Unsupervised Logistic Regression

DANTING, Jules A. (10934375)

DE VERA, Luis Paolo M. (11113944)

LIM, Alyssa Raphaella (1128941)

RAMOS, Brian Justin S. (11127120)

VALDEZ, Esteen Rae Salvador (1111132)

**1 INTRODUCTION AND MAIN CONCEPTS**

Unsupervised Learning is one type of machine learning algorithm that has a goal to let the machine learn without telling or showing on how to do it. [1]. It finds similarities in the training data but needs to have enough set of data to have a more accurate results. Unsupervised Logistic Regression is used to predict whether it is true or false, 1 or 0 respectively. It predicts the probability of the given data set whether it is classified as 0, false, or 1, true. Its function or the sigmoid is used to make the range of the training data to either 1 or 0. The goal of the experiment is to have a value of theta that can make the probability large so that it can belong to class 1 or make the probability small so that it can belong to class 0.

**2 METHODOLOGY**

**2.1 Implementation**

In the ex1/ directory of the starter code package you will find the file ex1\_linreg.m which contains the makings of a simple linear regression experiment. This file performs most of the boiler-plate steps for you:

1. The data is loaded from housing.data. An extra ‘1’ feature is added to the dataset so that θ1 will act as an intercept term in the linear function.
2. The examples in the dataset are randomly shuffled and the data is then split into a training and testing set. The features that are used as input to the learning algorithm are stored in the variables train.X and test.X. The target value to be predicted is the estimated house price for each example. The prices are stored in “train.y” and “test.y”, respectively, for the training and testing examples. You will use the training set to find the best choice of θ for predicting the house prices and then check its performance on the testing set.
3. The code calls the minFunc optimization package. minFunc will attempt to find the best choice of θ by minimizing the objective function implemented in linear\_regression.m. It will be your job to implement linear\_regression.m to compute the objective function value and the gradient with respect to the parameters.
4. After minFunc completes (i.e., after training is finished), the training and testing error is printed out. Optionally, it will plot a quick visualization of the predicted and actual prices for the examples in the test set.

The ex1\_linreg.m file calls the linear\_regression.m file that must be filled in with your code. The linear\_regression.m file receives the training data X, the training target values (house prices) y, and the current parameters θ.

Complete the following steps for this exercise:

1. Fill in the linear\_regression.m file to compute J(θ) for the linear regression problem as defined earlier. Store the computed value in the variable f.

You may complete both of these steps by looping over the examples in the training set (the columns of the data matrix X) and, for each one, adding its contribution to f and g. We will create a faster version in the next exercise.

**2.2 Evaluation**

Once you complete the exercise successfully, the results should match the expected results from the manual

**3 RESULTS AND DISCUSSION**

**3.1 OUTPUT**

Iteration 1:

Step Length: 1.20841e-006

Function Val: 6.56813e+003

Opt Cond: 4.08993e+003

Iteration 32:

Step Length: 0.00000e+000

Function Val: 4.84302e-001

Opt Cond: 1.36673e-001

Optimization took 9.185740 seconds.

Training accuracy: 100.0%

Test accuracy: 100.0%

**3.2 MATLAB CODE**

logistic\_regression.m

**function** **[**f**,**g**]** **=** logistic\_regression**(**theta**,** X**,**y**)**

%

% Arguments:

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

**3.3 DISCUSSION**

gradient.

f **=** 0**;**

g **=** zeros**(**size**(**theta**));**

%

% TODO: Compute the objective function by looping over the dataset and summing

% up 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'.

%

%%% YOUR CODE HERE %%%

%for i = 1:m

%h=1/(1+2.718^(-theta' \* X(:,i)));

%f = f + sum(-y.\*log(h) - (1-y).\*log(1-h));

%end

%f = (1/m)\*f;

%for i = 1:m

% h=1/(1+2.718^(-theta' \* X(:,i)));

% g = g + X(:,i)\*(h - y(i));

%end

%g = 1/m\*g;

f **=** **-**sum**(**y**.\***log**(**sigmoid**(**theta**'\***X**))** **+** **(**1**-**y**).\***log**(**1 **-** sigmoid**(**theta**'\***X**)));**

g **=** X**\*(**sigmoid**(**theta**'\***X**)** **-** y**)';**

**3.3 DISCUSSION**

The task is to teach the machine to identify numbers given a set of data thus taken from the MNIST dataset which has the images of either “0” or “1”. These are represented in a 28 x 28 grid of pixels thus gives a vector of 784 elements. The group imports the ex1\_load\_mnist.m containing the training and testing data. The group is to fill in the logistic\_regression.m function to return the values. Based in the code of the logistic regression function the group used the equation of logistic regression to get the gradient “g” and the objective value “f”. The results gave an iteration loop of 32 that took approximately 9 seconds gave a result of 100 % in training and test accuracy

**4 CONCLUSION**

In this laboratory activity we achieved the objective of of this experiment, to have a value of theta that can make the probability increase or decrease depending on what class you want it to belong, either class 1 for high probability or class 0 for low probability. With the use of these codes:

f = -sum(y.\*log(sigmoid(theta'\*X)) + (1-y).\*log(1 - sigmoid(theta'\*X)));

g = X\*(sigmoid(theta'\*X) - y)';

we were able to get a 100% training and test accuracy.

**REFERENCES**

[1] <http://onlinestatbook.com/2/regression/intro.html>

[2] Rheumatology (2013). [Mark Lunt. - link](http://rheumatology.oxfordjournals.org/search?author1=Mark+Lunt&sortspec=date&submit=Submit)<http://rheumatology.oxfordjournals.org/content/early/2013/04/16/rheumatology.ket146.long>