SE4060 – Machine Learning

Lab 3 – Logistic Regression

Octave Exercise for Logistic Regression

1. Open the lab3 directory. There you will find some octave scripts.

   ex2.m - Code to run logistic regression. This calls the appropriate Octave functions in other files to do logistic regression on the ex2data1.txt dataset. It contains student admission details based on the marks of two exams the student has taken. So there are two input features and you have to predict the admission status, which is a binary value. So, it's a binary classification problem.

2. Open the ex2.m file. You should get some errors. Now we'll try to modify the code to do logistic regression.

3. Add the following code under the "Your code here" section in the costFunction.m script. You can (ignore the grad calculation, which is just computing the gradient value of the cost function for the initial theta values).

```
temp1 = -1 * (y .* log(sigmoid(X * theta)));

temp2 = (1 - y) .* log(1 - sigmoid(X * theta));


J = sum(temp1 - temp2) / m;
```

4. Add the following code to the sigmoid function in sigmoid.m script.

```
denominator = 1 + exp(-1 * z);

g = 1 ./ denominator;
```

5. Add the following code to the predict.m script to predict the admission status of an unknown student.

```
result = sigmoid(X * theta);

p = round(result);
```

6. Now run ex2.m to see how the training is done and to see the prediction on the unknown student. Refer the lecture note and see whether you can understand the code.

Logistic Regression with regularization

1. The ex2_reg.m file runs a linear regression algorithm with regularization. It takes the input test data of two tests done on a microchip and tries to predict whether it's faulty or not. The mapFeature.m script makes the input feature set a polynomial feature set.

2. Open the costFunctionReg.m and add the following code to define the new cost function with the regularization parameter.

```
temp1 = -1 * (y .* log(sigmoid(X * theta)));

temp2 = (1 - y) .* log(1 - sigmoid(X * theta));


thetaT = theta;

thetaT(1) = 0;

correction = sum(thetaT .^ 2) * (lambda / (2 * m));


J = sum(temp1 - temp2) / m + correction;
```

3. The value of Lambda is currently set to 1 in ex2.m change it to different values to see that the training accuracy changes. Note that even though the training accuracy may be high when Lambda is lower, it may lead to overfitting.

Multiclass logistic regression

Download the Jupyter notebook for Iris dataset multiclass classifier built on logistic regression and upload to your Jupyter notebook and run it.

http://scikit-learn.org/stable/auto_examples/linear_model/plot_iris_logistic.html#sphx-glr-auto-examples-linear-model-plot-iris-logistic-py

The iris dataset used for this sample is explained at,

http://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.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.