Emotion Detection from Speech

Niraj Dev Pandey, Dhawal Thakar, Utkarsh Gupta, KD & Sharat |

Mentor: Omnisys Solution

Abstract - In this project, simulated English emotional speech database has been borrowed from a Toronto Speech Data Set 2010. Emotional classification is attempted on the corpus using spectral features. The spectral features used are Mel Frequency Cepstral Coefficients(MFCCs) and Sub band Spectral Coefficients (SSCs) The feature vector in use has 273 features, obtained from 7 individual features of 13 banks of MFCCs and 26 SSCs computed over the dataset. This dataset is trained on multiple classifiers, wherein with each classifier, related learning and error penalty rate parameters have been varied to find the best set of classifiers. The lists of accuracies, precisions, and f1-scores are compared. Our methods show that Support Vector Machines with Radial Basis Function kernel provides the best accuracy rates, with accuracy for male dataset being ……. and for female dataset being …… The results are on par with the results obtained by training on full Toronto speech dataset.

Keywords - Spectral Features, MFCCs, SSCs, SVM, RBF, DeepANN, MLP, Adaboost

I. Introduction

Understanding human-speech has been an integral and fascinating part of AI as well as Digital Speech Processing for a long time. Emotion recognition is also an integral component of understanding speech. Same phrases can convey different emotions when spoken differently. In our project we explore different classifiers to categorize the spoken utterance discretely into 8 states: anger, fear, disgust, happiness, surprise, neutral, sadness and sarcastic, to obtain a system capable of recognizing emotions in speech utterances, with reasonable accuracy. Also, since most people in world are familiar with the spoken English, we chose a English Emotional speech corpus for our testing. However, the system we have built should be able to train nicely, and give reasonable performance on Emotion-Corpus of of speech in any language.

II. Dataset & Platform

The dataset we are using is a subset of Toronto speech data set: Simulated Emotion English Speech Corpus. These stimuli were modeled on the Northwestern University Auditory Test No. 6 (NU-6; Tillman & Carhart, 1966). A set of 200 target words were spoken in the carrier phrase "Say the word _____' by two actresses (aged 26 and 64 years) and recordings were made of the set portraying each of seven emotions (anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral). There are 2800 stimuli in total. Two actresses were recruited from the Toronto area. Both actresses speak English as their first language, are university educated, and have musical training. Audiometric testing indicated that both actresses have thresholds within the normal range.

The platform we are using is Python, and the following opensource libraries were borrowed:

1. Scikit Learn
2. Pybrain

Research project submission – December 2017
Author – Niraj Dev Pandey and Team (pandey@uni-hildesheim.de)
Guidance & Support – Prof. Lars Schmidt-Theime and ISMLL Team

3. Python Speech Features, MFCC and SSC, James Leon

III. Classification Evaluation Method:

2 We want our system to be independent of the verbal content of the utterance itself, as the semantic content of the neutral speech used has no distinguishing features. For this, we are using an 'unrandomized' K-Fold Cross Validation. As per our dataset, for one speaker, we have approx. 100 words, each spoken in 8 emotions in 10 sessions, so our dataset is divided into 15 folds, each fold containing 100×10 = 800 utterances corresponding to one word. Then the classifier is simply tested for each fold (after training it on the remaining 14 folds), and then the average of accuracies of all the 15 cases is evaluated and maximized.

IV. MFCC and SSC as Features:

In DSP (Digital Signal Processing), a cepstrum is defined as the Inverse Fourier Transform (IFT) of the log of the Power-Spectrum of a signal. 'Power Cepstrum', defined by:

$$PowerCepstrum = \left\| F^{-1}\left\{ log(\|F\{f(t)\}\|^2) \right\} \right\|^2$$

A. *Mel Frequency Cepstrum:*

"The difference between the cepstrum and the Mel-Frequency Cepstrum is that in the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum" - Wikipedia.

B. *Mel Frequency Cepstrum Coefficients (MFCC):*

MFCCs are the coefficients that can characterise an MFC. Since we don't want our system to be dependent on the 'words' which are spoken, we use these cepstral features. For a given discrete time finite length signal window, MFCC is calculated by the tutorial provided on -www.practicalcryptography.com

1. Frame the signal into short frames.
2. For each frame calculate the periodogram estimate of the power spectrum.
3. Apply the mel filterbank to the power spectra, sum the energy in each filter.
4. Take the logarithm of all filterbank energies.
5. Take the DCT of the log filterbank energies.

In our case we split our audio files into smaller 25ms long 'windows', and then we calculate the MFCC using a python library. We are using 512 windows for calculating the FFT, 26 frequency subbands and then finally obtain 13 coefficients for each frame.

C. *Spectral Subband Centroid (SSC):*

For each of the Mel Frequency sub-band, SSC coefficient is calculated as:

$$SSC(i) = \frac{\sum_{k=1}^{n} f_i(k)\, x_i(k)}{\sum_{k=1}^{n} x_i(k)}, i = 1 to 26$$

Where, $f_i(k)$ is the kth frequency belonging to the ith bank, and $f_i(k)$ is its corresponding power amplitude, which is acting as a weight here. SSC features are used alongside MFCC because MFCC features aren't robust to white noise or to the variation of overall intensity of the spoken sound. So, we tried out this feature in hope of increasing the overall accuracy, which it did, and hence we have concatenated this feature as well.

D. Removing the effect of 'Different audio lengths':

We extracted the MFCC and SSC, but then we were facing two problems:

1. Average length of the files was around 7 seconds, window length being 25ms, so we had around $7 \times (13+26) \times 1000/25 = 10920$ features per utterance!

2. Our audio files are of different lengths so extracted features will be of different lengths as well (since no. of features d = no. of frames * 39)

Initially we concatenated all features of all frames into a single vector, and patched extra zeroes to make lengths equal. It was obviously giving very bad results. We found that people tackle this problem by representing each of the coefficient of MFCC and each subband centroid by its mean and variance over all the frames, thus removing the different-audio-length effect, and reducing the feature length to 39*2 = 78. We added many other things as well like maxima, minima, etc and after various combinations our final representation of each mel coefficient and SSC included mean, variance, maximum, minimum, variance of derivative over all the frames and mean of first half frames, and mean of second half frames as well. Hence, finally we have 7*(13+26) = 273 features per utterance.

V. Classification:

Under Construction