Speech Classification Using AdaBoost

Abhishek Vasu av1912
Priyanka Mohandas pm2641

December 23, 2016

Abstract

The paper delineates the basic steps involved in speech classification using supervised learning. We implement a simple speech recognition program in Python using Mel Frequency Cepstral Coefficients for feature extraction and AdaBoost for boosting.

1 Introduction

Speech classification involves the process of identifying the spoken words and translating them into text. This is still a challenging task due to the possibilities of speech signals.

In this project we create a speech recognition system in Python language. We use Mel Frequency Cepstral Coefficients (MFCC) for feature extraction and AdaBoost for learning. It is a supervised learning approach of binary classification.

2 Theory

2.1 Feature Extraction

Every speech signal has a unique set of characteristics which enable us to differentiate between two similar or different speech signals. The first step for speech recognition is to find such characteristics that are useful for classification. These characteristics are called features. Classification is done on these features than on the speech signals themselves. The feature extraction stage seeks to provide a compact representation of the speech waveform. There are various ways of extracting features.

1.Linear Predictive Coding Analysis (LPC)
LPC is one of the good analysis techniques for extracting features and hence encoding the speech at low bit rate. LPC has capability for speech compression, synthesis and as well as identification .LPC is spectral estimation technique because it provides an estimate of the poles of the vocal tract transfer function.[1]

2.Linear Predictive Cepstral Coefficients (LPCC)
This technique is just an extension to the above mentioned LPC technique. When linear predictive coefficient is represented in cepstrum domain then the

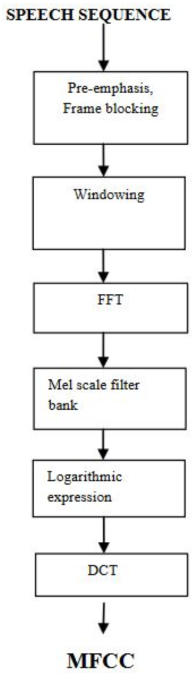

Figure 1: MFCC

obtained coefficients are linear predictive cepstral coefficients. Cepstrum is obtained by taking inverse DFT of logarithm of the magnitude of the DFT of the speech signal.They are more robust and reliable then LPC.[1]

3.Mel-Frequency Cepstral Coefficients (MFCC) The cepstrum coefficient is the result of a cosine transformation of the real logarithm of the short time energy spectrum expressed on a Mel-frequency scale. This is a more robust, reliable feature set for speech recognition then the LPC coefficients.The sensitivity of the low order cepstrum coefficient to overall spectral slope, and the sensitivity of the high-order cepstrum coefficient to noise, has made it a standard technique. It weights the cepstrum coefficient by a tapered window so as to minimize these sensitivities, frame and these are used as the feature vector. In MFCC's,the main advantage is that it uses Mel frequency scaling which is very approximate to the human auditory system.[1] Figure 1 shows the basic steps involved in computing the MFCCs.

It is found that MFCC used in Automatic speech Recognition system provide 80 percentage accuracy where as LPCC used in Automatic Speech Recognition provide 60 percentage accuracy.Results and calculations show that MFCC algorithm provides better result in comparison with LPCC algorithm.[4]

2.2 AdaBoost

Boosting is a voting method by weighted weak classifier and AdaBoost ("Adaptive Boosting") is one of method of Boosting (Y. Freund & et al., 1997).[2]The

Parameter	Description
signal	the audio signal from which to compute features. Should be an N*1 array
samplerate	the samplerate of the signal we are working with.
winlen	the length of the analysis window in seconds. Default is 0.025s (25 milliseconds)
winstep	the step between successive windows in seconds. Default is 0.01s (10 milliseconds)
numcep	the number of cepstrum to return, default 13
nfilt	the number of filters in the filterbank, default 26.
nfft	the FFT size. Default is 512
lowfreq	lowest band edge of mel filters. In Hz, default is 0
highfreq	highest band edge of mel filters. In Hz, default is samplerate/2
preemph	apply preemphasis filter with preemph as coefficient. 0 is no filter. Default is 0.97
ceplifter	apply a lifter to final cepstral coefficients. 0 is no lifter. Default is 22
appendEnergy	if this is true, the zeroth cepstral coefficient is replaced with the log of the total frame energy.
returns	A numpy array of size (NUMFRAMES by numcep) containing features. Each row holds 1 feature vector.

