Arlie Coles & Inés Patiño Anaya
260612588 & 260507962
LING 550
12 December 2016

GMM Voice Recognizer - Final Submission

1) Introduction

The goal of this project was to design and implement a user-interactive tool for speaker recognition. Answering the question "who is speaking?" in a given audio sample requires acoustic analysis and probability modeling and has a wide range of potential applications, including systems purposes (checking if a speaker is among an approved group to gain access to a secured area, or applying a set of preferences for a specific speaker on a personal computer system after recognizing them) and analysis purposes (audio recordings, perhaps for informational or intelligence reasons, could be categorized by speaker).

2) Speaker Recognition Methods

Text-independent speaker identification-- that is, speaker recognition unspecific to the content of the speech and without any claim of identity by the speaker at the time of testing-- is usually approached in two steps: an initial acoustic analysis of the training auditory signal followed by the construction of a probability model based on that analysis that will then be used for comparison against a testing signal. A widely-used form of signal processing used is the extraction of Mel-Frequency Cepstral Coefficients (MFCCs), which provide a 13-dimensional vector of auditory information for each 10 milliseconds of signal. Subsequent probability models and methods are varied, ranging from Hidden Markov Models, Vector Quantization, and Neural Networks (Tisby 1991; Soong et al. 1985; Farrell, Mammone & Assaleh, 1994). Our implementation takes inspiration from the approach of Reynolds and Rose, where training feature extraction is followed by the construction of a Gaussian Mixture Model (GMM) for the speaker and then the likelihood that the testing features are comprised by the GMM is calculated, resulting in successful speaker recognition

if the likelihood higher than that of any of the other speakers' GMMs in the database (Reynolds & Rose, 1995). Our application makes use of the Expectation-Maximization algorithm to fit the initialized GMM to the MFCC data, both for its ease of application and since previous research has not detected a difference in result between this method and more involved HMM- and phonemically-centered initialization methods (Dempster, Laird & Ruben, 1977; Reynolds & Rose, 1995).

3) Approach

Our application is web-based and completely interactive with the user, hosted on McGill's computer science servers and implemented through HTML and CGI (Common Gateway Interface). A full use of the system consists of a Training half and a Testing half (please see Figures 1 and 2 on the following page for a more detailed overview).

a) Training

The user is prompted to provide a training audio recording, whereupon the system extracts its MFCCs (using scikit.talkbox and scipy.wav utilities), builds a GMM for the speaker (using numpy, scikit-learn, and matplotlib utilities), and saves it in the file system under a username provided by the user.

