EE599 Speech Processing Spring 2019
Project 2 Specification
Due date February 28, 2019

Any common programming language may be used for this project, but Matlab is recommended.
Turn in all code and specified plots for the problems given.

This project will focus on isolated word recognition using a Dynamic Time Warping approach.

DTW algorithm details:

- Assume starting point in grid = (1, 1) and ending point = (I, J)
- Allow horizontal, vertical, and diagonal motion only, with weights = 1,1,2, respectively
- Feature extraction: Calculate 12 MFCCs for each frame using the "v_melcepst" command from the Matlab Voicebox toolbox. Use 256-point frames, with default parameters for window (Hamming), overlap (50%), mel-scale window shape (triangular), filter-bank type (absolute magnitude), number of cepstra (12), and low and high frequencies (0 and fs/2). Include the 0^{th} order cepstral coefficient, and do not include energy or delta or delta-deltas.
  Note: If you want, you can compute feature matrices one time for all waveforms and then use these as the data set directly.
- Distance measure: Define frame-to-frame distance as the weighted Euclidean distance, with the weighting matrix equal to the inverse covariance matrix over the entire data set.
- DTW implementation: Use a memoization approach to implement your DTW algorithm to find the minimum aligned distance between any two feature matrices. Specifically, this means your code will set up an IxJ matrix of local frame-to-frame distances and compute these, then set up a second IxJ matrix of global distances and fill these in using either row-wise or column-wise looping starting at (1,1) and working up to (I,J). Your DTW function should take as input two feature matrices and return as output the minimum aligned distance between the two.
  Graduate students: Students taking this course for graduate credit should implement their function to also return a Kx2 matrix that indicates the alignment path chosen, where K is the number of steps in the path and each row indicates the (i,j) index of that step in the path.

Dataset: We will be using a subset of the SUSAS (Speech under Simulated and Actual Stress) described in more detail in https://catalog.ldc.upenn.edu/LDC99S78 . The simulated stress portion of the dataset includes recordings of 35 words from 9 different speakers (three different regional accents) in 12 different simulated stress conditions. The vocabulary set that we will use includes the 5 words {Break, Eight, Eighty, Destination, and Zero}, taken from 3 of the speakers {General1,2,3} in 4 conditions {Train=normal, Fast, Slow, Soft}. The data files have been renamed to indicate word, speaker, and condition, converted to .wav format, and made available as a single zip download from Canvas.

You will select a single template for each of the five words, based on whichever single example from the "normal" condition has length closest to the mean length of all "normal" examples. These five examples will be used as the "training set", and all other 265 examples will be considered as the "test set". Run your DTW code to compare all files in the test set against each of the five templates in the training set. Select as the recognizer output whichever template gives the lowest DTW-calculated distance.

Display the results of your classifier as a confusion matrix. In a confusion matrix, the rows indicate the true category of the test example, and the column indicates how that example was classified. You can create one easily – each time you run a classification, just increment the corresponding element of the confusion matrix to keep a count. Once all examples are run, normalize the confusion matrix to a percentage by dividing by the sum over the row.

Undergrad students: Compute and present a single 5x5 confusion matrix showing the overall results
Graduate students: Students taking the course for graduate credit should also keep track of the classification results as a function of the speaker and the condition. Results should include an overall confusion matrix as well as individual confusion matrices for each of the 3 speakers and each of the 4 conditions (but not for sub-combinations of these, because there won't be sufficient data to do so). You should also present an example of the DTW alignment path for aligning a "fast" and "slow" example against the corresponding correct template.

Deliverables: Turn in your DTW code, your script for running the recognition experiments, and a report that briefly summarizes the experiment, gives the results as well as a discussion of those results.