Supervised Machine Learning Model for Accent Recognition in English Speech using Sequential MFCC Features

Dweepa Honnavalli[1*] and Shylaja S S[2]

Dept. of Computer Science, PES University, Bangalore, India
dweepa.prasad@gmail.com
shylaja.sharath@pes.edu

Abstract. Human-machine interfaces are rapidly evolving. They are moving from the traditional methods of input like keyboard and mouse to modern methods like gestures and voice. It is imperative to improve voice recognition and response since there is a growing market of technologies, world-wide, that use this interface. Majority of English speakers around the world have accents which are not exposed to speech recognition systems on a greater scale. In order to bridge the comprehension gap between these systems and the users, the systems need to be tuned according to the accent of the user. Accent classification is an important feature that can be used to increase the accuracy of comprehension of speech recognition systems. This paper recognizes Indian and American English speakers and distinguishes them based on their accents by constructing sequential MFCC features from the frames of the audio sample, oversampling the under-represented data and employing supervised learning techniques. The accuracies of these techniques reach a maximum of 95% with an average of 76%. Neural Networks emerge as the top classifier and perform the best in terms of evaluation metrics. The results gleaned indicate that concatenating MFCC features sequentially and applying an apposite supervised learning technique on the data provide a good solution to the problem of detecting and classifying accents.

Keywords: Accents · Speech · MFCC · Supervised Machine Learning.

1 Introduction

Voice interfaced gadgets are the new frontier of virtual assistants. Voice controlled digital assistants like Apple's Siri, Google Assistant and Amazon's Alexa have been integrated into smartphones, smart speakers and computers. Surveys have indicated that almost 50% of smart gadget owners in the US use voice-enabled virtual assistants [1]. They save hours every day by helping users multitask and managing their activities and are the latest assistive technology for the visually impaired [2]. Virtual assistants are integrated into almost every smartphone and standalone smart speaker.

People all over the world have access to this technology but this technology is not helping everyone the way it should. 1.5 billion people around the world speak English either as a native language or a foreign language. Seventy six percent of

this English-speaking population have accents that virtual assistants do not comprehend. According to a recent survey by Accenture, 1 in 3 online consumers in China, India, the US, Brazil and Mexico will own a standalone digital voice assistant by the end of 2018 [3]. This means 950 million consumers based in India, Mexico, Brazil and China will use what is tuned to the accents of 108 million US consumers.

Automatic speech recognition systems (ASR) seem to work better with American speech rather than accented speech mainly because of the lack of inclusive training data [4]. The problem lies in the fact that there are so many different languages with many dialects and accents. Consider the English language, there are close to 100 accents and dialects and it is not feasible to collect enough annotated data for all accents and dialects. This limits the ASR's capability. With the explosion of data, ASRs have become more inclusive but there is always scope for improvement.

With the rise in the use of machine learning to solve real-world problems, detecting accents find application in many different domains. One of its use cases is in crime investigation. Evidence usually involves usable audio clips of 3-4 seconds. It is imperative to be able to detect the accent in these short, distorted clips to be able to recognize the speaker as well as the speech. If accents can be classified accurately with such short audio clips, investigators can gain insight into the identity of the criminal by identifying the criminal's ethnicity.

Classifying accents is a step toward more intelligent virtual assistants and expanded user-base and usability. This paper seeks to classify accents by concatenating sequential MFCC features which may aid in the ASR's comprehension of accented speech.

2 Background and Related Work

Automatic speech recognition systems (ASR) are ubiquitous and there is a pressing need for them to cover accented English as they form a large part of the population. To help with the ASR revolution, accents need to be understood and incorporated. There is a lot of active research on accented English in speech recognition and efforts to improve automatic speech recognition systems (ASRs) and make them more inclusive.

Research in this area has taken many paths depending on various points of impact such as the audio clip size, the type of accents the speakers have, how adept they are in the English language, etc. Albert Chu et. al, provided a comparative analysis on machine learning techniques in accent classification using self-produced 20-second audio clips and extracting various features to determine their influence. Their paper provides insights on the use of SVM and varying the number of features for feature description using PCA [5].

