction for the same samples in training set (in sample error).

As we can see in the very first steps of the graph, samples describe data points very well but generalize very poorly. Due to lack of variation in small sub set comparing to full data set, coefficients that obtained from small subset of the data set, red line, give better score than mean squared error of sub set, blue line.

Increment in amount of samples triggers in sample error to increase since the generalization gets better and general error decreases.

As a final result we should not aim to minimize the in sample error, when we have a very small amount of samples. That would increase the out of sample error.

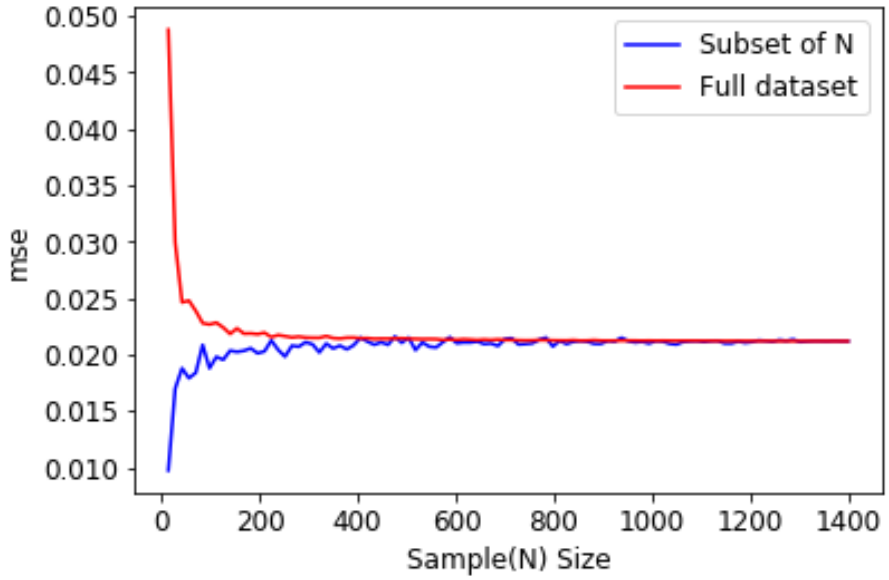


Figure 1: Mean Squared Error vs Number of samples

2. Plot the cpu-time as a function of the number of samples

Figure 2 shows the CPU time for the fitting of the weights against the amount of samples used.
CPU time against the amount of samples used is shown below. Time values are the average values of each 10 repeat.
The result is as should be, the regression takes longer the more samples we use.

