J1799d Fall 2014 Project 2 Due date: Sep 19th 2014

Write a routine for computing MFCC from audio

- Record multiple instances of digits
  - Zero, One, Two etc.
  - 16Khz sampling, 16 bit PCM
  - Compute log spectra and cepstra
    - Use 40 Mel spectral filters. They must cover the frequencies between 133.33Hz and 6855.4976Hz (you may use a different setting if you choose).
    - No. of features = 13 for cepstra (use first 13 DCT coefficients)
  - Visualize both spectrographically (easy using matlab)
    - Note similarity in different instances of the same word
  - Modify number of filters to 30 and 25 (over the same frequency range).
    - Patterns will remain, but be more blurry

Some suggestions

You are allowed to refer to other people's code and implement it by yourself.

- Dan Ellis has nice matlab code on his website.

  http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/

- The "wav2feat" code in CMU sphinx is good.
  - wav2feat.c, fe_sigproc.c, etc

However, we recommend doing your own code if you can.

Regardless of what you use, the feature computation code must be integrated with the audio capture routine.

- Assume keyboard hit for start of recording. Stop of recording is obtained via automatic endpointing.

How to visualize the spectrogram represented by cepstra

The Mel-log spectrum can be directly visualized as a matrix.

However, the cepstrum is a dimensionality-reduced and *transformed* version of the log spectrum. It is not visually meaningful. However, the truncated cepstrum can be converted *back* to a log spectrum by zeropadding it to 64 or 128 poitns and computing an *inverse* DCT (if you used a DCT to derive cepstra from log spectra). The IDCT-derived logspectrum is what the cepstrum really represents.

You can use matlab to visualize the IDCT-derived logspectrum offline.