**SE4060 – Machine Learning**

**Lab 3 – Linear Regression with one variable**

**Setting up the lab environment**

1. If you want to install the software on your own computer, you can download and install Octave and Anaconda (these should already be there in the lab PCs).

<https://www.gnu.org/software/octave/download.html>

<https://www.anaconda.com/download/>

**Exercise for Linear regression with one variable**

1. Open the lab1 directory. There you will find some octave scripts.

ex1.m – Code to run univariate linear regression. This calls the appropriate Octave functions in other files to do univariate linear regression on the ex1data1.txt dataset.

ex2.m - Code to run multivariate linear regression. This calls the appropriate Octave functions in other files to do univariate linear regression on the ex1data2.txt dataset.

1. Open Octave and then make the lab1 directory the current working directory. You can open octave by typing Ctrl + Alt + T (open the terminal) and typing octave.
2. Open the ex1.m file. The file calls the other files but there is some missing code in the other files that you need to fill. If you run ex1.m now, it will run the plotData function in the plotData.m file and display the data and then just exit after giving some warnings. You have to go to the Common window tab and press enter to proceed with the execution and see the results. The data contains the profit details of a company based on the population of each city they operate.
3. Paste the following code under the “Your code here” section in ComputeCost.m file. This defines the cost function for Linear regression.

J = 0;

for i = 1:m

J = J + (theta' \* X(i,:)' - y(i))^2;

end

J = J/(2\*m);

1. Paste the following code under the “Your code here” section in GradientDescent.m file. This defines the gradient descent algorithm for one variable.

temp0 = 0;

temp1 = 0;

for i = 1:m

temp0 = temp0 + (theta' \* X(i,:)' - y(i));

temp1 = temp1 + (theta' \* X(i,:)' - y(i)) \* X(i,2);

end

theta(1) = theta(1) - (alpha/m) \* temp0;

theta(2) = theta(2) - (alpha/m) \* temp1;

% ============================================================

% Save the cost J in every iteration

J\_history(iter) = computeCost(X, y, theta);

1. Now if you run ex1.m you should see the results. You have to go to the Common window tab and press enter to proceed with the execution and see the results. Go through the code and see whether you understand what is happening here. You may have to compare the given code with the lecture note.

The graphs that are generated are just there to visualize how Gradient descent works. Figure 2 shows the cost function plot, which is a convex function. Figure 3 shows the same as a contour plot, with the point where the Gradient descent algorithm has identified as the minimum.

**Running Linear Regression on Jupyter notebook**

Jupyter notebook is a Python development environment that has configured with the necessary libraries to run Machine learning algorithms.

1. Go to the following link and download the Jupyter notebook for the given example.

<http://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html#sphx-glr-auto-examples-linear-model-plot-ols-py>

1. Open Anaconda navigator by opening the terminal and typing ‘anaconda-navigator’.
2. Launch the Jupyter notebook.
3. It will show the directories of the current directory and the running notebooks.
4. Upload the downloaded notebook using the upload button.
5. Once uploaded, the uploaded notebook should appear on the list given.
6. Click on the notebook to run it. Now Jupyter will run the given code and display the results. Try to read the code and understand what’s going on. You may have to refer sci-kit learn documentation to understand what each function is doing. The dataset is a test dataset given in scikit-learn.

<http://scikit-learn.org/stable/modules/classes.html>

1. If you select ‘Cell’->’Run All’, it should run the python code in the cell and generate the results again (it might take some time depending on the speed of your PC).
2. Download the resulting file using file->Download as html.

**Submission:**

Upload the modified Octave code and the html file exported by Jupyter notebook as a single zip file to the courseweb link. The file name should be your registration number.