Logistic Regression Training vs. Testing

Lab Report # 6

|  |  |  |  |
| --- | --- | --- | --- |
| Sidney Carlo P. Lopez (11222522)  sidneylopez18@gmail.com | |  | Jeremy B. Reccion (11210192)  jeremybreccion@yahoo.com |
| Carl Michael O. Sy (11208929)  carl.michael.sy@gmail.com | Wellington S. Villasin (11209534)  wellvill08@gmail.com | | |
|  |  | | |

*Abstract*— a type of a statistical method where it is used to analyzed the list of given set of data that consists one or more numbers of given independent variables which would predict the end result is called Logistic Regression.

Keywords—regression; machine; logistic; statistical; testing;data

1. INTRODUCTION

Logistic Regression [1] is a type of regression that would predict the occurrence of an occasion by adding information to the logistic capacity. It has the capability where there is an employment of the indicating variables which could be categorical and numerical. This kind of regression would utilize the situations for learning in performing the right decisions. This would also help compute the relationship of the dependent variables over single or numerous numbers of independent variables with the use of probability estimation of a logistic function. It is important to know the outcome of the values with the use of a linear function.

1. OBJECTIVES

* Implement the function and gradient computations for logistic regression
* Predicting the values of the y(i) using a linear function
* Using the code that you have created to classify the images of digits from the MNIST dataset

1. DATA AND RESULTS
2. Calls ex1\_load\_mnist.m to load the MNIST training and testing data. In addition to loading the pixel values into a matrix X (so that that j’th pixel of the i’th example is Xji=x(i)j) and the labels into a row-vector y, it will also perform some simple normalizations of the pixel intensities so that they tend to have zero mean and unit variance. Even though the MNIST dataset contains 10 different digits (0-9), in this exercise we will only load the 0 and 1 digits — the ex1\_load\_mnist function will do this for you.

Load the MNIST data for this exercise.

train.X and test.X will contain the training and testing images.

Each matrix has size [n,m] where:

m is the number of examples.

n is the number of pixels in each image.

train.y and test.y will contain the corresponding labels (0 or 1).

C:\Users\sidneycarlolopez\Desktop\1.JPG

Add row of 1s to the dataset to act as an intercept term.

C:\Users\sidneycarlolopez\Desktop\1.JPG

Training set dimensions

C:\Users\sidneycarlolopez\Desktop\1.JPG

Train logistic regression classifier using minFunc



First, we initialize theta to some small random values.

C:\Users\sidneycarlolopez\Desktop\1.JPG

Call minFunc with the logistic\_regression.m file as the objective function.

TODO: Implement batch logistic regression in the logistic\_regression.m file!

C:\Users\sidneycarlolopez\Desktop\1.JPG

Now, call minFunc again with logistic\_regression\_vec.m as objective.

TODO: Implement batch logistic regression in logistic\_regression\_vec.m using

MATLAB's vectorization features to speed up your code. Compare the running

time for your logistic\_regression.m and logistic\_regression\_vec.m implementations.

Uncomment the lines below to run your vectorized code.

theta = rand(n,1)\*0.001;

tic;

theta=minFunc(@logistic\_regression\_vec, theta, options, train.X, train.y);

fprintf('Optimization took %f seconds.\n', toc);

Print out training accuracy.

C:\Users\sidneycarlolopez\Desktop\1.JPG

Print out accuracy on the test set.

C:\Users\sidneycarlolopez\Desktop\1.JPG

1. The code will append a row of 1’s so that θ0 will act as an intercept term

C:\Users\sidneycarlolopez\Desktop\1.JPG

Load the training data

C:\Users\sidneycarlolopez\Desktop\1.JPG



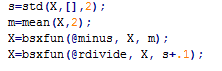
Take only the 0 and 1 digits



Randomly shuffle the data

C:\Users\sidneycarlolopez\Desktop\1.JPG

We standardize the data so that each pixel will have roughly zero mean and unit variance.



Place these in the training set



Load the testing data





Take only the 0 and 1 digits

C:\Users\sidneycarlolopez\Desktop\1.JPG

Randomly shuffle the data

C:\Users\sidneycarlolopez\Desktop\1.JPG

Standardize using the same mean and scale as the training data.



Place these in the testing set

C:\Users\sidneycarlolopez\Desktop\1.JPG

1. The code calls minFunc with the logistic\_regression.m file as objective function. Your job will be to fill in logistic\_regression.m to return the objective function value and its gradient.

C:\Users\sidneycarlolopez\Desktop\1.JPG

Arguments:

theta - A column vector containing the parameter values to optimize.

X - The examples stored in a matrix.

X(i,j) is the i'th coordinate of the j'th example.

y - The label for each example. y(j) is the j'th example's label.

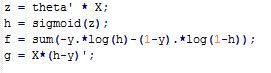
C:\Users\sidneycarlolopez\Desktop\1.JPG

Initialize objective value and gradient.

