UCLA
Dept. of Electrical and Computer Engineering
ECE214A, 2021
Project Description

Important Dates
Presentation: Wednesday, March 10th (in class)
Report: Monday, March 15th

Group Size: 3 people

Introduction

In this project, we are interested in finding a set of acoustic features and algorithms that predict whether two speech segments are uttered by the same speaker or not. Example features include pitch (f_o) and formant frequencies as you learned in class. There are several features to explore, such as subglottal resonance frequencies and voice source features.

Data

The UCLA Speaker Variability Database is a database designed to capture variability both between speakers and within a single speaker. The binary value indicating whether given token pairs are spoken by the same speaker is denoted $\mathrm{S}_{i,j}$, the *intra-speaker indication*. $\mathrm{S}_{i,j}$ = 1 if the waveforms *i* and *j* are uttered by the same speaker, and $S_{i,j} = 0$ if they are uttered by different speakers. Speech spoken by 50 males in both clean and noisy conditions (10 dB SNR babble noise) are included. The utterance is the read sentence “Help the woman get back to her feet” with about 5 repetitions of each sentence per speaker. You may use this data however you like. We suggest splitting the data into training speakers and testing speakers.

Note: The audio files provided are solely for educational purposes and may not be distributed or used outside of this project without written permission.

Project Package

a) Folder WavData containing 293 wav files per noise condition.
b) A list of all files allList.txt.
c) Two lists of training trials trainCleanList.txt and trainMultiList.txt.
d) Two lists of testing trials testCleanList.txt and testBabbleList.txt.
e) Script sample.m used to demonstrate a sample classifier training and testing setup.
f) Function fast_mbsc_fixedWinlen_tracking.m used for pitch tracking. You may or may not find this function useful for your algorithm.
g) Function compute_eer.m used to compute equal error rate.

Please look at all the individual files in the package to get familiar with the project. We also suggest you to listen to some of the provided wav files.

Wav File Naming Scheme

The files have been purposely named in a specific way so that you may set up training and testing lists on your own. The wav file 063A_0_HS05_06.wav corresponds to speaker ID 063, recording session A, sentence ID HS05, and recording number 06 during that specific session. Since we only use one specific sentence, every file contains “HS05”. You should use the speaker ID to set up training and testing lists.

Objectives

Your task is to derive a set of features and an algorithm to predict the intra-speaker indication. An example of the code is included in sample.m. In one case, you will train on clean data and test with clean and noisy data separately. In the second case, you will train on both clean and noisy data and again test with clean and noisy data separately.

Evaluation Metrics

In most applications, deciding whether two speech samples were from the same speaker is not enough. Instead, we may want a continuous metric that measures “how alike” these two speakers are. You should design a scoring method that can give high scores when the speakers are the same and low scores when the speakers are different. Some methods include likelihood ratios, dimension reduction and warping, and other various machine learning techniques. You will likely use the training data to train your scoring technique and the testing data to test your scoring methods.

False positive rate (FPR) is the error rate for which trials of different speakers are classified as the same speaker. **False negative rate (FNR)** is the error rate for which trials of the same speaker are classified as different speakers. These values change based on how you threshold your scores. A **receiver operating characteristic (ROC)** curve can give you a visual representation of the tradeoff between FPR and FNR.

Equal error rate (EER) is the percentage of error of your scoring technique when the threshold of your scoring function is set such that FPR is equal to FNR. The function compute_eer.m, which takes a list of scores and labels, has already been written for you.

Instructions

a) Download the project package from the course website.
b) Unzip it and open the folder.
c) Open and run sample.m. The script should run for about 10 minutes.

You should see an FPR of about 34% and an FNR of about 30% for clean training and testing. The code evaluates the score as the distance between the mean fundamental frequencies of the two speakers. The classifier training simply computes the EER threshold for the training data. The classifier then uses this threshold to test the system. Replace the “clean” test list with the “babble” test list in sample.m and rerun the code. The new FPR should be about 46% and the new FNR should be about 38%.

Next, replace the training list with the multi condition training list and test on clean data. The new FPR should be about 32% and the new FNR should be about 33%. Finally, do this once more for the babble test list. The new FPR should be about 43% and the new FNR should be about 42%. These numbers will serve as your baseline error rates. You should see how much your method can improve over this simple method.

Your task is to create a scoring and classification algorithm that can successfully differentiate between different speakers.

How to Modify the Code

The base script sample.m is free for you to modify as you see fit. You may also create whatever evaluations you wish. However, you should ensure your evaluations are fair. If you use a speaker to train the scoring function, be sure this speaker does not appear in testing.

Useful MATLAB Functions

imagesc	displays matrices in 2D
mesh	displays matrices in 2D
subplot	plots several figures in 1 panel
figure	plots a new figure without overriding the current one
tic/toc	captures running times
wavplay	plays wav files (make sure your speakers/headphones are on)

Oral Presentations

There will be oral presentations by the different teams describing their work. Presentations should be planned by the team as a group.

Report and Code

The report (one per group) should include:

- Introduction (what is the problem/why is it important)
- Background (literature survey)
- Project Description (features, algorithm, implementation, results, average run times, etc.)
- Summary and Discussion (also ideas for future work)
- References (cited throughout the report)

The report should be 6-pages long and have the same format as the INTERSPEECH speech conference. Figures and flowcharts generally help clarify the text.

Code should be turned in on the day of the presentation. Comments at the beginning of each function should describe what the function intends to do. You may submit only one score/classification function but any other helper files you wish. Thus, you should consider the tradeoff between doing well in clean conditions and noisy conditions. To evaluate the robustness of your system, we will use another dataset, which contains different file pairs to evaluate the intra-speaker classification accuracy. You will run your classifier on the unknown data and submit either the scores and a suggested threshold or classification labels. If you decide to submit classification labels, consider the tradeoff between FNR and FPR. The final report may be turned in by the following Monday (3/15).

You may find the following references useful.

References

Kreiman, J., Park, S. J., Keating, P. A., & Alwan, A. (2015). "The relationship between acoustic and perceived intraspeaker variability in voice quality." In *Sixteenth Annual Conference of the International Speech Communication Association*.

Nolan, F., McDougall, K., & Hudson, T. (2011). "Some acoustic correlates of perceived (dis) similarity between same-accent voices." In *Proceedings of the 17th International Congress of Phonetic Sciences, Hong Kong*, Vol. 17, No. 21, pp. 1506-1509.

Espy-Wilson, C. Y., Manocha, S., & Vishnubhotla, S. (2006). "A new set of features for text-independent speaker identification." In *Proceedings of INTERSPEECH*.

Park, S. J., Sigouin, C., Kreiman, J., Keating, P., Guo, J., Yeung, G., Kuo, F. Y., and Alwan, A. (2016). "Speaker Identity and Voice Quality: Modeling Human Responses and Automatic Speaker Recognition." In *Proceedings of INTERSPEECH*, pp 1044–1048.

Hansen, J. H. L., & Hasan, T. (2015). "Speaker Recognition by Machines and Humans: A tutorial review." *IEEE Signal Processing Magazine*, Vol. 32, No. 6, pp. 74 -99.

Reynolds, D. A. and Rose, R. C. (1995). “Robust text-independent speaker identification using Gaussian mixture speaker models.” *IEEE Transactions on Speech and Audio Processing,* Vol. 3, No. 1, pp. 72-83.

Kinnunen, T., and Li, H. (2010). “An overview of text-independent speaker recognition: from features to supervectors.” *Speech communication*, Vol. 52, No. 1, pp. 12-40.