Classification of Accents of English Speakers by Native Language

Morgan Bryant
mrbryant@stanford.edu

Amanda Chow
amdchow@stanford.edu

Sydney Li
sydli@stanford.edu

Introduction

Accents can reveal a lot about a person's background, such as their native language, place of origin, or ethnic background. Being able to recognize different type of accents can also improve the quality of speech to text transcription by allowing for specific preprocessing of recordings based on the type of accent. Our goal is to classify various types of accents, specifically foreign accents, by the native language of the speaker. Given a recording of a speaker speaking a known script of English words, we would like to predict the speaker's native language.

Dataset

The recordings were scraped from the George Mason University Department of English Speech Accent Archive [10]. Each recording is of a person speaking the same English script:

> *Please call Stella. Ask her to bring these things with her from the store: Six spoons of fresh snow peas, five thick slabs of blue cheese, and maybe a snack for her brother Bob. We also need a small plastic snake and a big toy frog for the kids. She can scoop these things into three red bags, and we will go meet her Wednesday at the train station.*

For each clip, the archive contains information about the speaker's background, such as their age, gender, birthplace, native language, other languages spoken, age of English onset, English residence, and length of English onset. The characteristics that are of importance to us are each speaker's native language. We chose to distinguish accents of only male speakers of the five most common native languages in the archive - English, Spanish, Arabic, French, and Mandarin - for a total of 293 sample recordings.

Methodology Overview

The distinguishing characteristic between accents is the different enunciation patterns of specific syllables, which are due to speakers' difficulties pronouncing English phonemes that do not appear in their native language. Therefore we designed our classification algorithm to capitalize on this difference by comparing different speakers' enunciations of each syllable in the recording and using this information to model how speakers of each language enunciate each syllable in the script. The testing pipeline then uses this information to determine which of the five accents the speaker's is the most similar to.

The algorithm we designed is a two-step process. First, we create a classification algorithm for each syllable that gives us the likeliest accent of the speaker based on that specific syllable. Then, we use this list of likeliest accents (one for each syllable), pick the most frequent accent, and declare it as the speaker's likeliest accent. We select the most important words to use in this step later on and filter out unimportant and repeated words. If we consider the list of likeliest accents to be a feature vector for the recording, this word selection process is analogous to Principal Component Analysis.

Preprocessing Data

Recordings used for speech processing typically require a decent amount of preprocessing before feature vectors can be extracted. The first preprocessing step would have been to remove background noise; however, the data set had minimal background noise.

Most of our preprocessing consisted of splitting up the recording as mentioned above. Because most of the words in the script are monosyllabic, we split the recording into individual words. We aligned words using the Munich Automatic

Segmentation System (MAUS) [7][8], which utilizes a variety of filters, processing, and heuristics to search for divisions between words. Aided with the script above, we used MAUS to split each recording into smaller clips and samples of each word. Clips of the same word spoken by different speakers inevitably end up being different lengths due to natural variation in speaking rate from person to person. We account this variation later on in the feature extraction step.

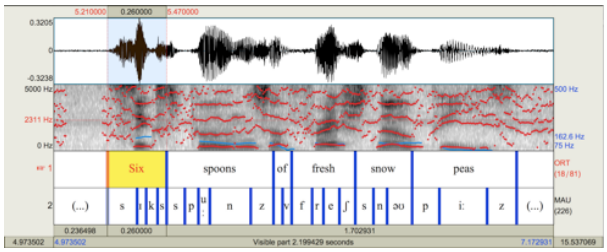


Figure 1. Interface of MAUS. The audio "Six spoons of fresh snow peas" is shown in time-domain and frequency-domain, and the phoneme/word boundaries extracted from the program are shown in blue.

The final step consisted of normalizing the clips by volume, because different speakers may speak at different volumes or distances from the microphone. We can normalize for variation in volume after aligning words by scaling based on the extrema of each word to the average speaking volume of each particular word over the entire training set.