Each accent has a unique intonation that arises from the root language it is based on. ASRs can be tuned to comprehend accented speech better if they know what to look out for. Liu Wai Kat et. al presented a fast accent classification approach using phenome class models. The paper found that detecting accents and transforming the native accent pronunciation dictionary to that of the accented speech reduces the error rate of ASRs by 13.5% [6]. Accent classification can be used to switch to a more tuned ASR for better comprehension.

Information from audio signals can be extracted in many ways- by sampling, windowing, expressing the signal in the frequency domain or extracting perceptual features. The most common features extracted are Linear predictive codes, Perceptual Linear Predictions (PLP) and Mel frequency cepstral coefficients (MFCC) [7]. As human voice is nonlinear in nature, Linear Predictive Codes do not work as well as PLP and MFCC. PLP and MFCC are derived on the concept of logarithmically spaced filter banks, clubbed with the concept of the human auditory system and hence had a better response than LPC. Studies show that both (PLP and MFCC) parameterization techniques provided almost comparable results. MFCC is chosen as the means of extracting information from the audio samples in this paper.

Hong Tang and Ali A. Ghorbani's paper on accent classification explores classifying accents based on features like word-final Stop Closure Duration, word Duration, intonation and F2-F3 contour. They used Pairwise SVM, DAGSVM and HMM to obtain good accuracy results [8].

This paper takes inspiration from information garnered and proposes and compares methods to classify audio clips of 3-5 seconds containing accented English speech using concatenated MFCC perceptual features.

3 Methodology

3.1 Proposed Method

The proposed method depicted in Fig. 1 involves analysis of audio signals and extracting required features, following which the features are considered and respective frames are concatenated to form the input feature set to the classifier. The results of the classifiers are then validated and compared using different accuracy metrics.

3.2 Dataset

The dataset is a collection of .wav files (3-5 seconds long) from VCTK-corpus. Speakers with American accents and Indian accents are recorded uttering the same content. There are 2301 audio samples in total which are divided into training (80%) and testing (20%).

Fig. 1: Block Diagram of Proposed Method

Table 1: Distribution of the dataset - I

Type pf Accent	Gender	Number of Speakers	Total number of Audio Samples
Indian	Female	1	376
Indian	Male	2	656
American	Female	2	846
American	Male	1	423

Table 2: Distribution of the dataset - II

Type of Accent	Total Number of Audio Samples
Indian	1032
American	1269

3.3 Preprocessing

Table 2 indicates that if the current data is split into training and testing dataset, 'Indian Accent' would be underrepresented. To work around this problem, the dataset is split into testing and training and the training set is oversampled on 'Indian Accent'. After choosing the oversampling algorithm [9], the ratio between the two outcomes in the training data is 1:1.

3.4 Feature Extraction

For each audio file in the dataset, 20 Mel-frequency cepstral coefficients (MFCC) are calculated [10]. Mel frequency cepstrum is the short-term power spectrum of a sound. Feature extraction is implemented using the Python library 'Librosa' [11].

The steps involved in calculating the MFCC cepstral features are:

1. Framing each signal into short frames of equal length with frame length as 2048 samples and hop length as 512 samples.
2. Calculating the periodogram estimate of the power spectrum for each frame.
3. Applying the Mel filterbank to the power spectra and summing the energy in each filter. The formula for converting from frequency to Mel scale is:

$$M(f) = 1125 \ln (1 + (f/700)) \tag{1}$$

4. Taking the logarithm of all filterbank energies.
5. Taking the Discrete Cosine Transform of the log filterbank energies.
6. Selecting the required MFCC coefficients. In this case 20 coefficients are selected.

These coefficients are used because they approximate the human auditory system's response closely. The coefficients of each frame of an audio file are concatenated to form an array of MFCCs.

The MFCC features extracted from an audio sample is outputted in the form of a matrix with 20 coefficients for each frame of the sample, i.e.

$$MFCC = \begin{bmatrix} c_0 f_0 & c_0 f_1 & c_0 f_2 & ... & c_0 f_m \\ ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... \\ c_{19} f_0 & c_{19} f_1 & c_{19} f_2 & ... & c_{19} f_m \end{bmatrix} \tag{2}$$

