Speaker Region Identification for CORAAL Dialects

Michi Jewett[1+*], *Shaosong (Patrick) Xing*[1*]

[1]Department of Electrical and Computer Engineering, UCLA
[+]Contribution to Paper [*]Contribution to Project

mbjewett318@g.ucla.edu

Abstract

This paper presents a classification system that uses the acoustic and spectral features of an utterance to predict the city of origin of the speaker based on their detected regional dialect. The model is trained, validated, and tested on the Corpus of Regional African American Language (CORAAL), which contains speech samples from five different US cities. Our results show that MFCCs are the most impactful features for classifying the speaker's city of origin, but also that a high accuracy can be achieved without them, using careful feature selection. Our findings support that noise mitigation techniques are critical for robustly achieving high accuracy in speaker identification systems and we propose innovative and novel future avenues to pursue to improve performance.

1. Introduction

Regional dialects can vary between countries, states, or even within cities, and their features are sometimes drastically different from each other and othertimes very sublte. The accurate classification of these dialects is incredibly important is eliminating biases that inherently exist in certain systems due to misinterpretations of local dialects that do not fit the prescribed lexicon or dialect of the mainstream. There are also applications in forensics and accurate/inclusive speech synthesis. In this project, we aim to implement a system that will classify the regional dialect of a speaker using two feature extraction pipelines: one without Mel-Frequency Cepstal Coefficients (MFCCs) used as a feature and one including them. A key goal of this study is both to verify the importance of MFCCs and demonstrate that great performance can still be achieved by other features in their absence. We train, validate, and test on datasets from the Corpus of Regional African American Language (CORAAL), which contains speakers each belonging to one of five different US cities: Rochester, NY (ROC); Lower East Side, Manhattan, NY (LES); Washington DC (DC); Princeville, NC (PRV); and Valdosta, GA (VDL).

2. Background

The African American Language (AAL), also known as African American English (AAE) or African American Vernacular English (AAVE), is a dialect of English that has developed within African American communities in the United States. It is a complex and diverse linguistic system with its own set of rules and conventions, and it has been shaped by a variety of factors, including African languages, Spanish language, Southern American English, other regional American English dialects, and cultural and social experiences.

These influences have contributed to the development of distinct regional dialects of AAL, which can be distinguished by subtle differences in pronunciation, grammar, syntax, and intonation. Through data-mining and machine-learning methods, it is possible to analyze the characteristics of these dialects and predict the origins of speakers based on their speech patterns.

The task of speaker dialect and region identification has been extensively studied in the speech processing community. Various techniques have been proposed to extract relevant features from the speech signal to perform this task. In the realm of speech and speaker recognition, MFCCs are considered the most predominant features to extract. However, as evidenced by the studies such as the one published in the Tsinghua Science & Technology Journal, by Zunjing Wu and Zhigang Cao, MFCCs exhibit heightened sensitivity to noise interference. This results in substantial performance degradation in recognition systems due to discrepancies between training and testing conditions. The study proposes an enhancement to the standard MFCC analysis by substituting the logarithmic transformation with a composite function to mitigate noise susceptibility. Additionally, the refined feature extraction procedure incorporates speech enhancement techniques such as spectral subtraction and median filtering to further suppress noise. Experimental outcomes demonstrate that the advanced robust MFCC-based feature considerably diminishes recognition error rates across an extensive signal-to-noise ratio spectrum [1]. Thus, in this project, we decided to use this knowledge by applying statistics which are more robust to noise (such as medians over means) and implementing noise reduction.

3. Project Description

3.1. Baseline Performance

We began by establishing a baseline using only MFCCs, which yielded 79.6% and 58.21% accuracy scores on clean and noisy data respectively. We then proceeded to implement the most common features (e.g., LPC & Pitch) quickly using the *Librosa* Library, to ensure we were on the right track. This gave us an upgraded baseline performance to iterate on. We are using the XGBoost classifier, which is an optimized distributed gradient boosting library that has been engineered to offer exceptional efficiency, adaptability, and portability. This library encompasses machine learning algorithms within the framework of Gradient Boosting and uses a parallel tree boosting approach—often referred to as Gradient Boosted Decision Trees (GBDT) or Gradient Boosted Machines (GBM)—which addresses a myriad of data science challenges with remarkable speed and precision [2].