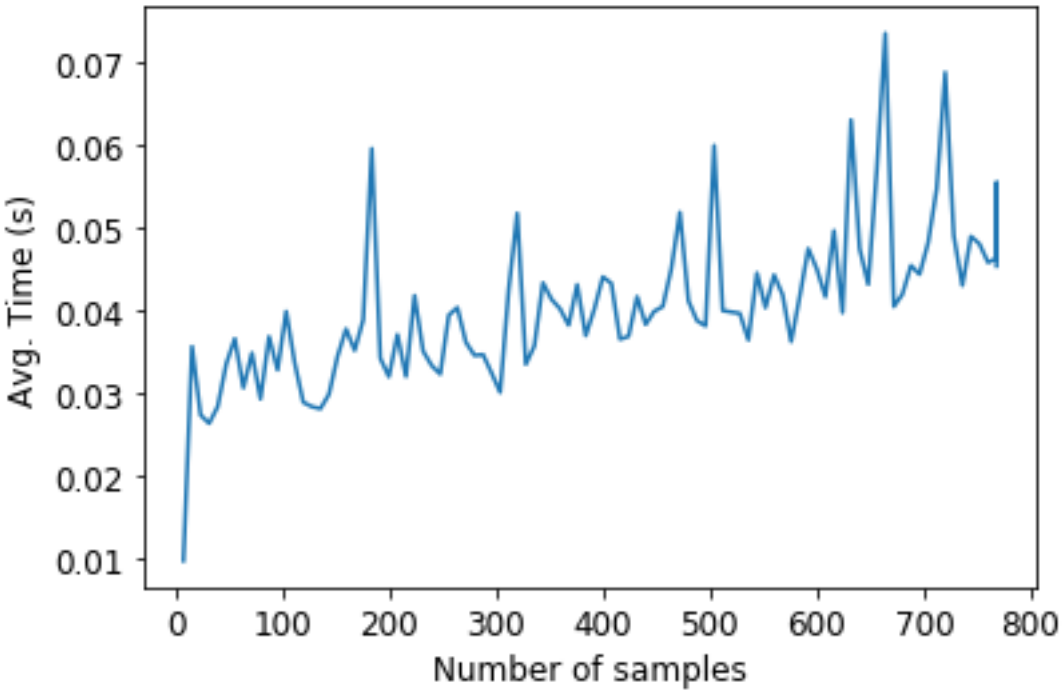


Figure 2: CPU Time vs Number of samples

3. Explain the behaviour of both curves.

Mean squared error over the full training set decreases exponentially with the function of increasing samples until we reach approximately 400 samples(See Figure1) where it stabilizes and after that decreases very slowly. For in sample error, the situation is the opposite; it increases exponentially.

This means that we have ideal amount of samples regarding in sample errors which do not change but only increase the computational time.

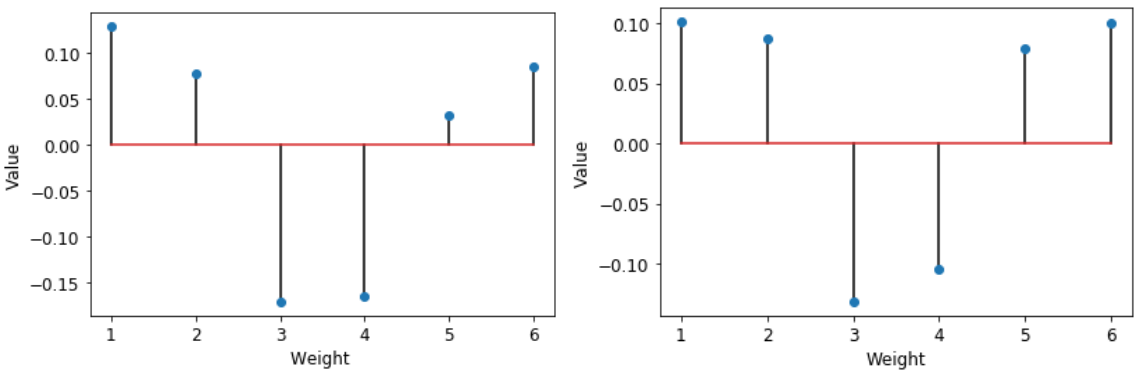
4. Plot the learned weights for two different number of training samples. Can you find an interpretation for the learned weights?

For this part of the exercise we will have three graphs which shows weights in the steps of different number of samples. We will start with our smallest sample, 14 and then finish with all elements in dataset.

If we compare figure 3a and 3b we can see that weights are quite similar, but weight number 5 is much bigger when using 140 samples, however weights 2, 4 and 6 changed in small amount.

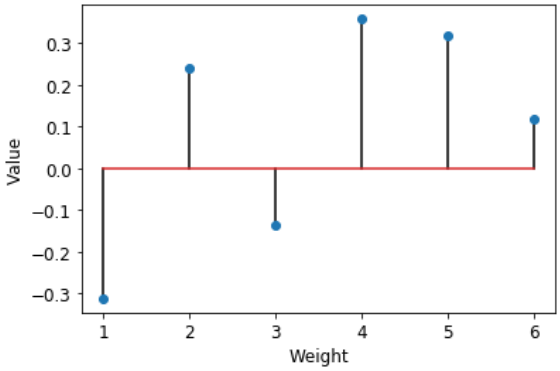
If we now compare them to the final weights, in figure 3c, the difference in change is much more higher. Such values as weight 1 and 4 opposed themselves with very huge amount of increase.

As conclusion, few samples give us small changes in the weights which would not optimize our algorithm properly. However, regardless amount of the samples we use, weight 6 seems as to be most important, while it is almost not changed at all, where weight 5 seems to be ignored as the change in that is hugest.



a) 14 samples

b) 140 samples



c) 1385 samples

Figure 3: Weights for different samples

Figure 3 shows the change in all 6 weights for different amount of samples. Deviation is very high in the beginning of the process, while they converge to 0 through end of the iterations.