Figure 2: MFCC Parameters

Boosting decides the weak classifiers and their weights based on the minimizing of loss function in a two-class problem. It can be used in conjunction with many other types of learning algorithms to improve their performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. The general steps involved are as given below.

3 Implementation

We implemented the speech classification using the Python libraries python_speech_features for extracting MFCC and scikit-learn for the boosting.
python_speech_features is a Python library provides common speech features for ASR including MFCCs. The Python code is as follows

mfcc (signal, samplerate=16000, winlen=0.025, winstep=0.01, numcep=13, nfilt=26, nfft=512, lowfreq=0, highfreq=None, preemph=0.97, ceplifter=22, appendEnergy=True)
Figure 2 gives a list of parameters and their default values.

scikit-learn is a machine learning library that provides the AdaBoost ensemble for the boosting. It can be implemented in Python as follows.

sklearn.ensemble.AdaBoostClassifier(base_estimator=None, n_estimators=50, learning_rate=1.0, algorithm='SAMME.R', random_state=None)
Table 1 gives the list of parameters and their default values.

The audio data accumulated for the purpose of this project was divided into two sets; a training set and a testing set. The training set was used to train the estimators for arriving at a good hypothesis. The testing set was given as the input to the trained estimators and the accuracy or score of the classification was measured. The project implements the classification in real time using the Python library for audio streaming, PyAudio. The minimum accuracy measured in this implementation is 0.75 and the maximum is 0.91667.

Table 1:

Parameters	Description
base_estimato	The base estimator from which the boosted ensemble is built.Default is the Decision Tree Classifier
n_estimators	The maximum number of estimators at which boosting is terminated. In case of perfect fit, the learning procedure is stopped early.
learning_rate	Learning rate shrinks the contribution of each classifier by learning_rate. There is a trade-off between learning_rate and n_estimators.
algorithm	If 'SAMME.R' then use the SAMME.R real boosting algorithm. base_estimator must support calculation of class probabilities. If 'SAMME' then use the SAMME discrete boosting algorithm. The SAMME.R algorithm typically converges faster than SAMME, achieving a lower test error with fewer boosting iterations.
random_state	If int, random_state is the seed used by the random number generator; If RandomState instance, random_state is the random number generator; If None, the random number generator is the RandomState instance used by np.random.

References

[1] Kaur, Gagnapreet; Singh, Dheerendra; Kaur,Gagnadeep, "A Survey on Speech Recognition Algorithms",International Journal of Emerging Research in Management & Technology, vol. 4, 2015

[2] T. Takiguchi, N. Miyake, H. Matsuda and Y. Ariki (2 007). Voice and Noise Detection with AdaBoost, Robust Speech Recognition and Understanding, Michael Grimm and Kristian Kroschel (Ed.), ISBN:978-3-902613-08-0, InTech.

[3] Gulzar,Taabish; Singh,Anand; Sharma,Sandeep, "Comparative Analysis of LPCC, MFCC and BFCC for the Recognition of Hindi Words using Artificial Neural Networks", International Journal of Computer Applications (0975 –8887), vol.12, 2014

[4] Rana, Mukesh; Saloni,Miglani, "Performance Analysis of MFCC and LPCC Techniques in Automatic Speech Recognition", International Journal Of Engineering And Computer Science ISSN:2319-7242, vol. 3, 2014

[5] Zhang,Yan, "Speech Recognition Using Deep Learning Algorithms "