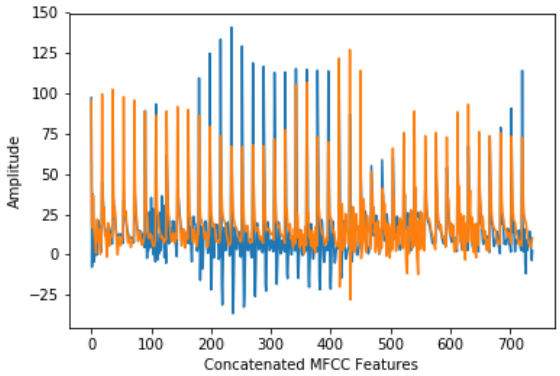


Fig. 2: Distinguishing plot of American and Indian accents

where $c_i f_0 ... c_i f_m$ are the values of coefficient i for frames $1...m$

The above matrix represents the MFCC coefficients for an audio sample with m frames. This 20 x m matrix needs to be transformed into a format that is recognised by the machine learning model.

To be able to contrast the two accents based on their feature sets, MFCC coefficients are sequentially concatenated. This retains enough information required to identify the accent from the feature set.

The features are concatenated and flattened into a one-dimensional array of features for each frame. A sample vector of sequential MFCC features from a single audio sample is shown below.

$$M = \left[c_0 f_0 \ ... \ c_0 f_m \ ... \ ... \ ... \ c_{19} f_0 \ ... \ c_{19} f_m\right] \tag{3}$$

These features are used to train the model to distinguish a particular accent from others. The plot (Fig. 2) of MFCCs extracted from a speaker with American accent versus a speaker with Indian accent, both uttering the same sentence 'Please call Stella', shows there is a variation in speech. This variation helps the model predict the class of accents.

3.5 Supervised Learning

To differentiate between American and Indian accent, the features are extracted and fed the feature data into a machine learning model to get the results.

K nearest neighbor

K- Nearest neighbours algorithm as a classifier calculates five nearest neighbours of each data point in the test and classifies it as either American or Indian based on which type of accent it is situated closest to. It uses Euclidean distance as

the distance metric to calculate the nearest neighbours. It classifies based on the majority vote. Python's Sklearn package has a KNN classifier which does the above taking into consideration 5 of the nearest neighbours.

Support Vector Machine

Support Vector Machines are classification algorithms which find the decision boundaries between the classes. Since the data at hand is of higher dimension, we use a radial basis function kernel to achieve the desired functionality.

Equation for rbf (radial bias function) kernel:

$$K(X, Y) = \exp(-\gamma \frac{||XY||^2}{2\sigma^2}), \gamma > 0 \tag{4}$$

Gaussian Mixture Model

Gaussian Mixture Model is a K means Clustering algorithm that groups the data points into two clusters on for each class: American and Indian. Sklearn's Gaussian mixture model is used to implement this.

Neural Networks

Multi-layer perceptron for binary classification is used with the number of features as the input layer of the neural network and 1 neuron as the output which states if the accent is Indian or American. The optimizer 'Adam' is used to update the weights after each iteration. The neural network was trained over 30 epochs. MLP is implemented using the python library 'Keras' with Google's TensorFlow as the backend.

Logistic Regression

Logistic regression is the suitable for analysis when the dependent variable is dichotomous. Math behind logistic regression:

$$l = \beta_0 + \beta_1 * x_1 \tag{5}$$

Where l is log-odds, logistic regression was opted for the comparative analysis study because of various of reasons. The output is a probability that the given input point belongs to a certain class.

4 Results and Discussion

The results are evaluated based on various metrics like Precision, Recall, Reject rate, accuracy score, area under the Receiver Operating Characteristic (ROC) curve (AUC) and K-fold validation results. From the Test case and K-fold cross-validation results, we observe that Neural networks, K- nearest neighbour and Logistic Regression perform the best in terms of overall accuracy, AUC and K-fold validation.

Model	Mean Validation Score	Std Dev of Validation Score
Neural Networks	0.95	0.02
KNN	0.91	0.06
Logistic Regression	0.95	0.03
SVM	0.36	0.01
GMM	0.39	0.10

Fig. 3: Validation Results

Model	Precision	Recall	f-measure	Reject Rate	Accuracy
Neural Networks	0.96	0.94	0.95	0.97	0.95
KNN	0.9	0.92	0.91	0.9	0.91
Logistic Regression	0.94	0.96	0.95	0.95	0.95
SVM	1.0	0.02	0.04	1.0	0.54
GMM	0.43	1.0	0.60	0.0	0.43

Fig. 4: Test Case Results

4.1 Validation Accuracy

K-fold cross-validation is an evaluation metric in which the sample is partitioned into 'k' partitions. These partitions are of equal sizes and are partitioned at random. Among these k partitions, 1 of them is the validation or testing set and the rest serve as training data. The cross-validation process is repeated k times, such that all of the partitions are used as training data once. In general 'k' remains an unfixed parameter, in our experimentation, k is taken to be as 10. The validation accuracies are tabulated in Fig. 2.