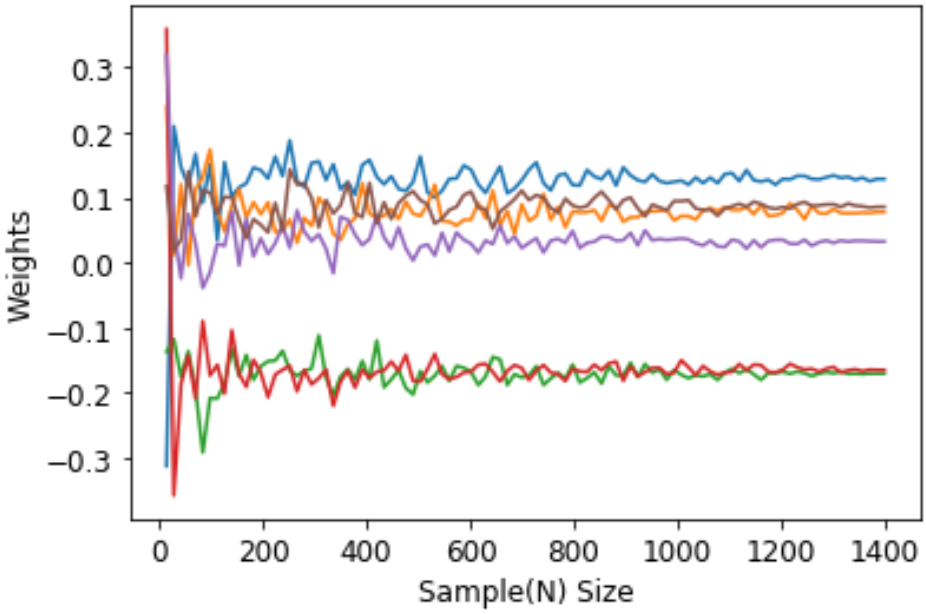


Figure 4: Weights for pointwise data values

2- Classification Task

In this part of the exercise I have used a data diabetes with 768 data points and8 features. We use increments of 100 samples until reach all elements in the data set. Like in the previous task, we average the results over 10 iterations.
For the parameters of the logistic regression method I choose:
_ solver = “lbfgs”
_ multi class = multinomial
_ C = 1e05
_ max iter = 10000

2.1 Plot the approximation error (or mean accuracy) on the training set as a function of the number of samples (i.e. data points in the training set)

Figure 5 shows the cross-entropy error as the samples used increase, and it is quite similar to the one in the regression task, the blue line represents the error when we use the same small set of samples for prediction (in sample error) and the red line shows the error when we use the weights to predict over the whole data set. The more samples we have the better we are at generalizing and our error over the whole data set decreases greatly.

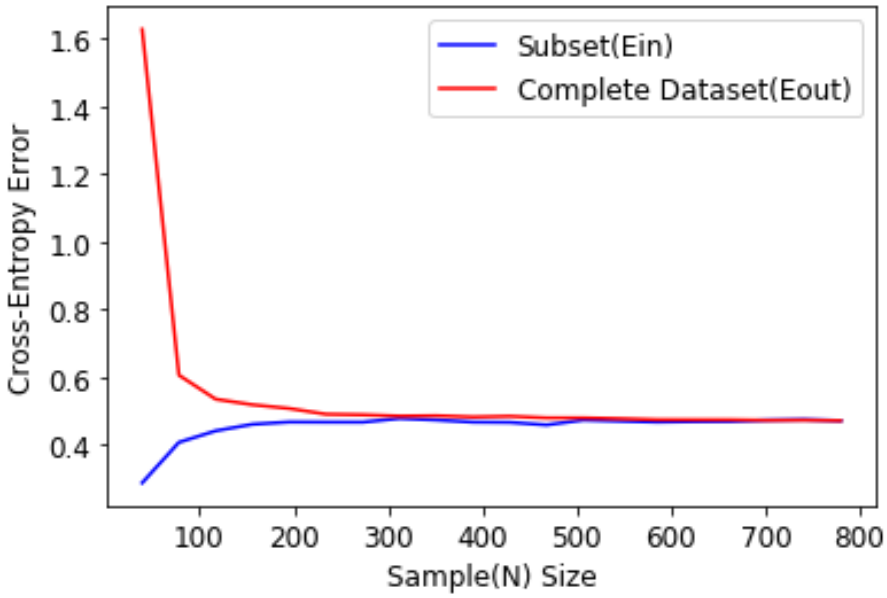


Figure 5: Cross-Entrophy Error

2.2 Plot the cpu-time as a function of the number of samples

The time that we spend to perform fitting increases when we increase the amount of samples in Figure 6. We can see here that thanks to the strong regularization of C = 1e05, the time it takes to compute the weights is really high. If we had selected C=1 it would have taken much less time.