3.2. Choosing a Larger Feature Set

To keep a reasonable scope, we desired a feature set of around 100 features to consider, leading us to the extended Geneva Minimalistic Acoustic Parameter Set (eGeMAPSv02) of 88 features [3]. We leveraged the *openSMILE* library implementation of this feature set [4]. We created a python script to rapidly prototype the addition and omission of features to our main pipeline using a naive, iterative grid-search and thereby determine the best 10-20 features to run on. We initially decided to use PLP over LPC, but we decided to forgo both in the end since they provided only marginal impact and produced errors.

We selected a feature set that included all our key targets, with some variations in parameters and ranges to eliminate our need for excessive fine-tuning. We closely investigated each feature in the set and determined the features that seem especially useful when classifying a dialect, as they give information about speech variability and vocal tract characteristics, so we could test our assumptions about their viability after we optimized the piplines. Some examples are the following: Jitter, the variability in the time between glottal pulses; Shimmer, the variability in the amplitude of the glottal pulses; Spectral Slope, the Linear Regression Slope of the log power within some frequency band; Hammarberg Index, the energy difference in lower vs higher frequency bands; and Spectral Flux, the difference of the spectra in two consecutive frames; and Harmonic Difference, the Ratio of the energies between the fundamental frequency and a given formant range.

3.3. Naive Noise Mitigation

To mitigate noise, we used techniques such as Cepstral Mean Subtraction, Spectral Gating, and Wiener Filtering. Cepstral Mean Subtraction is used to remove the effect of channel distortion and noise from speech signals by demeaning the MFCCs by assuming that the channel distortion and noise affects the signal in a similar way across the entire signal. Spectral Gaiting is a technique that programs like Audacity use to analyze the frequency content of the signal and selectively filter out frequency ranges that are dominated by noise. Finally, Wiener Filtering computes a statistical estimate of an unknown signal using a related signal as an input and filtering that known signal to produce the estimate as an output. In the end, some of these improved the noisy performance of various pipeline designs, but negatively impacted the clean performance, so in the end we decided to not utilize them. However, they were still notable steps in the process of deciding upon our feature extraction, since certain features performed better with or without them.

3.4. Intelligent Noise Mitigation

There are various methods of intelligent noise mitigation that we experimented with, but we did not have the time or resources to delve fully into them as we would have wished. One attempt was to utilize the zero crossing rate and signal to noise ration (SNR) to discriminate between clean and noisy signals during pre-processing, and only apply noise mitigation and filtering to those signals so that the performance on clean data would not be affected. We will expand on this in the *Future Work* section.

3.5. Feature Selection

We performed feature extractions into two different pipelines. Pipeline 1 utilized Pitch, Mel Spectrogram, F2 & F3, Jitter & Shimmer, Harmonic Difference F0:F3, Loudness, Spectral Slope 0-500Hz, and Zero Crossing Rate. Note the absence of MFCCs. Pipeline 2 utilized MFCCs 1-12, Pitch (20th Percentile), Mel Spectrogram (Mean & Median), F1 (Standard Deviation), Spectral Slope 0-500Hz (Mean), and Zero Crossing Rate (Mean & Median).

Data	Pipeline	Accuracy
Given Clean	1	94.3%
Given Noisy	1	70.2%
Given Clean	2	93.1%
Given Noisy	2	82.5%
Given Clean	BC	96.4%
Given Noisy	BC	55.5%
Given Clean	BN	87.5%
Given Noisy	BN	87.6%
Hidden Clean	2	74.2%
Hidden Noisy	2	46.5%

Table 1: *Summary of Performance on Test Data*

3.6. Runtime Performance

We extracted all 88 features and then saved them as pickle files to eliminate the need to run extractions for every single iteration of our optimizer. Therefore, the average feature extraction process using openSMILE took around 2-3 hours, and optimization took about an hour longer than this. However, if we factor in noise mitigation and filtering techniques, the runtime increased by about 2x or 3x depending on which filters were running. Additionally, because noise mitigation is a pre-process before extraction, we had to re-run all 88 features if we wanted to test a different noise mitigation or filtration process.