4.2 Test Case Accuracy Metrics

Metrics like precision, recall, reject rate and overall accuracy provide a holistic view on the performance of a model. The results presented apply to the data which is split into training and testing where the former constitutes 80% of the data and the latter 20%.

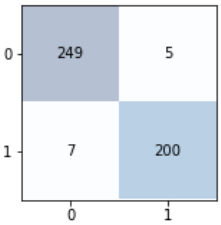
4.3 Analysis of Neural Networks, Logistic regression and K-Nearest Neighbour

Neural Networks

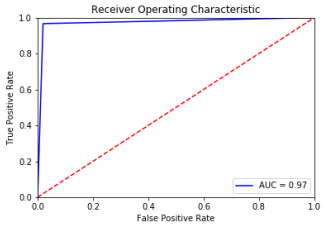
Neural Networks emerge as the top classifier for the job in terms of all the evaluation metrics. It has a high Area under the curve (Fig. 5(b)) value which indicates that it is highly competent in separating the two classes. Confusion matrix describes high precision, recall and reject rate values which indicate the classifier is doing what is required by the problem. However, the only shortcoming is in terms of the time taken for computation which is very high and may prove undesirable for larger inputs, which is tabulated in Table 3.

Table 3: Time taken for computation

Model	Time (sec)
Neural Networks	18.33
KNN	5.22
Logistic regression	0.58



(a) Confusion Matrix



(b) ROC Curve

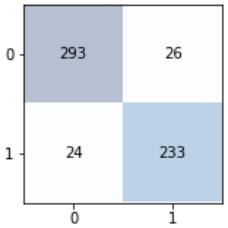
Fig. 5: Neural Networks

K - Nearest Neighbour

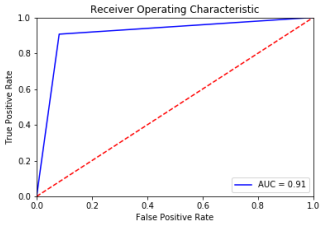
The K- Nearest neighbour classifier does not perform as well as Neural Networks or Logistic Regression classifiers, but it performs well, nonetheless. The results of Neural Network being slightly better than that of KNN can be attributed to the fact that as the number of features in the input data set increases KNN tends to perform worse. It is not far behind Neural Networks and Logistic regression, and it does not take as much time to run as Neural Networks.

Logistic Regression

Logistic Regression and Neural Network classifiers are very close to each other with respect to their evaluation metrics values. Logistic regression, like Neural networks, perform very well in separating the two classes and predicting the classes with good accuracy. It has an edge over Neural networks in the fact that it takes considerably less time to run on big inputs.

