Project Report on Logistic Regression

Introduction

In statistics, logistic regression is a type of probabilistic statistical classification model. Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables, which are usually (but not necessarily) continuous, by using probability scores as the predicted values of the dependent variable.

Logistic regression can be seen as a special case of generalized linear model and thus analogous to linear regression. The model of logistic regression, however, is based on quite different assumptions (about the relationship between dependent and independent variables) from those of linear regression. In particular the key differences of these two models can be seen in the following two features of logistic regression.

- The conditional distribution $p(y \mid x)$ is a Bernoulli distribution rather than a Gaussian distribution, because the dependent variable is binary.
- The linear combination of the inputs $w^T x \in R$ is restricted to [0,1] through the logistic distribution function because logistic regression predicts the probability of the instance being positive.

Logistic Regression, P(Y|X) where Y can take on any of the discrete values {y1,... yK}, then the form of P(Y = yk|X)

for Y = y1,Y = y2,...Y = yK−1 is:

$$P(Y = yk|X) = [\exp(wk0 + \sum_{i=1}^{n} wkiXi)] / [1+\sum_{j=1}^{K-1} \exp(wj0 + \sum_{i=1}^{n} wjiXi)]$$

When Y = yK,

$$P(Y = yK|X) = 1/ [1+\sum_{j=1}^{K-1} \exp(wj0 + \sum_{i=1}^{n} wjiXi)]$$

Here wji denotes the weight associated with the jth class Y = yj and with input Xi.

This function is concave and can be optimized using the gradient ascent/descent algorithm.

$$W\_new \leftarrow W\_old + \eta [\sum (i\ X\ l) (\delta(Y= yj) - P\hat{}(Y=yj|X ,W)) - \lambda * wji]$$

Implementation

Initially, we have our input data as .wav files. So we need to extract sample data and frames/sec of .wav file and generate fft values and store those fft values in the corresponding array or list.

1. Reading sample data values from the .wav files.
   We can do this, using audioread() function in Matlab. Audioread() function takes path of the file as input, reads the file and generates sample rate along with sample rate.

For ex:

```
[data,rate]=audioread('data\genre\classical\classical.0000.wav');
```

2. Now, we need to generate Fast Fourier Transfrom (fft) values and Mel Frequency Cepstral Coefficients (mfcc) values using the sample data. We can calculate both fft values and mfcc values using the pre-defined functions in Matlab.

2.1 A) Implementation using FFT Values:
We can generate fft values using the fft() function in Matlab.

Fft() function accepts the sample data values as input and returns them as fast fourier transform(fft) values. For the sake of complexity, and speed, I fixed the number of FFT components to the first 1000.

```
FFT_Val(i,2:1001)=fft(y(1:1000));
```

B) Implementation using MFCC Values:
We can generate MFCC values using the mfcc() function in Matlab.

Mfcc() function accepts the sample data, sample rate generated by audioread() function and Tw, Ts, @hamming, R, M, N, C, L as input parameters and returns mfcc values.

```
[cc, fbe, frames] = mfcc(y,fs,Tw,Ts,alpha,@hamming,R,M,C,L);
```

2.1.1 We need to normalize the generated fft values to avoid overflow of the data.

We normalize the songs using the maximum fft value in that corresponding song.

```
Normalized_FFT(i,:) = FFT_Val(i,:)./max(FFT_Val(i,:));
```

Where max() would be equal to the max at that position on the vector over all the songs. This makes it so the largest point along any vector is 1.

2.1.2 Now, initialize a weight matrix of size(#of songs * #of attributes+1). We added to +1 for weight bias Wi0. Initialize the initial weights to zero because the training of the training data set starts from 0 and the weights there by adjust themselves accordingly depending upon the error rate.

We assign a predicted class to a song by multiplying the Weights with FFT values. But here Weight matrix size = 6 * 10001
FFT values matrix size = 600 * 1000
We need to add additional column with column_id = 1 with all 1's. And then multiply with Weights and transposed FFT values matrix.

```
Probability = exp(Weight * Normalized_FFT);
```

Now, find the maximum value per each song and the index with the maximum value is the predicted genre by our model.