4. Results

4.1. Summary

We trained Pipelines 1 and 2 on the same given training set. They were also both tested on given, known test data (both noisy and clean.) Additionally, we chose Pipeline 2 to be tested on the "Hidden" dataset that was unknown to us at the time of doing the project, and was used to test our model after completing it. In a sense, this is the true "Test" data, since it measures the models performance on sets unseen, while the known Clean and Noisy data given to repeatedly test on is more akin to "Validation" data, since we were able to optimize feature extraction for those datasets. Along with these six metrics, we included metrics for Pipeline "BC" and Pipeline "BN", which indicate respectively the pipelines with the best performance individually on Clean data and Noisy data. The best accuracy achieved by *any* pipeline on clean data was 96.4%, but the accuracy for that pipeline on noisy data was only 55.5%. Similarly, the best accuracy achieved by *any* pipeline on noisy data was 87.6%, but the accuracy for that pipeline on clean data was only 87.5%. This was notably the only pipeline with a higher accuracy score on noisy data, which means we were on the right track with our noise reduction techniques. Pipeline BN utilized Cepstral Mean Subtraction, Spectral Gaiting, and a Wiener filter. Ultimately, however, we decided to settle on Pipeline 2 as the best overall feature extraction pipeline. A summary of all these results can be seen in Table 1.

4.2. Pipeline 1 In-Depth Performance

Pipeline 1 achieved an accuracy of 94.3% on clean data and 70.2% on noisy data. The spectral slope, loudness, and harmonic difference were the standout features that gave the biggest improvement. Figures 1 and 2 show the confusion matrices for Clean Test Data and Noisy test Data Respectively. A confusion matrix shows the truth labels of the data versus the bins they were classified into; thus, ideally we would like to see all of the values on the diagonal. Through these matrices we can also make observations about which dialects our data performs well on, and which dialects it confuses easily. It seems that the dialect of speakers from VLD are mislabelled most commonly, both being mistaken for other dialects (such as DCB) and having other dialects being classified as them (again, commonly confused with DCB).