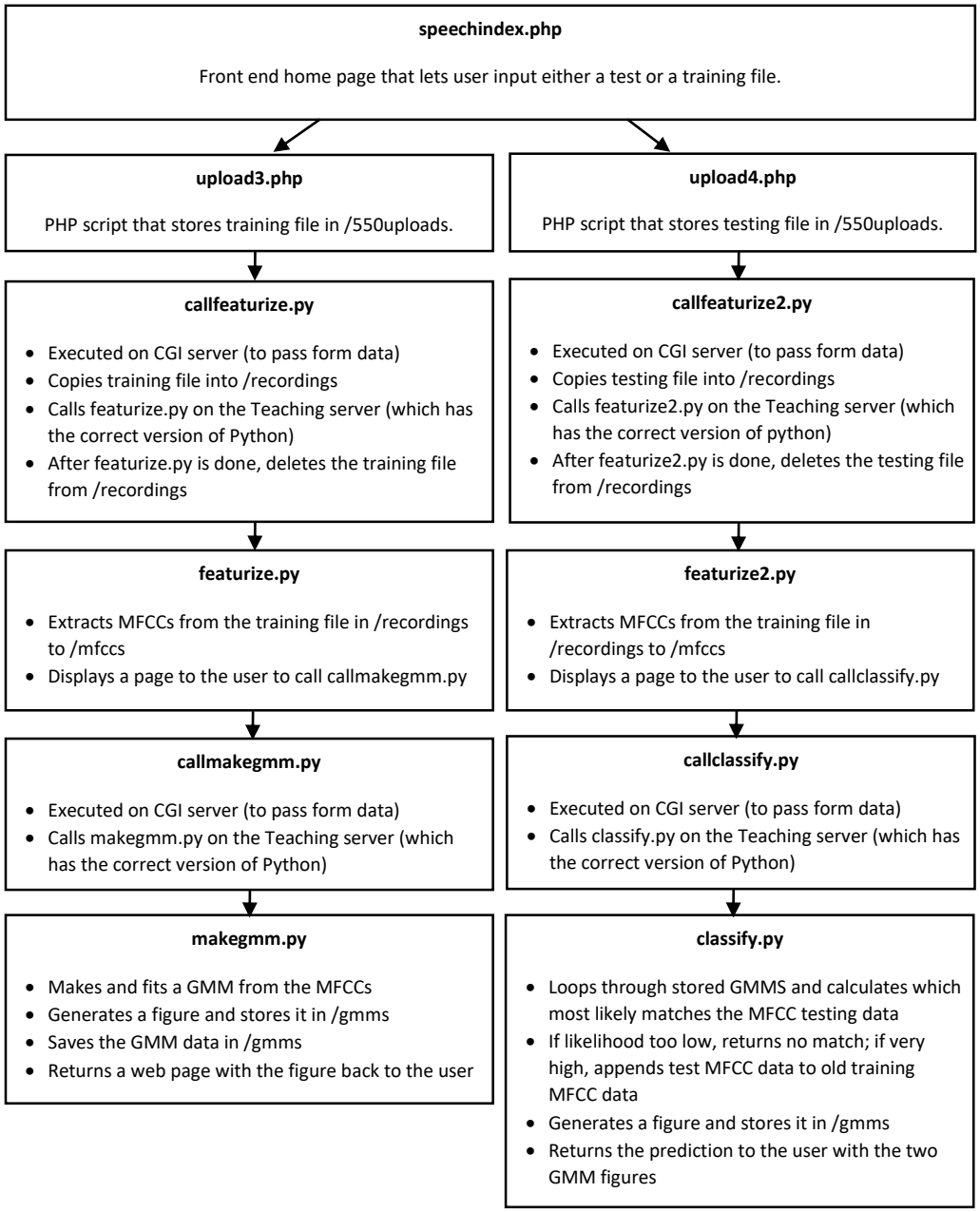
b) Testing

The user is prompted to provide a testing audio recording, whereupon the system extracts its MFCCs and calculates which stored GMM is most likely to correspond to the new data (using numpy, scikit-learn, and matplotlib utilities) and returns the recognition result to the user. If the likelihood of a match for any and every stored GMM is extremely low, no match is returned to the user. If the likelihood is extremely high, indicating a high degree of certainty in the recognition, the testing data is appended to the original training data and a new, more refined GMM for that speaker is created (a form of supervised learning).

Figure 1: File Architecture

public_html					
speechindex.php upload3.php upload4.php	550uploads/				cgi-bin/
	featurize.py featurize2.py makegmm.py classify.py	recordings/	mfccs/	gmms/	callfeaturize.py callfeaturize2.py callmakegmm.py callclassify.py

Figure 2: System Flow



4) Results, Shortcomings, and Future Improvements

In order to evaluate the performance of our tool and make necessary adjustments we tested our program on subsets of recordings from the TIMIT speech corpus (Garofolo et al., 1993). By means of this testing we found that the GMM that maximized accuracy was one that fit only two curves to the MFCCs and used a full covariance matrix. This yielded an accuracy of approximately 52% on this testing data (models fitting three, four, five, and six curves performing even worse). While this performance is significantly higher than chance given the sample space of 40 speakers, it is certainly not ideal for practical application and leaves much room for improvement, although we have not, as of yet, been able to determine where the approach most significantly falls short. One possible improvement for future implementation might be to normalize the volume of the recordings to some predetermined value. Since energy is a feature encoded by MFCCs and speaker files may vary greatly in this respect, this might improve performance. In terms of the interface of our application, we would hope to change the graphical representation of the GMM that is displayed at both training and testing stages, as the current one is not quite as successful in depicting the multi-dimensional Gaussians in an intuitively readable way. Despite this and the results of the accuracy testing, actual user experiences with the application, though limited, appear to be positive, the anecdotally observed performance seemingly much higher than that computed for our TIMIT testing sets. One last functionality feature that could be added in the future is a login system or similar means for users to have control over their own data once it is in our system, specifically giving them the capability to remove it if desired, since as of now the only way to remove a speaker's data from the system would be to contact one of us and request so, which could be undesirable from a privacy perspective.

References

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, 1-38.

Farrell, K. R., Mammone, R. J., & Assaleh, K. T. (1994). Speaker recognition using neural networks and conventional classifiers. *IEEE Transactions on speech and audio processing*, *2*(1), 194-205.

Garofolo, John, et al. TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1. Web Download. Philadelphia: Linguistic Data Consortium, 1993.

Reynolds, D. A., & Rose, R. C. (1995). Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE transactions on speech and audio processing*, *3*(1), 72-83.

Soong, F. K., Rosenberg, A. E., Juang, B. H., & Rabiner, L. R. (1987). Report: A vector quantization approach to speaker recognition. *AT&T technical journal*, *66*(2), 14-26.

Tisby, N. Z. (1991). On the application of mixture AR hidden Markov models to text independent speaker recognition. *IEEE Transactions on Signal Processing*, *39*(3), 563-570.

Libraries

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay. Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 12, 2825-2830 (2011) (publisher link)

John D. Hunter. Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90-95 (2007), DOI:10.1109/MCSE.2007.55 (publisher link)

Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering, 13, 22-30 (2011), DOI:10.1109/MCSE.2011.37 (publisher link)