Accent Finder: Classification of user accents

Parijat Parimal
Computer Science Department,
Courant Institute of Mathematical Sciences,
New York University
NY, USA
pp2206@nyu.edu

Patel Rahul
Computer Science Department,
Courant Institute of Mathematical Sciences,
New York University
NY, USA
rbp320@nyu.edu

***Abstract*—**

With the advent of personal assistants such as Siri, OK Google, Cortana, etc. in the past decade, the domain of speech-to-text and vice versa has become quite active. Although there has been an advancement in the machine learning models used to handle speech data, classification of speech still remains an active thread. We aim to pick this up and go one step further in classifying user speech on the basis of the geography.

Keywords—Accent classification, Machine learning, CNN, SVM, MFCC.

I. Introduction

During the last decade, the growth and applications of personal assistants such as Siri, OK Google, Samsung Voice, and Cortana have greatly influenced the modernization of Natural Language Processing. Scientists and engineers come up with better algorithms by the day to enhance user experience. However, one field which still has a potential to explode in terms of applications is that of 'Accents'. Even for a language as widely used as English, it is difficult to distinguish between multiple accents of the same. As a result work is still under progress when it comes to tuning personal assistants to adapt to the accent of a specific user. As of now, personal assistants need to be manually configured to the accent that the owner of the device dictates.

The objectives of our project include the following:

- To build an NLP application that is able to distinguish between multiple accents of English language.
- Classify users on the basis of their geography using a best guess estimate from their accent.

II. Related Work

Fortunately, due to the nature of evolution of this field, work done previously is available for us to build on. Ma, Fan and Zhou [1] concluded that Gaussian Mixture Model clubbed with Hidden Markov Model would work best for accent classification. Chen, Lee, and Neidert [2] received an accuracy of 57.12% while using SVM on a dataset with Mandarin and German non-native speakers. Upadhyay's [3] approach was a bit different as he incorporated deep belief networks to classify the audio files. Wang et al. [4] noted that when their train set included of male speakers alone, it did not perform well while testing on female speakers' data. Ge, Tan and Ganapathiraju [5] observed that accents were swayed by vowel pronunciations more so that the consonants and hence they worked on vowel extraction.

Taking the best of insights from these researches, we find that working with large datasets is difficult and that generalizing a model is far from reality until we have success with the former.

III. Challenges

The difficulty in accent recognition and speech recognition in general is the problems added with accent specific pronunciation of words. In particular, there are words that have different pronunciations across different accents and on the other hand, there are different words that have the same pronunciation across different accents. For example, words like laugh and class have different pronunciations in US and UK accents. Whereas different words like floor-flaw and flower-flour has same pronunciation in UK accents, but each of these words has unique pronunciation in Indian accent. We can see that if we know the accent of the speaker before performing speech recognition tasks, it could help in solving some of these problems. But the same problem exists with accent recognition. To solve this problem, we need to find a way to extract features in such a way that there is no dependency of individual word pronunciation in finding the accent. That is where Mel-Frequency Cepstral Coefficients (MFCC for short) becomes really useful. In this paper, we would see how MFCC helps in solving our problem.

IV. Mel Frequency cepstral co-efficient

MFCC is a mathematical trick that converts the input spectrum from the audio file into a vocal signature for each small frame of the input. The mean of the vocal signatures for each of these frames is computed which gives the vocal signature of the speaker. This is used to by most AI based home devices to recognize individuals speaking to them and allows them to have custom behaviors for known users. Our intuition is that speakers of similar accents will have similar vocal signatures compared to speakers of different accents. We would then use this to classify the user's accent. Below we discuss what MFCC does and how it gives us what we need.

Since the audio signal is constantly changing, we simplify it by dividing it to small frames of 20-40ms where each frame is now relatively constant with minimum changes. Then power spectrum of each frame is calculated which is inspired by human hearing (more specifically cochlea in our ear). This helps in identifying frequencies in the frame. The spectrum still contains a lot of information that we don't need. The periodogram is then obtained by performing Fourier transformations on the power spectrum.

