UCLA
Dept. of Electrical and Computer Engineering
ECE214A: Digital Speech Processing
Winter 2019

Speaker Verification in case of text and/or style mismatch
Project Description

Oral presentations and evaluations are scheduled for **Wednesday, March 13th, 2019 from 10:00-11:55 a.m. in class. Online students can do the presentations by Skype if they wish or in person (please contact Ms. Afshan to arrange for a presentation time).**

I. 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 (F0) and formant frequencies as you learned in class. There are several other features to explore, such as cepstral features, subglottal resonance frequencies, and voice source features.

II. Data

The UCLA Speaker Variability Database is a database designed to capture variability both between speakers and within a single speaker.

Speech utterances by 50 females in two different styles: read speech and phone call conversation are included in the training set. The utterances in the read sentences are text-dependent i.e all speakers are speaking the same text - “The soft cushion broke the man’s fall”. The utterances in the phone-call style are text independent i.e., different texts in each utterance. In the case of a phone call recording, only one side of the conversation is recorded in the studio. The microphone and environment conditions are the same for both styles. We also provide 20 different speakers in the test set with the same structure. You may use this data however you like. All the wav files provided to you have been sampled at 22.5kHz.
Note: The audio files provided are solely for educational purposes only and may not be distributed or used outside of this project without written permission.

III. Project Package

- **A.** Folder *WavData_Style* contains 2 folders - read and phoneCall for 2 styles of speaking.
- **B.** Folder *read* contains 268 wav files. Folder *phoneCall* contains 297 wav files.
- **C.** A list of all files *allFiles.txt*.
- **D.** Two lists of training trials *train_read.txt* and *train_phone.txt*.
- **E.** Two lists of testing trials *test_read.txt, test_phone.txt* and *test_mismatch.txt*
- **F.** Script *sample.m* used to demonstrate a sample classifier training and testing setup.
- **G.** Function *fast_mbsc_fixedWinlen_tracking.m* used for pitch tracking. You may or may not find this function useful for your algorithm.
- **H.** Function *compute_eer.m* used to compute equal error rate.

In the file containing training and testing trials, the binary value indicating whether given token pairs are spoken by the same speaker is 1, and it is 0 if they are uttered by different speakers. Please look at all the individual files in the package to get familiar with the project. We also suggest that you listen to some of the provided .wav files.

The wav file names contain some information about the file, which might be helpful for your analysis. For example, the wav file 001A_0_HS08_08.wav in the read folder corresponds to speaker ID 001, recording session A, sentence ID HS08 and recording number 06 during that specific session. The wav file 001B_0_phone_02.wav in the phoneCall folder corresponds to speaker ID 001, recording session B, speaking over the phone.

IV. Baseline

As indicated earlier, you have been provided with a sample.m file that serves as a baseline for the project. The baseline uses the absolute difference of pitch between the pair of utterances being compared as the feature to the classifier. In the baseline, we use k-nearest neighbors as the classifier.

V. 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 read-read pairs and test with 1) read-read, 2) phone call - phone call and 3) read - phone call. In the second case, you will train on phone call-phone call pairs and again test with 1) read-read, 2) phone call - phone call, and 3) read-phone call separately.

VI. 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.

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

VII. Instructions

a) Download the project package from the course website.
b) Unzip it and open the folder.
c) Open and run sample.m. The baseline script should run for about 10-15 minutes.

You should see that the EER, in this case, is about 33.74%. Next, replace the training list and the testing list to verify all the below-given combinations. The following table will serve as baseline error rates.

	EER (in %)		
Train/Test	**Read-Read**	**Phone Call-Phone Call**	**Read-Phone Call**
Read-Read	33.7398	51.345	47.0294
Phone Call-Phone Call	36.748	51.6667	47.4074

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 with/without text mismatch and style mismatch.

VIII. 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. Please do not change/modify the test trial files. This is to ensure your evaluations are fair. Note that the run time of the code may increase when you use other sophisticated features and/or classifiers.

IX. Oral Presentations

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

X. 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.

The 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 tex and/or style matched conditions and text and/or style mismatch conditions. To evaluate the robustness of your system, we will use speech from a different set of unseen speakers to evaluate the speaker verification performance. You will run your classifier on the unknown data and submit either the scores and a suggested threshold or classification labels. The final report may be turned in by the following **Monday (03/18).**

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.