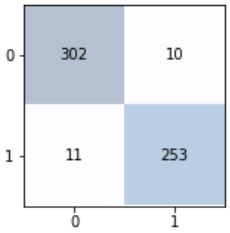


(a) Confusion Matrix

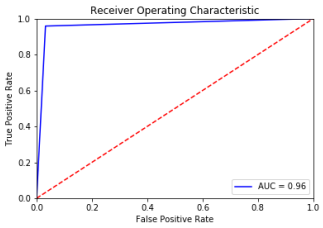


(b) ROC Curve

Fig. 6: K- Nearest Neighbour



(a) Confusion Matrix



(b) ROC Curve

Fig. 7: Logistic Regression

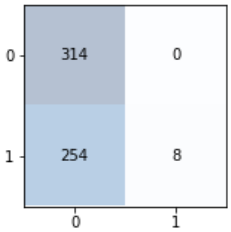
4.4 Analysis of SVM and GMM

Support Vector Machine

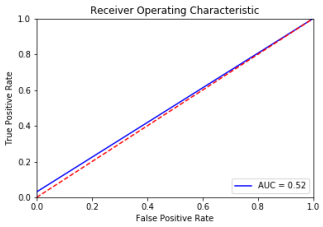
SVM with RBF kernel, in this case, does a poor job of separating the two classes of accents. This can be concluded this by looking at the AUC (Fig. 8(b)) which is 0.52 which is close to a classifier that predicts classes randomly. However misleading conclusions may be made that SVM in indeed a good classifier due to its unusually high precision. On further observation, it is understood that it classifies the majority of the data samples as 'American'. This contributes to the unusually high precision, but starkly low recall and overall accuracy close to 50% which almost relates to a random classifier.

Gaussian Mixture Models

GMM, on the other hand, displays similar characteristics as the SVM model differing in the fact that it classifies the majority of the test case samples as 'Indian' which contributes to the high recall but low precision. The overall accuracy, AUC and K-fold cross-validation results (Fig. 9) lead to inferring that GMM does not work well for the problem at hand.

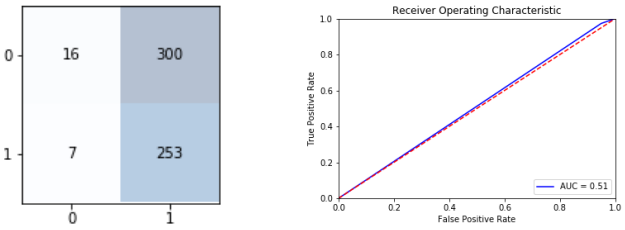


(a) Confusion Matrix



(b) ROC Curve

Fig. 8: Support Vector Machine with rbf kernel



(a) Confusion Matrix (b) ROC Curve

Fig. 9: Gaussian Mixture Models

5 Conclusion and Future work

Classifying Accents based on their acoustic features using machine learning techniques provides good results. By using sequential MFCC features to construct the input, important information is retained that distinguishes the accents in a mathematical perspective. The models are calibrated with this input and validated using various metrics to arrive at a suitable solution to the problem at hand.

Accent classification is a preprocessing step to speech recognition. It can only help in fine-tuning speech recognition systems to detect accented speech better. Future work includes incorporating this into speech recognition systems to improve comprehension of accented English. This project could be extended to various accents and dialects, for example, regional Indian accents like Kannada, Malayali, Tamil and Bengali; and British regional accents like Scottish, Welsh etc. Classification using HMM has not been explored in this paper.

References

1. Center, P.R.: Voice assistants used by 46% of americans, mostly on smartphones (2017), https://pewrsr.ch/2l4wQnr
2. CBC: Smart speakers make life easier for blind users (January 2018), https://www.cbc.ca/radio/spark/380-phantom-traffic-jams-catfishing-scams-and-smart-speakers-1.4482967/smart-speakers-make-life-easier-for-blind-users-1.4482978
3. Accenture: Accenture digital consumer survey (2018), https://www.accenture.com/t20180105T221916Z__w__/us-en/_acnmedia/PDF-69/Accenture-2018-Digital-Consumer-Survey-Findings-Infographic.pdf
4. Ellis, P.: Why virtual assistants can't understand accents (August 2017), https://www.huffingtonpost.co.uk/philip-ellis/is-siri-racist-why-virtua_b_11423538.html?guccounter=2, [Online]
5. Chu, A., Lai, P., Le, D.: (June 2017)
6. Kat, L.W., Fung, P.: Fast accent identification and accented speech recognition. In: 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258). vol. 1, pp. 221–224 vol.1 (March 1999). https://doi.org/10.1109/ICASSP.1999.758102

7. Dave, N.: Feature extraction methods lpc, plp and mfcc in speech recognition. International Journal For Advance Research in Engineering And Technology(ISSN 2320-6802) **Volume 1** (07 2013)
8. Tang, H., Ghorbani, A.A.: Accent classification using support vector machine and hidden markov model. In: Canadian Conference on AI (2003)
9. Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research **18**(17), 1–5 (2017), `http://jmlr.org/papers/v18/16-365.html`
10. Mel frequency cepstral coefficients, `https://bit.ly/1mDpzDu`
11. Librosa : Audio and music processing in python (August 2018). https://doi.org/https://zenodo.org/record/1342708.XDG9LS2B01I