2.1.3 We compare our obtained genres of the song with the original genres and calculate error rate to train the weights. Weight updating is done using Gradient Descent algorithm.
Gradient Descent algorithm uses
Lambda = Penalty parameter = 0.001
Epochs = 100
Eta = Learning parameter = 0.01
Eta new = Eta old / (1+current_epoch/Epochs)
Gradient Descent is:

```
Temp = (Delta-Probability) * transpose(Normalized_FFT);
Temp1 = Eta * (Temp - (Lambda * Weight));
Weight_new = Weight + Temp1;
```

where Delta = True labels of the songs
Probability = Weight * Normalized_FFT

2.1.4 The final updated weights are multiplied with testing data fft values and the genre with maximum probability is predicted as genre of that particular song.

2.1.5 The above process is used with 10 fold cross validation where each fold is used as testing data, which helps to increase the accuracy.

2.1.6 Accordingly Accuracy and Confusion Matrix is calculated for each and every fold.

Similarly, the process is repeated for Mfcc values too.

Questions:

A) Use the 1000 first FFT components as features.

Ans:

Confusion Matrix:

accuracy_test_fft | cc | accuracy_test_fft | confusion_fft

10x1 double

	1	2
1	0.2333	
2	0.2333	
3	0.2333	
4	0.2333	
5	0.2500	
6	0.1667	
7	0.3000	
8	0.3000	
9	0.2667	
10	0.3667	
11		

6x6 double

	1	2	3	4	5	6	7
1	83	2	13	0	0	2	
2	54	9	30	3	0	4	
3	59	5	32	3	0	1	
4	42	13	23	13	1	8	
5	45	11	16	3	12	13	
6	39	17	33	4	1	6	
7							
8							

Accuracy range is 16.67 to 36.67 for all the folds. Accuracy fluctuates as epoch count varies.

Bias: Bias influences learning rate parameter. Bias is implemented to avoid NAN's after 2 epochs.

B) Using your knowledge from the previous homework, design a method to rank the FFT components and select the best 20 per genre. Use the selected 120 features to classify the data set. Explain how this step affects your accuracy.

Ans:

The songs are ranked according to the standard deviation values generated for each genre.

Standard deviation values are generated for each genre separately and sorted by saving the index. Now the songs from 41 to 60 which are in middle of the 100 songs / genre are selected as the best 20 songs per genre.

Now, we have 6 genres and so we get 120 songs which are best among 6 genres.

Among these 120 songs, 108 songs are used for training and 12 songs for testing using 10 fold cross validation.

Confusion Matrix:

accuracy_test_best — 10x1 double

	1	2
1	0	
2	0.2500	
3	0.2500	
4	0.5833	
5	0.5000	
6	0.3333	
7	0.1667	
8	0.5833	
9	0.5000	
10	0.2500	
11		

accuracy_test_best | confusion_best — 6x6 double

	1	2	3	4	5	6	7
1	10	3	1	5	1	0	
2	12	3	0	2	3	0	
3	8	3	4	2	2	1	
4	10	0	0	9	1	0	
5	8	1	1	4	6	0	
6	5	4	0	2	0	9	
7							
8							

Accuracy range is 0 to 58.33 for all the folds. Accuracy fluctuates as epoch count varies.

Bias: Bias influences learning rate parameter. Bias is implemented to avoid NAN's after 2 epochs.

C) Extract the MFCC and use them as your data features.
Ans:

Confusion Matrix:

accuracy_test_mfcc — 10x1 double

	1	2
1	0.4167	
2	0.2833	
3	0.1333	
4	0.5833	
5	0.4833	
6	0.4333	
7	0.3833	
8	0.4500	
9	0.4333	
10	0.3333	
11		

accuracy_test_mfcc | confusion_mfcc — 6x6 double

	1	2	3	4	5	6	7
1	97	1	0	1	0	1	
2	61	7	24	3	4	1	
3	54	1	42	0	3	0	
4	3	5	8	65	16	3	
5	45	3	9	2	41	0	
6	42	7	30	7	10	4	
7							
8							

Accuracy range is from 13.33 to 58.33. Accuracy fluctuates as epoch count varies.
Bias: Bias influences learning rate parameter. Bias is implemented to avoid NAN's after 2 epochs.

Improving Classification:

- As the dataset size increases we can increase the accuracy.
- If the weights are not made to zero after each and every fold, the overall accuracy can be increased.
- Same with the learning rate parameter.
- By selecting a specific epoch value, accuracy can be increased.