Laboratory 3

Logistic Regression Classification

Due Date: Beginning of Week 5 Lab

Concepts:

- Logistic Regression Classification
- Sigmoid Activation function
- The classification model evaluation

Total Points: 100 points

Objectives:

Logistic regression is one of the most popular supervised classification algorithm. The classification algorithm mostly used for solving binary classification problems. Students will implement the logistic regression, and evaluating performance of the developed model on the test data that is not used in the training.

Files Needed:

- Lab3.mlx
- sigmoid.m
- mySplitData.m
- Skin.csv

Assignment:

Binary classification using logistic regression

In this lab, you will implement logistic regression for the binary classification problem. The data set used is the skin dataset. It is constructed over B, G, R color space. The class labels 'skin' and 'non skin' are generated using skin textures from face images of diversity of age, gender, and race people. The objective of the logistic regression classifier is to predict the class label whether it is 'skin' or 'non skin' given the input feature data.

As we learned in the class, logistic regression predicts the probability y=1 given the input x. To squash the output in the range [0, 1], the sigmoid activation function is used. The gradient descent algorithm is used to find the optimal weights $\boldsymbol{w}$ to minimize the log loss cost function. For the implementation of the binary classifier, you will follow the steps below:

1. Load the data from a file

- The file "skin.csv" contains the R,G, B color values as input features to represent the textures from human face images and the output variable has two classes {"skin" or "non-skin"}. Since the file is a CSV file (comma-separated-values), you can use Matlab ***load( )*** function. In the data file, the first three columns are B,G,R (x1,x2, and x3 features) values and fourth column is of the class labels (decision variable y).

- Implement a Matlab function "mySplitData.m" that splits the data into training and testing sets. The function takes three parameters,' x', 'y', and 'rate' where 'x' is the input feature matrix, ' y' contains the corresponding class labels, and 'rate' is the training sample rate. The return variables are defined as below:

```
function [XTrain XTest yTrain yTest] = mySplitData (x, y, rate )
```

  Complete the provided file "mySplitData.m".

- Add the lines of code for data loading and feature extraction in **Section (1)** in "Lab3.mlx".

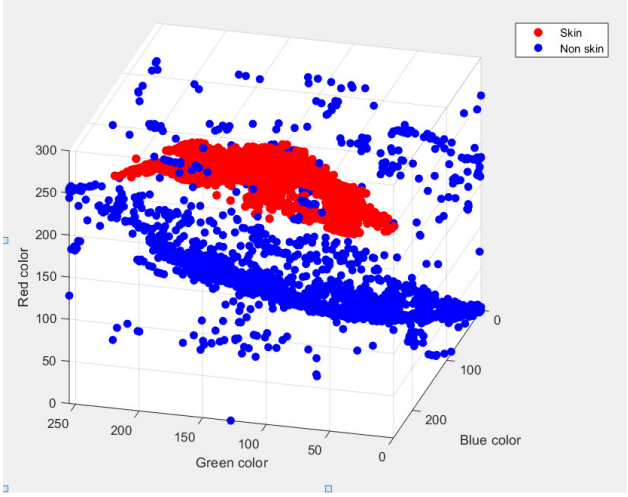
- Note: The training data set will be used to build your logistic regression classifier. For **Section (2)- Section (7)** in "Lab3.mlx", you are assumed to use the training data only. The evaluation of the developed logistic classifier will be done by using the test data set in **Section(8)**.

2. Visualize the data

- Before starting on any prediction model, it is often useful to understand the data by plotting it.
- First, plot the positive samples ('skin') with red dots using the **scatter3()** function. The input to the scatter3() function is the B,G,R feature values. To select the positive samples from the input data, you can use the **find( )** function.

  Example)
  %find indices of positive samples in the variable y
  pos=find(y ==1)

- Using the command "***hold on***" to visible the current plot to the next consecutive plots.
- Then, plot the negative samples ('Non skin') with blue dots using the **scatter3()** function.
- To learn more about the plot command, you can type help **scatter3** at the MATLAB command prompt or to search online for plotting documentation.
- The 3D plot should be similar with the below figure:



- Complete the lines of code in Section 2.