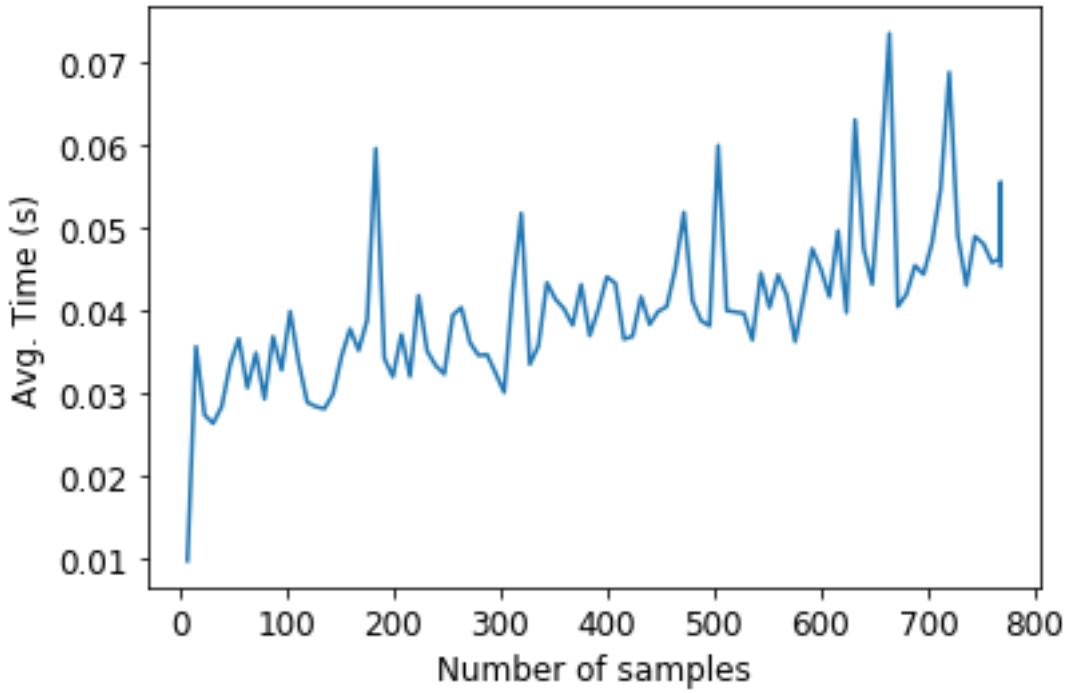


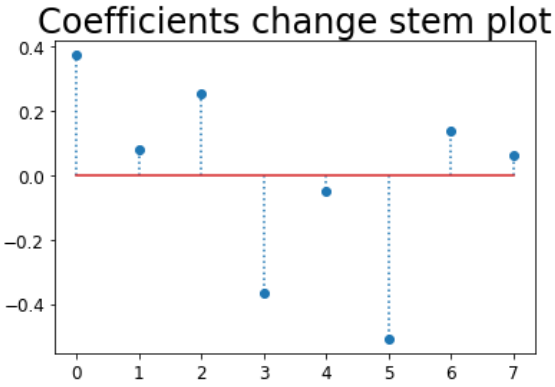
Figure 6: CPU Time vs Number of Samples

2.3 Explain the behaviour of both curves.

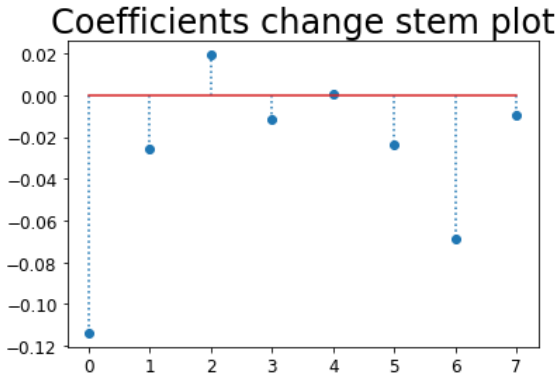
Similarly to previous task, the in sample error exponentially increases over iterations until it stabilizes (although for this case it stabilizes very early) while out of sample error suddenly decreases and in several iterations stabilizes itself.