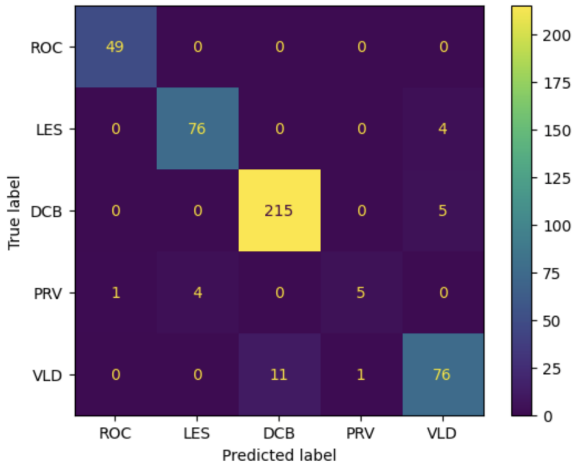

Figure 1: *Pipeline 1 Confusion Matrix for Clean Data*

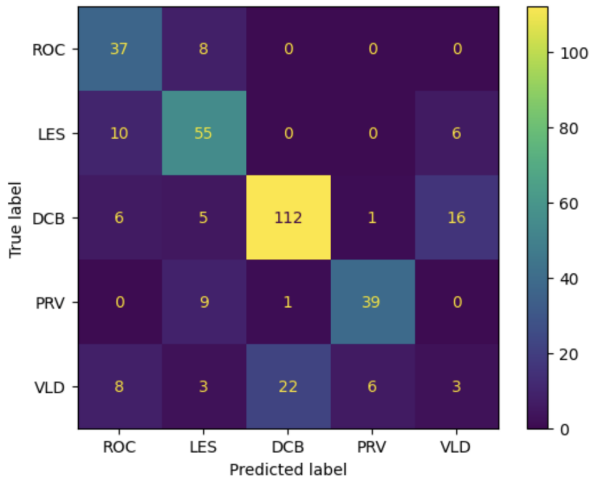

Figure 2: *Pipeline 1 Confusion Matrix for Noisy Data*

In Figure 3 we use the *shap* Python library to create a plot revealing the importance of each feature to Pipeline 1. Based on the plot, the spectral slope, loudness, Harmonic Difference (F0:F3), and Pitch were the standout features that gave the biggest improvement. The following were the most important discriminators per class:

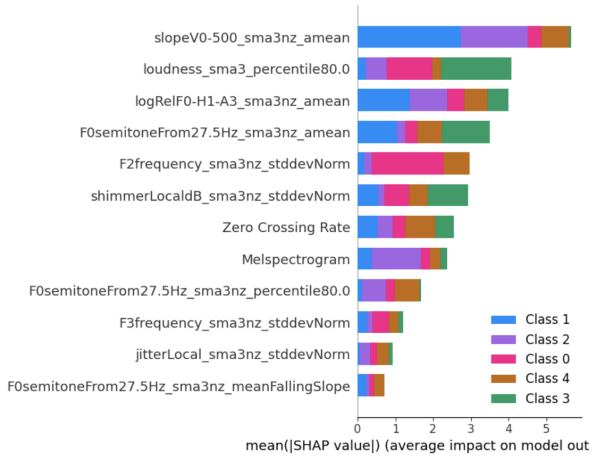

Figure 3: *Pipeline 1 Feature Importance*

- ROC (Class 0): F2 Std. Dev. & Loudness
- LES (Class 1): Spectral Slope
- DC (Class 2): Mel Spectrogram & Spectral Slope
- PRV (Class 3): Loudness
- VLD (Class 4): All features nominally equivalent in impact

4.3. Pipeline 2 In-Depth Performance

Next we see that, Pipeline 2 achieved an accuracy of 93.1% on clean data and 82.5%, which is noticeably better than Pipeline 1. However, the performance on the Hidden Data left much to be desired. We hypothesize that there were features in the hidden set that were not captured by our model but that were very impactful. Given how low the noisy accuracy is, we suspect that, if we had spent more time on noise mitigation, Pipeline 2 would have at least had less variance in the accuracy metrics on the Hidden sets between clean and noisy signals.

Figures 4 and 5 show the confusion matrices for Clean Test Data and Noisy test Data Respectively. The data show again that our main error comes from the VLD dialect being confused with LES, DCB, and ROC. However, Pipeline 2 actually confuses clean LES signals with VLD instead, but continues to confuse DCB with VLD for noisy signals, leading us to believe some feature of the noise is dampending the expression of a key distiguishing feature of the LES dialect that brings it closer to a DCB dialect (as our model incorrectly predicts about the same number of each to be VLD). Notably, it only mis-predicts VLD dialects as ROC when noise is present, suggesting that perhanps ROC has acoustic or spectral qualities closer to noise than other dialects.

In Figure 6 we use the *shap* Python library to create a plot revealing the importance of each feature to Pipeline 1. Based on the plot, the MFCC's contributed a lot (especially MFCC 0 mean), while Jitter and Shimmer were outclassed by MFCC performance, yet the Spectral Slope 0-500 Hz remained important. The following were the most important discriminators per class:

