FINAL REPORT

APRIL 11, 2020

IMPLEMENTATION ASSIGNMENT #1

PREPARED FOR

DR. XAIOLI FERN

CS 434

PREPARED BY

TEAM 8

ADAM STEWART
HAO DENG
YUHANG CHEN

Abstract

This document aims to provide a general overview of Implementation Assignment #1, while explicitly addressing questions given in the assignment description. Additionally, plotted figures will be included to more clearly illustrate the emergent trends.

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

For this assignment, our group was tasked with implementing what we have been learning about over the past few weeks in lecture: linear and logistic regression. The script invocations fit the syntax given in the assignment (though you can find a readme file in our GitHub repository that explicitly indicates how to run the scripts for each part of the assignment). Running these scripts will generate .png files for the relevant plots within the same directory. For the sake of conciseness, only some of these plots have been included within this report to provide clearer visualizations. In the next few sections, we will be answering the questions as they were given in Implementation Assignment #1.

2 LINEAR REGRESSION

2.1 Part 1

Q: *Report the learned weight vector.*

A: The learned weight vector is:

$$\begin{matrix} \langle 3.67103960e+01, & -1.10220511e-01, & 4.25270181e-02, & 9.94268803e-03, & 4.03688262e+00, \\ -1.81193844e+01, & 3.91213593e+00, & -3.24572263e-03, & 4.03688262e+00, & -1.81193844e+01, \\ 3.91213593e+00, & -3.24572263e-03, & 9.33221332e-03, & -5.87431614e-01\rangle & \end{matrix}$$

2.2 Part 2

Q: *Report the training and testing ASEs respectively. Which one is larger? Is it consistent with your expectation?*

A: The ASE over the training data is: 21.635964362901742 and the ASE over the testing data is: 23.69036052568365. The ASE over testing data seems to be larger. This is consistent with our expectation because the model is trained with dataset A, and when testing the trained model with dataset B, the ASE is higher due to the difference and variations in the data from datasets. The ASE over the testing data is expected to be larger or higher than the ASE over the training data. And the ASE over the testing data is more objective to represent the accuracy of the trained model.

2.3 Part 3

Q: *How does this change influence the ASE on the training and testing data? Provide an explanation for this influence.*

A: If you remove all 1 from the dummy data, there will only be 13 features in the weight vector. As a result, b will be missing from the function, instead of the equation we might normally expect to see:

$$Y = b + w_1X_1 + w_2X_2 + ... + w_{13}X_{13}$$

As a result, we are missing the bias term and thus, the ASE will be higher and the model will be less accurate.

2.4 Part 4

Q: *What trends do you observe for training and testing ASEs respectively? In general, how do you expect adding more features to influence the training ASE? How about testing ASE? Why?*

A: With additional random features included in the model, we can see the downward trend for the training ASE and upward trend for the testing ASE respectively. (Plots of these trends can be seen in figures 1 and 2 on the next page.) In general, the training ASE will be expected to gradually diminish as more random features are added; on the contrary, the testing ASE will be expected to increase. This is due to the complexity of the model from increasing the number

of features. A more complex model means a more overfitting model, which results in a higher accuracy with training dataset but lower accuracy with testing dataset. Especially when the added features are random; indicates that they are useless.

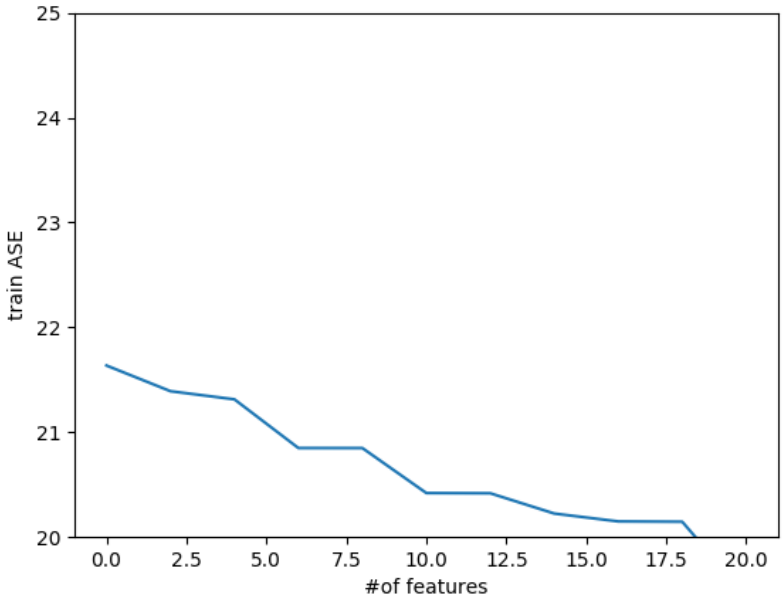

Fig. 1: A plot of Average Squared Error (ASE) over number of features added while training the model.

