| KING SAUD UNIVERSITY<br>COLLEGE OF COMPUTER AND INFORMATION SCIENCES<br>Computer Science Department | | |
|---|---|---|
| **CSC 462: Machine Learning** | **Project Specifications**<br>Gender Recognition by Voice | **1st Semester 1439-1440** |

## Introduction

Human brain -with the help of ears- has an excellent mechanism to percept voices. Listeners -usually- have the ability to distinguish between persons based on their gender, age, emotional state, loudness, and other factors. Although recognizing a speaker as male or female considered as an easy task; building an automatic voice gender recognition system is challenging. Number of researches are devoted to develop and examine suitable recognition systems. The core component of such systems is the utilized classification technique, such as Support Vector Machine (SVM), Neural Network (NN), and logistic regression.

In this project, you are required to investigate the effectiveness of the logistic regression classifier in gender recognition by voice.

## Dataset

To achieve the required task, you will use Gender Recognition by Voice and Speech Analysis dataset from Kaggle. The dataset consists of 3,168 recorded voice samples, collected from male and female speakers. The voice samples are pre-processed by acoustic analysis in R, with an analyzed frequency range of 0hz-280hz (human vocal range). The following acoustic properties of each voice are measured and included within CSV file:

1. ***meanfreq:*** mean frequency (in kHz)
2. ***sd:*** standard deviation of frequency
3. ***median:*** median frequency (in kHz)
4. ***Q25:*** first quantile (in kHz)
5. ***Q75:*** third quantile (in kHz)
6. ***IQR:*** inter-quantile range (in kHz)
7. ***skew:*** skewness
8. ***kurt:*** kurtosis
9. ***sp.ent:*** spectral entropy
10. ***sfm:*** spectral flatness
11. ***mode:*** mode frequency
12. ***centroid:*** frequency centroid
13. ***peakf:*** peak frequency (frequency with highest energy)
14. ***meanfun:*** average of fundamental frequency measured across acoustic signal
15. ***minfun:*** minimum fundamental frequency measured across acoustic signal
16. ***maxfun:*** maximum fundamental frequency measured across acoustic signal
17. ***meandom:*** average of dominant frequency measured across acoustic signal
18. ***mindom:*** minimum of dominant frequency measured across acoustic signal
19. ***maxdom:*** maximum of dominant frequency measured across acoustic signal
20. ***dfrange:*** range of dominant frequency measured across acoustic signal

21. ***modindx:*** modulation index. Calculated as the accumulated absolute difference between adjacent measurements of fundamental frequencies divided by the frequency range
22. ***label:*** 0 (male) or 1 (female)

To simplify the process for you, the dataset is divided into training set (75% of the whole set) and testing set (25% of the whole set).

## Requirements

In this project you are required to build logistic regression model to predict the gender of a speaker (male or female) based on his/her extracted voice features. In order to obtain that model, you need to implement the following functions:

- **`function x = normalizeFeatures(x)`**

  Normalizes the received features according to this range scaling function: $\frac{x_j^{(i)} - \min(x_j)}{\max(x_j) - \min(x_j)}$, where ***i*** is the example number and ***j*** is the feature number.

- **`function g = sigmoid(z)`**
  Returns the calculated sigmoid function of the given parameter ***z***.

- **`function [cost, grad] = computeLRCost(x, y, theta, lambda)`**
  Returns the regularized ***cost***, and a vector ***grad*** which contains the partial derivative of the cost function with respect to each theta value.

- **`function theta = learnLRTheta(x, y, lambda)`**
  Learns and returns the best ***theta*** parameters for the given ***lambda*** value, where the initial theta values are zeros. Hint: use MATLAB function ***fminunc*** to learn ***theta*** values.

- **`function [theta, lambda] = trainLRModel(x, y, lambda_values)`**
  Trains the logistic regression model and returns its parameters. This function applies 10-fold cross validation to choose the best ***lambda*** value among ***lambda_values***. As a facilitation, this function should plot the curve of training and validation errors (costs) Vs. lambda values; so, you need to calculate them for each lambda value. Recall that, the best lambda is the one with the minimum validation error. After fixing the ***lambda***, you should learn the ***theta*** values that associated with that ***lambda***. Hint: (1) use MATLAB function ***cvpartition*** to partition the training data for cross validation, (2) remember to remove the regularization term when calculating the errors.

- **`function y = predictClass(x, theta, threshold)`**
  Returns the predicted **y** for the given ***x***, ***theta***, and ***threshold.***

- **`function [acc, recall, precision, fScore] = testPerformance(y, y_predicted)`**

Calculates and returns the following measurements for the received actual and predicted labels: accuracy, recall, precision, and f-score. Hint: build the confusion matrix to simplify the calculations.

## Report

After completing your implementation, apply the following experiments and report their results:

- ***Experiment 1: logistic regression without feature normalization***
  In this experiment, you will train your classifier with following lambda values: 0, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10. As a result, you will obtain a model that consists of theta parameters and a specific lambda value. Use your model to predict the labels of the test set and then calculate the performance measures. You should report the following: training and validation errors Vs. lambda values curve, the selected lambda value and how you selected it, and the computed performance measures.

- ***Experiment 2: logistic regression with feature normalization***
  In this experiment, you will repeat experiment 1 but with applying feature normalization before model training. Report the results and compare them with experiment 1 results.

- ***[Bonus – one mark] Experiment 3: compare logistic regression with support vector machine (SVM)***
  In this experiment, you will train SVM model using MATLAB function (no need to implement it). Use the built model to predict the labels of the test set, and then calculate the performance measures and report them. Compare the results of SVM model with the results of logistic regression models that built in experiments 1 and 2.

## Deliverables and Rules

The submission will be through **LMS** and the **deadline** is **2/12/2018**. You should deliver:

- A written report.
- Source code.

You have to read and follow the following rules:

- This project is to be conducted by groups of exactly **four** students.
- The students should implement their code using **MATLAB**.
- Every group member should participate in all parts of the project: designing, programming, and report. Members of the same group may receive different marks according to their participation in the work.
- The submitted code will be evaluated in a demonstration which all the group members should attend.
- Any member of the group who **fails to attend** the demonstration without a proper excuse shall receive **0 in the demo mark**.
- In accordance with the university regulation, **cheating** in the project will be **penalized** by the **mark 0** in the project.