Assignment 5: Perceptual audio features

CS 4347: Sound and Music Computing

due 11 March 2015, 11:59pm SGT

0. This assignment will make use of the "Music Speech" dataset of Marsyas we used for assignments 1–4.

   `http://opihi.cs.uvic.ca/sound/music_speech.tar.gz`

1. Write a program that:
   - Reads a collection file (format: filename \t (tab) label).
   - Load each wav file and split 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

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

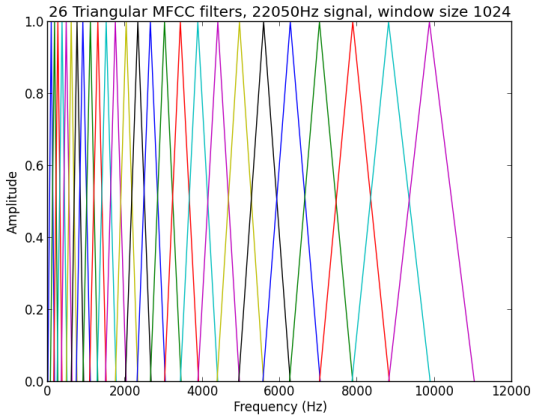
     - Use a Hamming window before the mag-spectrum calculation
     - Mel-scale of frequency f is:

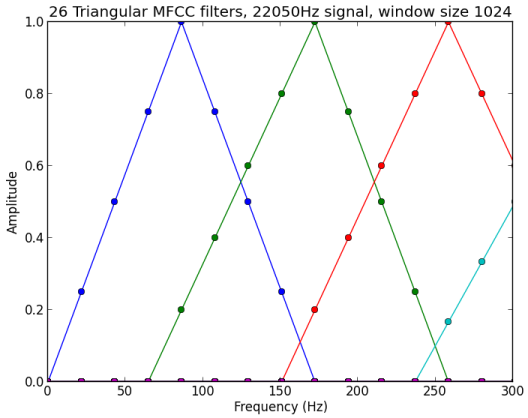
       $$Mel(f) = 1127 \ln(1 + \frac{f}{700})$$

     - 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.
   - Classify the data with `trees.J48` with 10-fold cross-validation and save the results. In addition, select 2 other classification algorithms and record their output. Write a file called `classifications.txt` which compares the results of these three algorithms.

The file should begin with 1-3 sentences stating which was the best algorithm and summarizing its performance in comparison to the other two algorithms. This should be followed by the Weka results ordered from best to worst.

- 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. Submit: the 2 PNGs, the ARFF file, the discussion of weka output, and your source code.

Grading scheme:

- **2/6 marks**: PNG files showing the triangular windows.
- **2/6 marks**: Discussion of weka output.
- **2/6 marks**: readable source code (good variable names, clean functions, comments when needed).