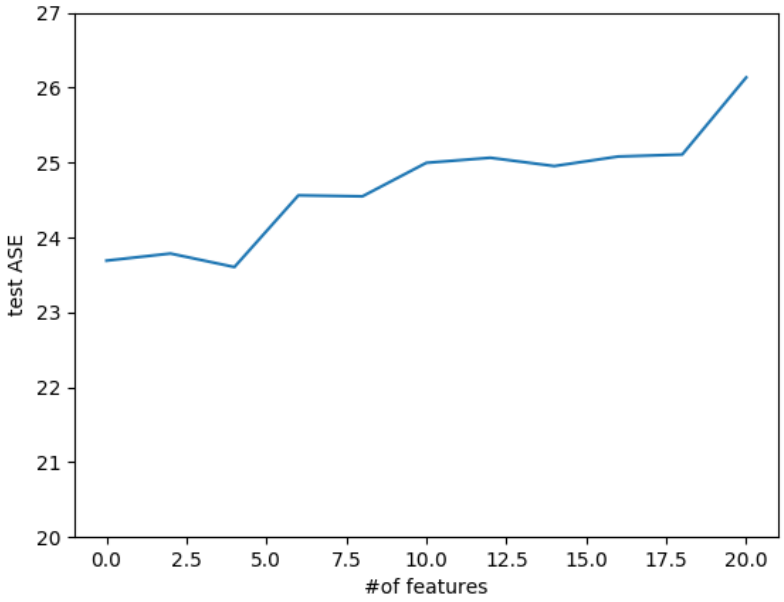

Fig. 2: A plot of Average Squared Error (ASE) over number of features added for the testing of the model.

3 Logistic Regression with Regularization

3.1 Part 1

Q: *The behavior of Gradient descent can be strongly influenced by the learning rate. Experiment with different learning rates, report your observation on the convergence behavior of the gradient descent algorithm. For each gradient descent iteration, plot the training accuracy and the testing accuracy of your model as a function of the number of gradient descent iterations. What trend do you observe?*

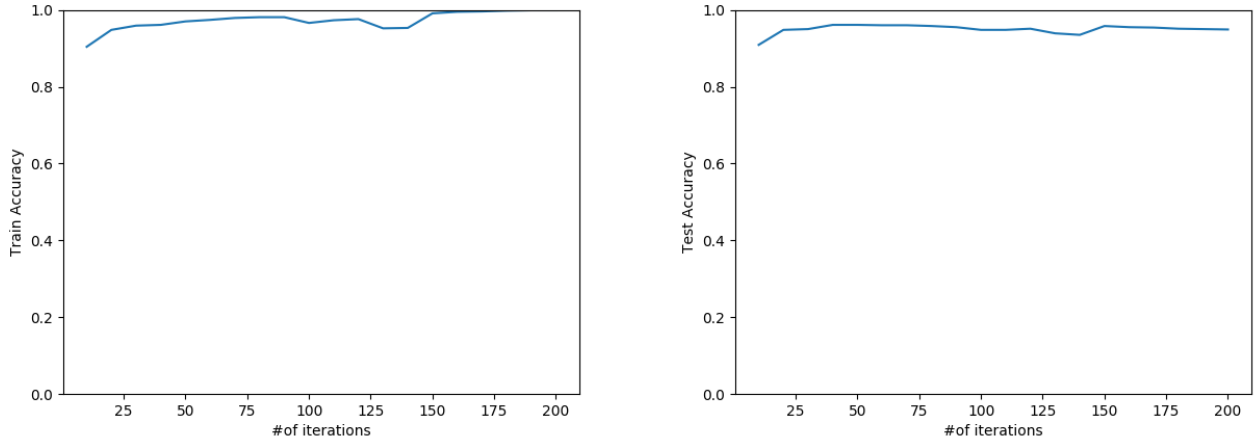

Fig. 3: Plots of the training accuracy (left) and testing accuracy (right) of our model as a function of the number of gradient descent iterations.