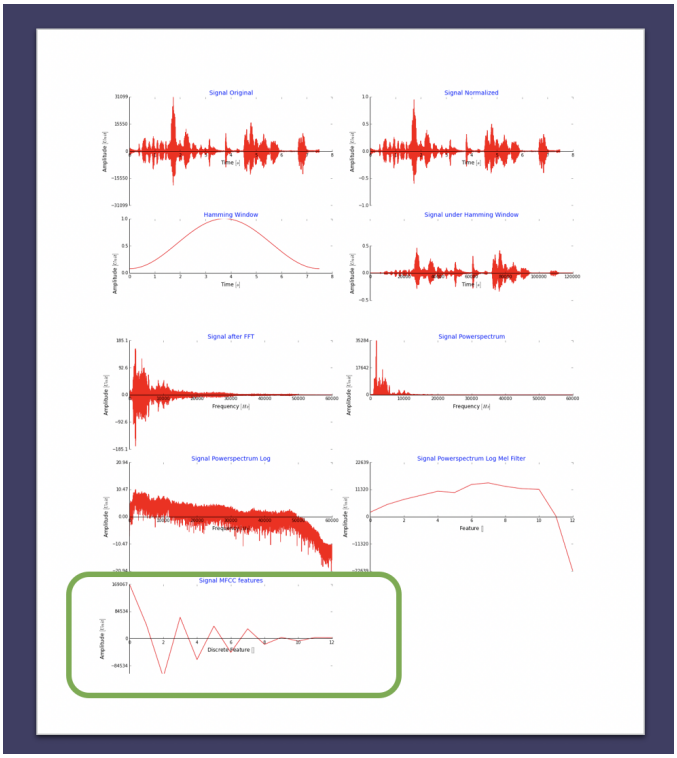

Figure 1: Conversion of waveform to MFCC

It is interesting to observe that we cannot distinguish between close frequencies, especially at higher frequencies. For example, if we hear a sound at 100 Hz and 200 Hz, we might notice the difference, but we will probably not be able to differentiate sounds at 1100 Hz and 1200 Hz. We utilize this observation and use logarithm of the periodograms to obtain which allows us to perform cepstral mean subtraction, which is a normalization technique.

There is still overlapping in this cepstral obtained. To handle that, we take the discrete cosine transformation on the logarithmic periodogram to obtain the final Mel-Frequency Cepstral Coefficients. We then compare it with the Mel-scale. We usually keep only 12-26 coefficients as it is observed to degrade the features and based on observation mentioned before, we do not need

changes in higher frequencies as they are not noticeable for humans. For our experiments, we have taken only 12 MFCCs.

V. DATASETS

A. George Mason University

The first dataset contains 2140 speakers from 177 countries, having 214 different native languages. The content of this dataset is made by speakers speaking same sentence. The audio contains noise.

B. University of California, Irvine

The second dataset contains 400 speakers from 6 different accents speaking in English. The data is derived from random words and sentences spoken by different speakers. There is no noise in this dataset. The audio is converted to MFCC.

Both of these datasets have respective metadata .csv files that give information about the speaker's geography, native language, filename, age, etc. Using this and the audio files, we further generate another metadata .csv file where each row represents the target variable i.e., the country to which the given accent is supposed to be classified and the 12 MFCC features associated with that speaker.

VI. DESIGN AND IMPLEMENTATION

The project involves the following stages of implementation:

- Data gathering from sources
- Conversion of .mp3 files to .wav files
- Removing silence from .wav files
- Converting .wav files to MFCC features
- Splitting the dataset into train and test data.
- Baking the models – SVM and CNN
- Testing the models
- Deriving performance matrix
- Plotting the results

Implementation code can be found here: https://github.com/parijatparimal29/Accent_Finder

VII. MODELS

Support vector machines algorithm provides margin guarantees that make it a good classification algorithm. The main idea is to set up a hyperplane that divides two different category of input points. The algorithm finds the possible margin, such that the hyperplane divides these points with minimum error. The margin guarantees in SVM are derived from geometric calculation of margins and then maximizing it. The same is extended to multiple hyperplanes for multi-class classifications.