2.4 Plot the learned weights for two different number of training samples. Can you find an interpretation for the learned weights?

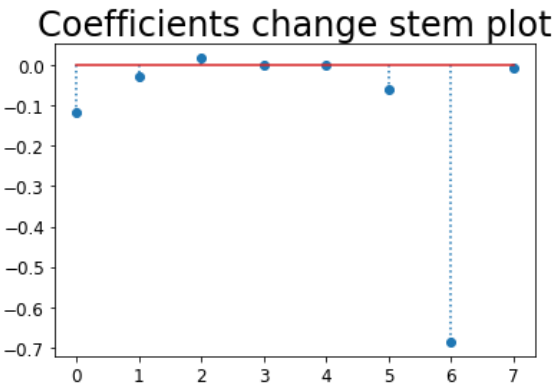
Figures below show the weights for 4 different samples from our dataset. The changes is comparively high for small samples as to be compatible with our first example.
The change is converged to zero except weight 6, which is our most inefficient coefficient.



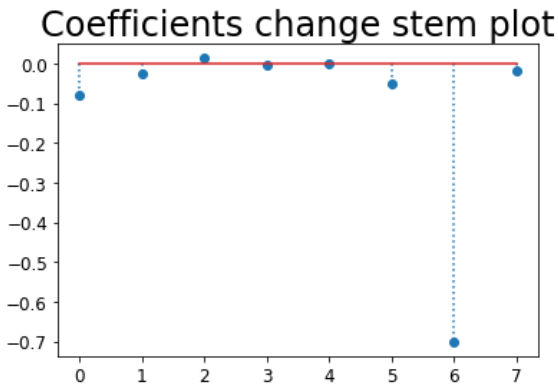
a) 8 samples



b) 208 samples



c) 808 samples



d) 768 samples

Figure 7: Weights for different samples

Figure 8 shows all weights over iterations through the samples.

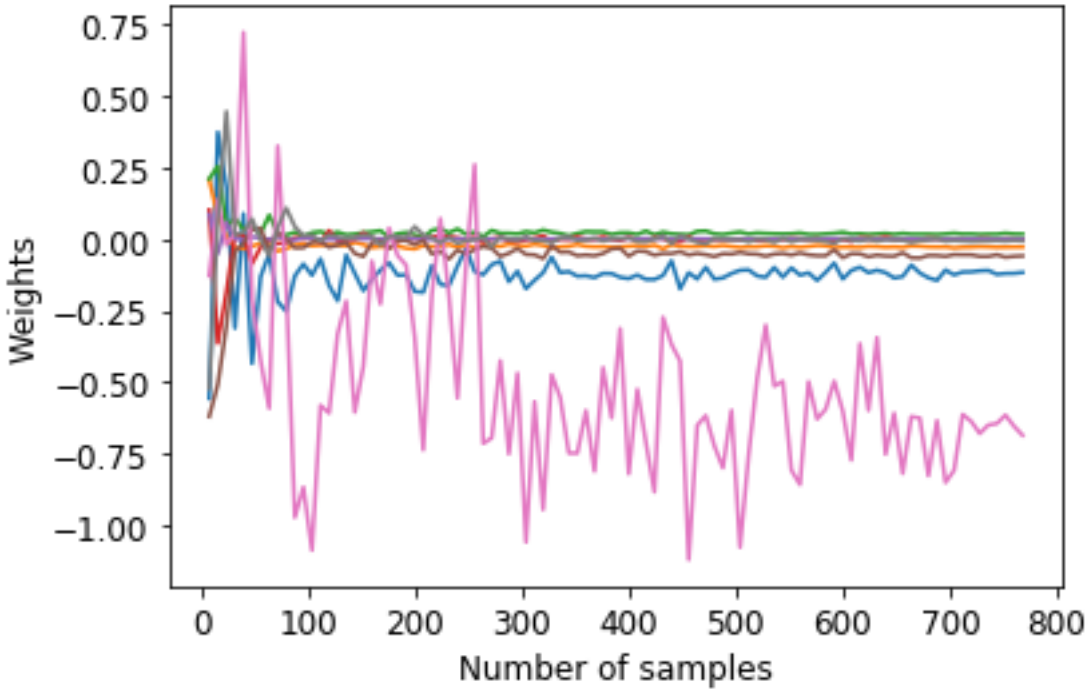


Figure 8: Weights for pointwise data values