C:\Users\sidneycarlolopez\Desktop\1.JPG

TODO: Compute the objective function by looping over the dataset and summingup the objective values for each example. Store the result in 'f'.

TODO: Compute the gradient of the objective by looping over the dataset and summing up the gradients (df/dtheta) for each example. Store the result in 'g'.

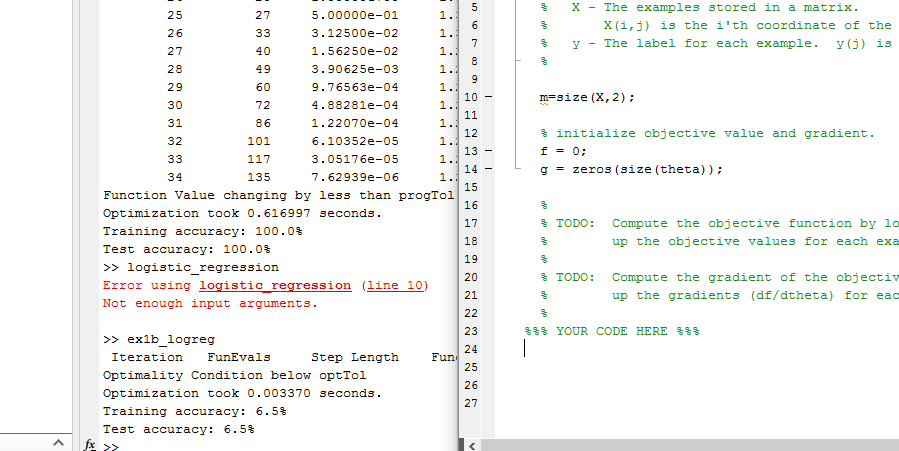


1. After minFunc completes, the classification accuracy on the training set and test set will be printed out.
2. Implement logistic\_regression.m to loop over all of the training examples x(i) and compute the objective J(θ;X,y). Store the resulting objective value into the variable f. You must also compute the gradient ∇θJ(θ;X,y) and store it into the variable g. Once you have completed these tasks, you will be able to run the ex1b\_logreg.m script to train the classifier and test it.

Preliminary Report Results:

Training Accuracy: 6.5%

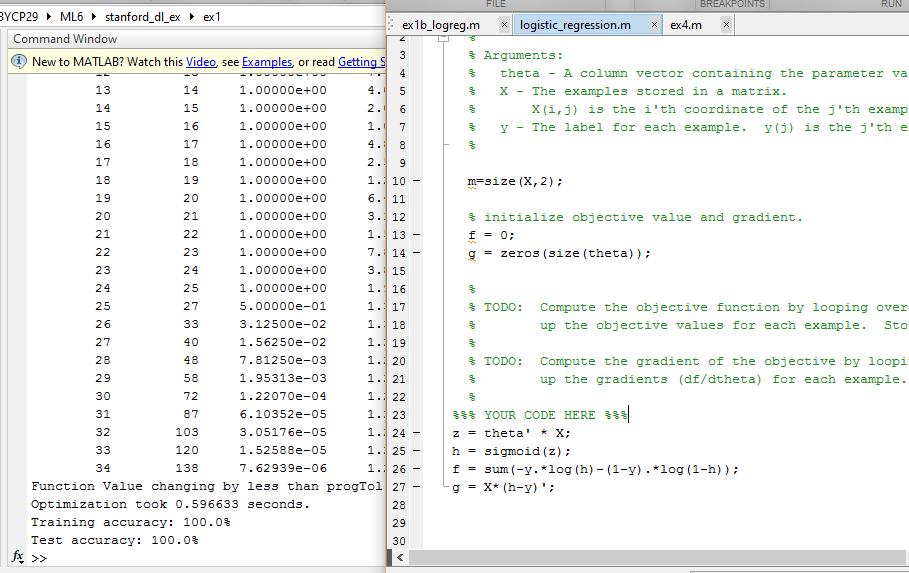
Test Accuracy: 6.5%



Final Results

Training Results: 100%

Test Accuracy: 100%



1. ANALYSIS AND CONCLUSION

For this experiment, the group followed the given procedures and analyzed the results. After setting up the codes, the group run the program or code (the one without the tasked code to be added) and a training and testing accuracy of 6-8% was observed. After implementing the tasked code in the logistic\_regression.m file, the resulting accuracy for both training and testing was 100%. The code that the group implemented consists of vectorized version of the logistic regression. The variable h represented the sigmoid or hypothesis and this was used in f wherein f is the cost

function. Lastly, variable g is the gradient descent.In conclusion, the accuracy of 100% was achieved.

REFERENCES

[1] Aimotion.blogspot.com, 'Machine Learning with Python - Logistic Regression - Artificial Intelligence in Motion', 2011. [Online]. Available: http://aimotion.blogspot.com/2011/11/machine-learning-with-python-logistic.html. [Accessed: 22- Sep- 2015].