3. Data normalization

- Using the standardization method, normalize the input features.
- Complete the lines of code for data normalization in Section 3 for the given input feature

4. Add Bias term

- The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis.
- Complete the lines of code in Section 4.

5. Find parameters using gradient descent

a) Sigmoid function

The logistic regression hypothesis is defined as a sigmoid function of an input Z where Z is a liner model:

$$h_w(x) = \frac{1}{1-e^{-z}} \text{ , where } Z = \sum_{i=0}^{k} \boldsymbol{w_i x_i}$$

- To implement the hypothesis, you need to define the sigmoid function. **Write the function 'sigmoid' in sigmoid.m so it can be called to calculate the logistic regression classifier. Complete the provided file 'sigmoid.m'**.
- Note: the sigmoid function should work with vectors and matrices for the vectorization.

b) Log loss cost function

- The cost function in logistic regression is the log loss function :

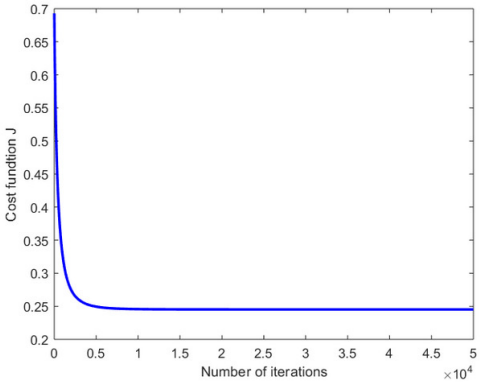
$$J(w)= -[\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y^{(i)}} \cdot \log\left(\boldsymbol{h_w(x^{(i)})}\right) + (\mathbf{1}-\boldsymbol{y^{(i)}}) \cdot \log\left(\mathbf{1}-\boldsymbol{h_w(x^{(i)})}\right)]$$

- You will fit the classifier parameters, w with the training data using the gradient descent method. The objective of the training is to find the parameters, w that minimize the cost function:

$$\underset{w}{\text{Minimize}} \quad J(w,x)$$

- Complete the lines of code to implement the gradient descent method in Section 5.
- At the end of the for loop iteration, you have the final log loss cost and the final parameters, $\boldsymbol{w}$ for the classification model.

6. **Plot the Log Loss cost convergence graph.** Generate a plot that shows how the cost changes over the iterations. The plot of the cost versus the iteration looks like below:



- Complete the lines of code to implement **the Log Loss cost convergence graph** in Section 6.

7. **Calculation of the training accuracy**

   - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
   - Calculate the sigmoid output of the input Z
   - If the sigmoid output >= 0.5, the predicted output is 1
     else (sigmoid output < 0.5) , the predicted output is 0

- Write the lines of code for the calculation of the training accuracy in Section7.

Question 1: what is the accuracy of the developed model on the training data?

8. **The developed classifier evaluation with the test data**

- Normalize the input feature. **You have to use the same mean and the same standard deviation which are calculated with training data** in section Part1-3.
- Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
- Make the predictions for the new testing data using the logistic regression classifier trained in Part1-5.

- **Calculate the accuracy of the developed prediction model on the test data**

  - Calculate the linear model $Z = \sum_{i=0}^{k} w_i x_i$
  - Calculate the sigmoid output of the input Z
  - If the sigmoid output >= 0.5, the predicted output is 1
    else (sigmoid output < 0.5) , the predicted output is 0

- Add the lines of code to evaluate the logistic classifier on the test data set in Section8.

- **Question 2: W**hat is the accuracy of the developed model on the test data set?
- **Question 3:** By comparing both accuracies on the training and the testing sets, do you have any underfitting /overfitting issues?

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (5 pts): contains answers for Question1 –Question3.**
  2) (75 pts): The live script file, " Lab3.mlx", in that all outputs are generated either inline or on the right side.
  3) (10 points) : sigmoid.m
  4) (10points) : mySplitData.m

- **Pay attention to the zip file name convention:**
  Lab3_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab3_Green_Smith.zip