Feature Selection per Word

For each sample of each word, our features are the Mel-frequency cepstral coefficients (MFCC's), a convenient and compact way to extract features that represent audio samples of waveforms. Past papers have shown that MFCC's are particularly useful for speech recognition purposes, which is exactly what we are doing. The Mel spectrum is a method of categorizing the frequencies of a sound in a way that distinguishes phonemes effectively. The MFC process takes a signal's Fourier transform and puts the powers into sized buckets, as defined on the Mel spectrum. The time window must be small, around 10 milliseconds, for the process to be effective. As mentioned above, we normalized for different speaking rates of each word while extracting the MFCC's. We accomplished this by scaling the sampling rates of the .wav files, which often vary from microphone to microphone. The resulting vector of the MFCCs for each word is the set of features that we use to model each word.

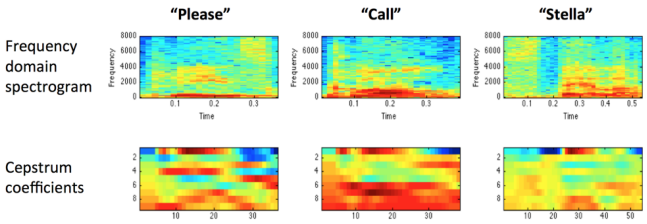


Figure 2. The top row is the sample in frequency-domain, and the bottom row is a representation of the cepstrum coefficients. The "redness" of a square represents a higher valued number. We used these coefficients as our features.

Since the vector has many more features than we need, we identified the 16 most important features for each word with PCA, running the unsupervised learning model on the set recordings of every word. The selected MFCC buckets were not necessarily the same between two given words. For each word, we set the vector of only specific index-identified MFCCs to be our newly slimmed-down features vector for that word. Overall, using such a truncated feature set was necessary for computational practicality.

Word Models

Since our dataset was labeled, we used a variety of both supervised and unsupervised learning algorithms to create models for each word. For supervised learning, we trained our models on a randomly sampled 80% of the full data set multiple times. To calculate the test error of a given model, we ran the trained models on the remaining 20% of the dataset and took the average error. The training sets contained 205 samples, and the test sets contained 88 samples.

The supervised training algorithms used were SVM, Naïve Bayes, Softmax logistic regression,

and GDA. The SVM model was implemented exactly as learned in class, with a Gaussian Kernel. The Naive Bayes model was modified to classify between multiple classes instead of two. Essentially, we performed binary classification on each category, such that all samples in that category are classified as positive and all other samples are classified as negative.

The Softmax model was generalized to classify into five classes to minimize the cost function

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))\right]$$

in the form of multinomial Softmax logistic regression with the resulting hypothesis

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1|x^{(i)};\theta) \\ p(y^{(i)} = 2|x^{(i)};\theta) \\ \vdots \\ p(y^{(i)} = k|x^{(i)};\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}$$

For GDA, we modeled our data as follows:

$$y \sim \text{Multi}(\phi_1, \ldots \phi_5)$$

$$\text{for } k = 1 \text{ to } 5: \quad x|y = k \sim N(\mu_k, \Sigma)$$

Then we maximized the log likelihood:

$$l(\phi, \mu, \Sigma) = \log \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}; \phi, \mu, \Sigma)$$
$$= \log \prod_{i=1}^{m} p(x^{(i)}|y^{(i)}; \phi, \mu, \Sigma)\, p(y^{(i)}; \phi)$$

We also experimented with two unsupervised learning algorithms, a Gaussian Mixture Model and k-means clustering, to see whether the clusters would match reasonably with our labelings. Also, from previous research, GMMs seem to be a standard for tasks like speech recognition and analysis. GMMs and k-means were ran without modification as learned in class, and then compared to our labels.