Convolutional Neural Network is most commonly used for image analysis. Our problem of working with audio files is actually reduced to an image analysis one, thanks to the conversion of .wav files to MFCC features. Each image representation of .wav file is fed into the 2D CNN model as a tensor. The convolutional layer transforms the image to a feature map. The activation is performed by the ReLU layer. Further a pooling layer is used to reduce the dimension of the data.

VIII. RESULTS

Multiple algorithms were tried to see which algorithm gets us better classification. We tried k-Nearest Neighbors (KNN), Multi-Layer Perceptron (MLP), Support Vector Machines (SVM) and Convolution Neural Network (CNN) to find Accents from MFCC data. We found SVM and CNN to perform better than others by at least 10% increment in accuracy.

We used the count of occurrence of the most common accent in our test dataset as the baseline accuracy to be the minimal accuracy acceptable for our dataset. On the second dataset we obtained an accuracy of 86.6% [Figure 2] with SVM and 93.4% accuracy with CNN [Figure 3].

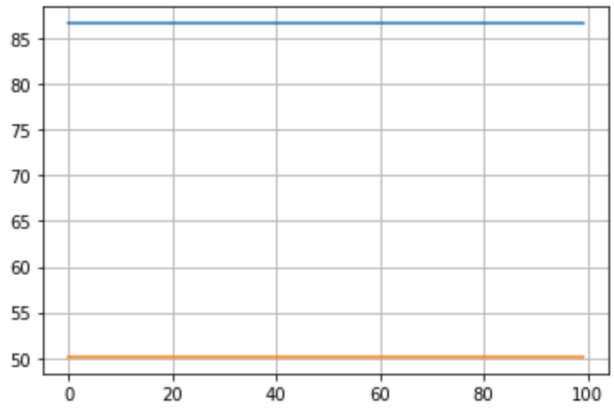

Figure 2: SVM: Accuracy per iteration [y-axis = Accuracy; x-axis = iteration; blue = Accuracy; Yellow = Baseline]

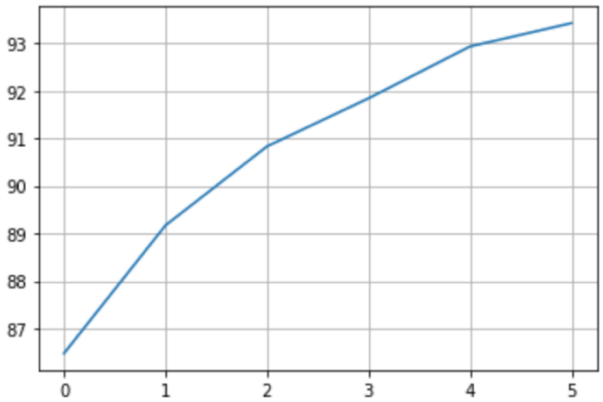

*Figure 3: CNN: Accuracy per 500 iterations. [y-axis = Accuracy; x-axis = (x+1) * 500 iterations]*

On dataset A, our accuracies were barely beating the baseline we had kept (53% with baseline of 50% and 72% with baseline of 70% with SVM). Although we tried to reduce the noise, we believe the noise reduction was not good enough, which is why the accuracies are not as good. As the Dataset B has no noise, the performance is great. We wanted to work on reducing noise and improving the model for Dataset A as well, but with the limited time, we were not able to reach at satisfactory results yet. Thus, we derive our conclusions based on the results from Dataset 2.

The model performed well on the test data without significant standard deviation. We achieved an average accuracy of 79.34% using the SVM model [Figure 4], whereas the CNN performed better at this too with 87.3% accuracy [Figure 5].