- ROC (Class 0): MFCC 0 Mean
- LES (Class 1): Spectral Slope

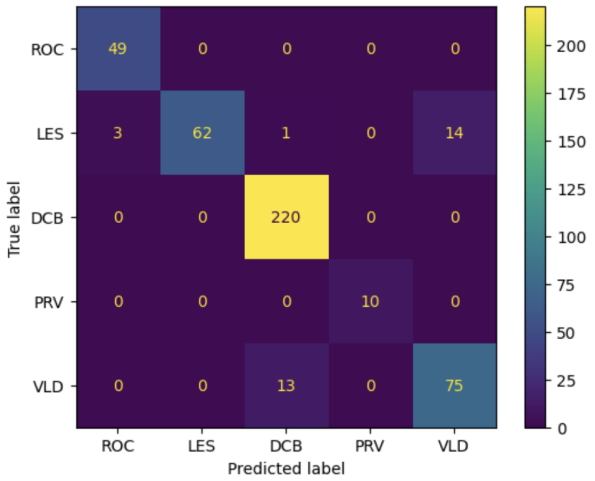

Figure 4: *Pipeline 2 Confusion Matrix for Clean Data*

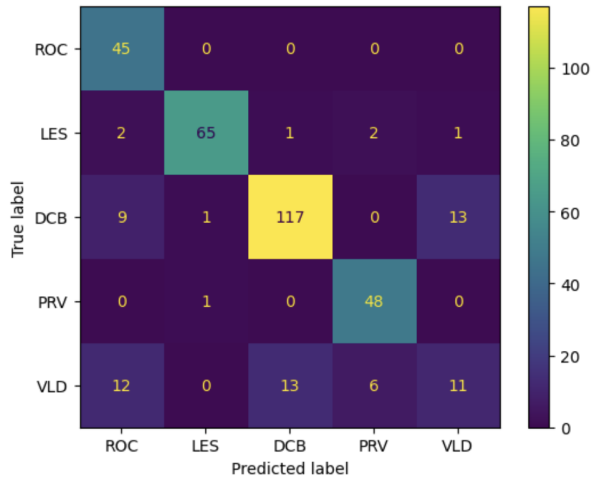

Figure 5: *Pipeline 2 Confusion Matrix for Noisy Data*

- DC (Class 2): Mel Spectrogram & MFCC 0/2
- PRV (Class 3): MFCC 0 Mean (Almost Completely!)
- VLD (Class 4): MFCC 6/10 Mean

5. Discussion

5.1. Significance of Results

Our results overall show that the MFCCs were the most impactful features for classifying the speaker's city of origin. We found that Pipeline 2, which used MFCCs, achieved a higher accuracy on noisy data than Pipeline 1, which did not include MFCCs. This is consistent with previous research, which has shown that MFCCs are a powerful feature for speech recognition and speaker identification. However, using various features besides MFCCs enabled accurracy as high as 96.4% (yes, Pipeline BC did not use MFCCs!)

We also found that certain features, such as Jitter and Shimmer, were outclassed by the MFCC performance in Pipeline 2. These features provide information about speech variability and vocal tract characteristics that are specific to different

