Assignment 3: Perceptual Features Extraction

CS 4347: Sound and Music Computing

due Wednesday 22 February 2017, 11:59pm

0. This assignment will use the same "music / speech" dataset that we used in assignments 1–2.

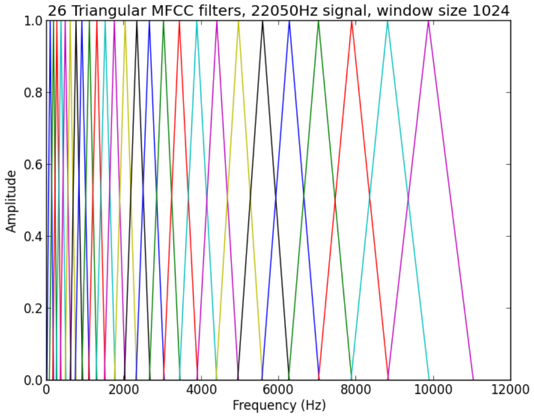
1. Write a program that:
   - Reads the ground truth `music_speech.mf` file.
   - Loads each wav file and splits the data into buffers of length 1024 with 50% overlap. Only include complete buffers; if the final buffer has 1020 samples, omit that buffer.
   - Calculates the MFCCs for each window as specified in the lecture notes. A few more details:
     - Given input $x(t)$ and output $y(t)$, the pre-emphasis filter should be

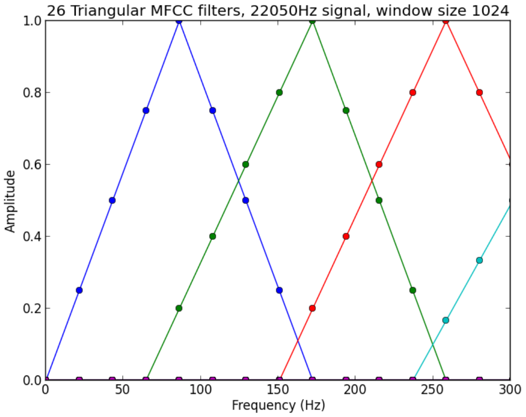
       $$y(t) = x(t) - 0.95x(t-1)$$

     - Use a Hamming window before the mag-spectrum calculation
     - Mel-scale of frequency f is:

       $$Mel(f) = 1127 \ln(1 + \frac{f}{700})$$

     - Calculate 26 mel-frequency filters, covering the entire frequency range (from 0 Hz to the Nyquist limit). To calculate the filters,
       * find the X-axis points of the filters (left side, top, right side). All points must be convereted into integer FFT bins; the left side should use the `floor()` operation; the top point should use `round()`; the right point should use `ceil()`.
       * assign the left bin to be 0, top bin to be 1.0, right bin to be 0; linearly interpolate between the rest
     - the log step should be log base 10.
     - scipy has DCT built-in: `scipy.fftpack.dct()`
     - do not calculate any delta-features
   - Calculates the mean and standard deviation for each MFCC bin over the entire file. So if there are M MFCC bins each each buffer, you will end up with a feature vector of length $2M$ for each song.
   - Writes the data to an arff file (each line should contain the 26 means, followed by the 26 standard deviations, and finally the class).
   - Make two plots: the overall range of the triangular windows, and the triangular windows from 0 to 300 Hz. They should match the examples below.





2. Upload your 2 PNGs, ARFF file, and source code to:

   `http://cs4347.smcnus.org`

   This will automatically grade the values you calculated. If any mistake is found, please check your program and resubmit – you are welcome to submit as many versions as you wish before the submission deadline.

   Submit a zip file containing your program's source code (as a `.py`), the ARFF file, 2 PNGs, and an optional README.txt file to the same website.

   - You may use anything in the python standard library, numpy (including pylab / matplotlib), and scipy libraries. No other libraries are permitted.

If you are familiar with python and understood the lecture, this should take 2–3 hours.
Grading scheme:

- **3/6 marks**: 2 correct PNGs, and correct ARFF file (automatically graded by computer).
- **3/6 marks**: readable source code (good variable names, clean functions, comments when needed).