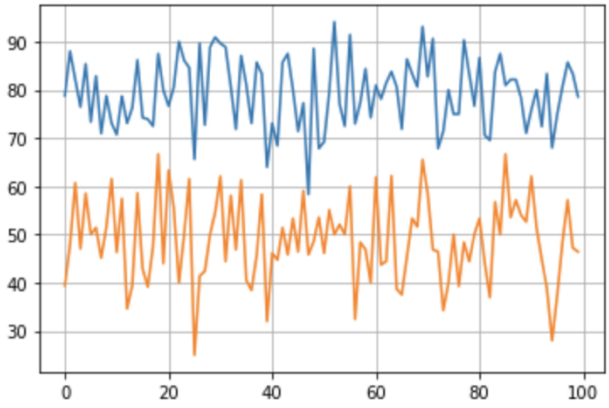

Figure 4: SVM's mean accuracy of 79.34% on test data

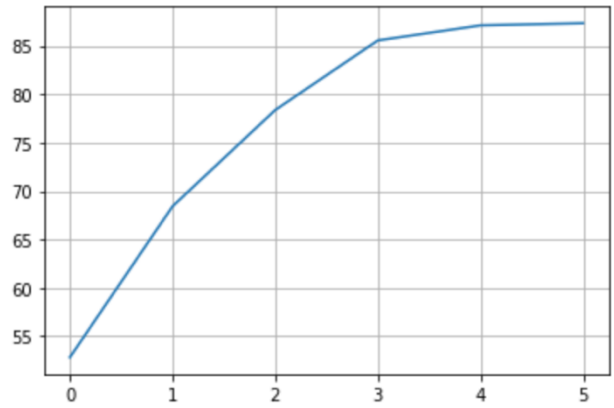

Figure 5: 87.3% Test accuracy achieved on CNN

IX. Future Work

We achieved promising results on the few hundred samples that we worked with. However, there are a few more areas to focus on. Noise removal / reduction when working with datasets of varied quality will help to extract better MFCC features. In terms of features, metadata from the audio file such as the duration of the audio, pitch, bitrate, etc. could be used to enhance the classification. A step up from here would be to test the application for real time input with a pre-trained model. Extending the use case to classify with overlapping geographical target. For instance, speakers from the South Asian region may have similar accents. It is difficult to segregate a Sri Lankan English accent from that of a South Indian English accent.

X. Conclusion

We achieved our objective of classifying multiple accents based on the geography that the speaker belongs to. MFCC extracts the vocal signature of the speaker and removes dependency of pronunciations of individual words to derive accents. SVM performs well to classify accents with accuracy of 86.6%. CNN performs even better with an accuracy of 93.4%. We conclude that using MFCC is a very useful tool for feature extraction, not only for Accent recognition but also for other speech recognition tasks. It would be interesting to scale up the experiments on larger datasets and handle the noise to make the application more robust. We would further want to use our code to create a user-friendly application that records audio from user and using the learned parameters predicts the geographical location the user is from. Further, accent recognition can be used to improve accuracy of speech recognition by using accent specific speech recognition techniques for the user's accent.

XI. References

1. Ma, B., Yang, F., Zhou, W., Accent Identification and Speech Recognition for Non-Native Spoken English 178 https://web.stanford.edu/class/cs221/2017/restricted/p-final/boweima/final.pdf

2. Neidert, J., Chen, P., Lee, J., Foreign accent classification, 171 http://cs229.stanford.edu/proj2011/ChenLeeNeidert-ForeignAccentClassification.pdf 172

3. Upadhyay, Rishabh. Accent Classification Using Deep Belief Network, University of Mumbai, pages 6-7, 176 2017. 177

4. Wang, X., Guo, P., Lan, T., Fu, G., Study of Word-Level Accent Classification and Gender Factors 173 http://students.cse.tamu.edu/xingwang/courses/csce666_accent_native_indian.pdf 174

5. Ge, Z., Tan, Y., Ganapathiraju, A., Accent Classification with Phonetic Vowel Representation

6. Rachael Tatman, (2017). Speech Accent Archive. https://www.kaggle.com/rtatman/speech-accent-archive

7. Fokoue, E. (2020). UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/Speaker+Accent+Recognition#

8. Garg, Yatharth. (2017). Speech Accent Recognition. https://github.com/yatharthgarg/Speech-Accent-Recognition

9. Ma, Zichen & Fokoue Ernest. (2015). A Comparison of Classifiers in Performing Speaker Accent Recognition Using MFCCs. arXiv:1501.07866.

10. Sheng, Leon & Edmund Mok. (2018). Deep Learning Approach to Accent Classification. Stanford University. http://cs229.stanford.edu/proj2017/final-reports/5244230.pdf