A: We were able to converge toward a relatively stable training and testing rate with the $learning\ rate$ set to 0.01 as can be seen above in figure 3. We decided to set the stopping condition by using a fixed number of iterations. We recorded the accuracy every 10 iterations up to 200 total iterations. It is clear that there is a significant increase in both training and testing accuracy from 10 to 20 iterations. The training accuracy converges toward 1.0 as the number of iterations approaches 200, with $MaxTrainingAccuracy = 0.999$. However, the testing accuracy peaks early, with $MaxTestingAccuracy = 0.961$ at 40 and 50 iterations, but tends toward 0.95 with some instability.

3.2 Part 2

Q: *Here we will explore L2 regularization... Find the gradient for this objective function and modify the batch gradient descent algorithm with this new gradient. Provide the pseudo code for your modified algorithm.*

A: Gradient_Descent(*w*, *lambda*, *learning_rate*) {

Create an array for new weights

Get previous weights transposed, $\mathbf{w}^T$

Initialize array for sum of the loss function, sum_li

For each $\mathbf{x}_i$ in the dataset {

decrease_rate = get_sigmoid($\mathbf{w}^T \times \mathbf{x}_i$)

decrease_rate = decrease_rate - y_i

```
        li = rate*x_i
        sum_li += li
      }
      new_w = w - learning_rate * ( sum_li - (lambda*w) )
      new_w[0] = new_w[0] - ( learning_rate * sum_li )
      return new_w
  }
```

3.3 Part 3

Q: *Report the training and testing accuracies (i.e., the percentage of correct predictions) achieved by the weight vectors learned with different λ values. Discuss your results in terms of the relationship between training/testing performance and the λ values.*

A: The figures below depict the training and testing accuracy over the number of lambdas used. (We tried lambda values: [0.01, 1.0, 100.0, 1000.0, 10000.0, 100000.0].) Typically, there is a balance between too few and too many lambdas that must be struck. (If the lambda goes too low, it may cause you to overfit, but if it goes too high, it may cause you to underfit.) To find this balance, we had to turn our learning rate down significantly to avoid the "overflow encountered in exp" warning once we got to higher lambda values. Consequently, our accuracy is much lower as can be seen in the figures below. Though it is hard to see in the graphs below, 10000 lambdas proves to bee the best option out of the lambdas we tried, with a training accuracy of 0.74 and a testing accuracy of 0.725.

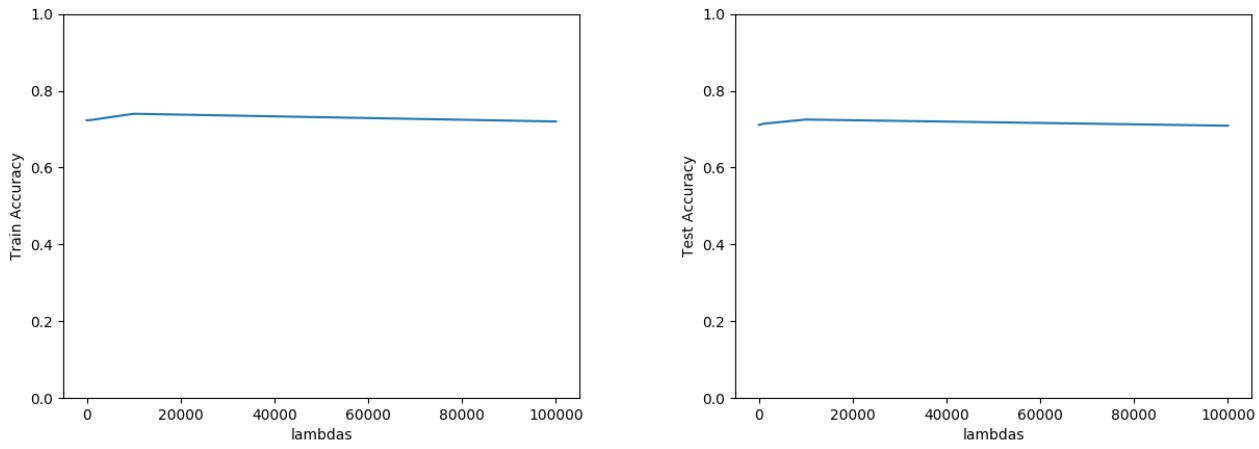

Fig. 4: Plots of the training accuracy (left) and testing accuracy (right) of our model as a function of the number of lambdas.