Word Selection "PCA"

To eliminate bias due to repeated words or unimportant words, we analyzed the strength of each word as indicative of a certain accent. We used the SVM model to classify between English and each of the other four languages using features from each of our words. The words that resulted in the lowest classification errors between each of our language pairs were then selected as features the classification algorithms above.

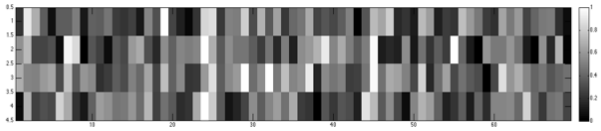
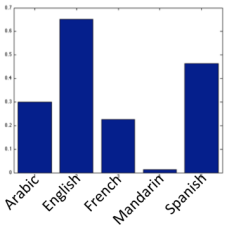


Figure 3. Relative classification error from models trained on each word. Along the horizontal axis is the word being trained. Along the vertical axis is the native language being classified against English. A brighter rectangle for a word indicates a higher classification accuracy than other words for the same native language. For word evaluation, we chose to discard words with under 55-60% classification rates (close to random chance).

Results

Our best classification rate for the supervised learning algorithms was 42% with GDA and Naïve Bayes. With GMM and k-means clustering, we were able to achieve up to 34% labeling accuracy on average, and up to 40% accuracy.



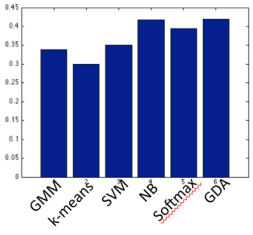


Figure 4a (left). Average classification accuracy of each accent over all the models. The number of samples for each accent type, respectively: 56, 100, 31, 24, 82
Figure 4b (right). Average classification accuracy of each model

The training and test errors of our models are summarized below:

Learning model	Training error	Test error
SVM	0.5678	0.6386
Naive Bayes	0.5800	0.6080
Softmax regression	0.5185	0.6216
GDA	0.5093	0.4057

Unsupervised model	Classification error
Gaussian mixture models	0.3311
k-means clustering	0.3140

Conclusions

Our final classification accuracies of ≈40% are much better than random chance (20%) and are on par with other attempts to classify accents [1][2][3][5]. As expected, the supervised learning algorithms performed better than the unsupervised learning algorithms (Figure 4b). Among various supervised and unsupervised learning methods, our Naive Bayes and GDA models were the most successful at correctly guessing the accent of a test file. We believe that GDA and Naïve Bayes yielded the best results because both account for our uneven prior distribution of samples in each accent category. The classification accuracies for each language also have a very strong correlation with the number of samples for each language (Figure 4a).

However, we had several sources of error that were out of our control. For instance, we did not have the manpower to manually verify the results of our word separation with MAUS during preprocessing, and we only accounted for each speaker's native language rather than their familiarity with English. These sources of error could potentially cause our model to misrepresent the accents.

The second issue occurs the most prominently with our Mandarin speakers, which is a second explanation (aside from the small sample size) for the low classification accuracy of Mandarin. Of our 24 Mandarin samples, about 20 of them had lived in an English-speaking country for a significant number of years. We actually listened to each of the samples and found that only a handful of the 24 spoke with accents - the rest spoke nearly perfect English.

For supervised learning algorithms, the likely result of this "mislabeling" is a model of a Mandarin accent that is very similar to unaccented English. This would affect the classification of all unaccented English samples (of which we have many), which could be almost arbitrarily classified into either the "English" or "Mandarin" class. For unsupervised learning algorithms, the likely result would be the mislabeled Mandarin samples being clustered with the unaccented English samples. If only a handful (the truly accented samples) are clustered into the "Mandarin" cluster, that would explain the low classification accuracy of Mandarin samples.

Overall, we have obtained a reasonable classification algorithm, but there is definitely room for improvement in our methodology.