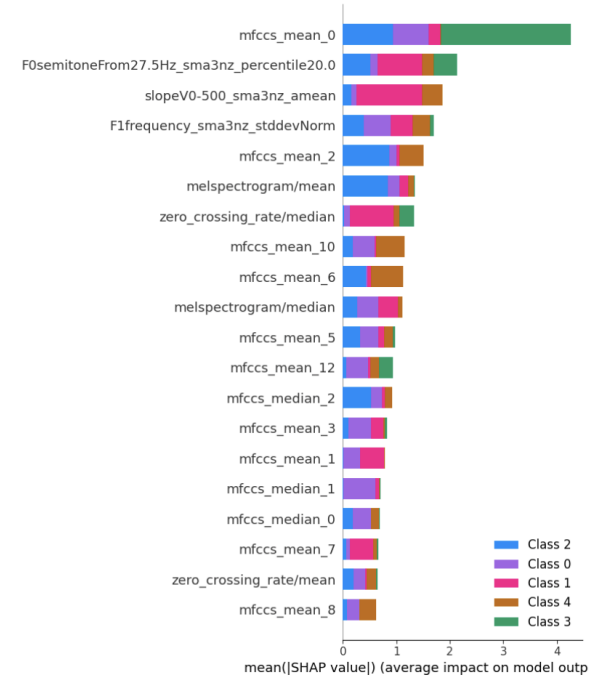

Figure 6: *Pipeline 2 Feature Importance*

dialects, and were able to achieve great performance on Clean Test data even without the MFCCs in the mix. In summary, our study shows that MFCCs are a powerful feature for speaker identification, and that feature selection and noise mitigation techniques, such as Cepstral Mean Subtraction, Spectral Gating, and Wiener Filtering can significantly improve the accuracy of speaker identification systems in noisy environments and are critical for achieving high accuracy in speaker identification systems.

5.2. Future Work

Given more time, we would have like to have explored the use of other feature sets or classifiers. Though we did not utilize noise mitigation or filtering directly in our final pipeline, future work on this project would entail developing methods of more intelligent and selective noise reduction that will not damage the clean signal performance in the process. One particular area of interest to us is the prospect of discriminating during pre-processing between clean data and noisy data, and applying noise mitigation techniques to *only* data considered noisy enough, so that we can improve our performance on noisy data without hampering the performance on clean data. Candidates for features to aid in the discrimination of noisy signals include:

- Short-Time Objective Intelligibility (STOI): A higher STOI score indicates a cleaner signal, while a lower score implies a noisy signal.
- Signal-to-Noise Ratio (SNR): A high SNR indicates a cleaner signal, while a low SNR implies a noisy signal.
- Spectrogram Analysis: In a clean speech signal, the spectrogram will display well-defined formant structures and harmonic patterns, whereas a noisy signal will exhibit irregular and random patterns.
- Voice Activity Detection (VAD): Analyze energy levels, zero-crossing rates, or spectral features to differentiate be-

tween speech and non-speech (noise) components. A clean signal will predominantly contain speech segments, whereas a noisy signal will have more non-speech components.

- Statistical Analysis: Examine mean, variance, kurtosis, and skewness. Clean speech signals generally have distinct statistical characteristics compared to noisy signals.

It is also worth noting that Deep-Learned noise reduction is a relatively recent prospect and has been showing very promising results. In a 2019 article from the Analog Integrated Circuit and Signal Processing journal, we see two types of autoencoders, convolutional and denoising, reconstruct the audio signals in the output of a neural network after extracting the meaningful features that present the "pure and the powerful information" [5]. This will become more and more viable as deep learning and higher computing power continue to become more accessible.

6. Conclusion

In this project, we have implemented a speaker identification system that predicts a speaker's regional dialect given an utterance, using speech feature extractions in tandem with machine learning. Our results confirm that MFCCs are a powerful feature for speaker identification, but more importantly demonstrate that with careful optimization, respectable performance can be achieved without them. Our study also highlights the both the importance and potential of optimized feature selection and intelligent noise mitigation techniques in speaker identification systems and suggests avenues for future research to further improve performance in this area.

7. References

[1] Wu, Zunjing & Cao, Zhigang. (2005). Improved MFCC-Based Feature for Robust Speaker Identification. Tsinghua Science & Technology. 10. 158-161. 10.1016/S1007-0214(05)70048-1.

[2] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785

[3] Xue, Wei & Cucchiarini, Catia & Hout, Roeland & Strik, Helmer. (2019). Acoustic correlates of speech intelligibility: the usability of the eGeMAPS feature set for atypical speech. 48-52. 10.21437/SLaTE.2019-9.

[4] Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. Opensmile: the munich versatile and fast open-source audio feature extractor. In Proceedings of the 18th ACM international conference on Multimedia (MM '10). Association for Computing Machinery, New York, NY, USA, 1459–1462. https://doi.org/10.1145/1873951.1874246

[5] Abouzid, H., Chakkor, O., Reyes, O.G. et al. Signal speech reconstruction and noise removal using convolutional denoising audioencoders with neural deep learning. Analog Integr Circ Sig Process 100, 501–512 (2019). https://doi.org/10.1007/s10470-019-01446-6