Future

There are several potential measures we could take to improve our results in the future, beginning with the quality of our data. As discussed above, some of our samples were "mislabeled" in the sense that each speaker's native language was self-identified and sometimes unrelated to their speaking accent. In the future, to create better models of true accents, we would take into account each speaker's familiarity with English as an estimate of how heavy their accent is, using the provided data of the length and nature of their exposure to English.

For the data preprocessing steps and overall model design, there are several improvements that can be made to make our model more precise. First, we had a convenient system (MAUS) at our disposal for splitting the recordings into words, but rather than relying on the fact that most of the words in the script are monosyllabic, we would ideally build a more robust model and split our recordings into syllables or even further into individual phonemes.

Next, the classification algorithms of each word only output a hard 1-5 to represent which of the five types of accents is the likeliest for that word. The issue with this approach is that it simply takes the accent with the highest posterior probability and discards the posterior probabilities of all of the accents without regard for their values. For all we know, a clip could fit the Mandarin accent only very slightly better than it fits the English accent (which could easily happen when our "mislabeled" Mandarin samples cause the English and Mandarin models to be very similar). One solution would be for the classification algorithms for each word to output the vector of all five posterior probabilities. The second step of our model would need to be slightly modified to account for the posterior probabilities of the five accents over all the syllables.

Finally, the second step of our model assigns black and white labels to each word about whether it is important or not to distinguishing between accents. Rather than assigning these black and white labels, we can use some sort of weighted linear regression to weight the syllables. And because certain pronunciations of certain syllables are a red flag for certain accents, we might even weight the syllables different for each accent type, calculate a final posterior for each accent using the posteriors described above, and take the highest posterior to be the likeliest accent.

Given that the distinguishing factor between accents is the pronunciation of individual syllables and phonemes, these three improvements should make our model more precise because they place more emphasis on

An interesting and related path we could follow is the existence of different English dialects and regional accents, such as British accents or regional US accents such as Bostonian accents. Given enough samples, we would hopefully be able to cluster our English samples into various regional accents, and create models for these regional accents in the same way we created models for the foreign accents. These models could form a classification algorithm for predicting where the speaker originated from based on their recording of the script.

Reference

[1] "Accent Classification," Phumchanit Watanaprakornkul, Chanat Eksombatchai, Peter Chien.

[2] "Accent Issues in Large Vocabulary Continuous Speech Recognition (LVCSR)," Eric Chang, Chao Huang, and Tao Chen. Microsoft Research. August 2011.

[3] "Accent Recognition with Neural Network," Matthew Seal, Matthew Murray, Ziyad Khaleq.

[4] "Accurate Short-Term Analysis Of The Fundamental Frequency And The Harmonics-To-Noise Ratio Of A Sampled Sound." Paul Boersma. Institute of Phonetic Sciences, University of Amsterdam. 1993.

[5] "Foreign Accent Classification," Paul Chen, Julia Lee, Julia Neidert.

[6] "melfcc.m," PLP and RASTA (and MFCC, and inversion) in Matlab. Daniel P. W. Ellis. December 2014. Online web resource. 2005. <http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/>

[7] "The Munich Automatic Segmentation System." Ludwig-Maximilians-Universitat, Munich, Germany. Florian Schiel. December 2014. Web. 21 March 2013. <http://www.bas.uni-muenchen.de/Bas/BasMAUS.html>

[8] "Phonemic Segmentation and Labelling using the MAUS Technique," Florian Schiel, Christoph Draxler, Jonathan Harrington. Bavarian Archive for Speech Signals, Institute for Phonetics and Speech Processing. Ludwig-Maximilians-Universitat, Munchen, Germany.

[9] "Praat: doing phonetics by computer." Paul Boersma, David Weenick. December 2014. Computer program. University of Amsterdam. 13 November 2014. <http://www.praat.org/>

[10] "The Speech Accent Archive." George Mason University. Steven H. Weinberger. December 2014. Web. 20 November 2014. <http://accent.gmu.edu/>