Assignment 1: Dealing with Time-Domain Audio

CS 4347: Sound and Music Computing

due Wednesday 4 February 2015, 11:59 pm

0. This assignment will make use of the "Music & Speech" dataset of Marsyas:
   - `http://opihi.cs.uvic.ca/sound/music_speech.tar.gz`
   - This dataset has two copies of each song; delete the `music/` and `speech/` directories and use the files in `music-wav/` and `speech-wav/` directories. There are 64 music and 64 speech files; each file is 30 seconds of audio stored as 16-bit signed integers, 22050 Hz.
   - Ground truth data for this dataset:
     `http://www.comp.nus.edu.sg/~duanzy/music_speech.mf`
     Format of the file is `filename` `\t` (tab) `label` `\n` (newline), one song per line:

     ```
     filename1\tlabel1\n
     filename2\tlabel2\n
     ...
     filename128\tlabel128\n
     ```

     The `label` will be `music` or `speech`.

1. Write a python program that will:
   - Read the ground-truth `music_speech.mf` file
   - Load each wav file and convert the data to floats by dividing the samples by 32768.0. Hint: use `scipy.io.wavfile.read()`
   - Calculate 4 features for each file according to the given formulae. Use only one vector per file (don't use multiple buffers for each file).
     Given $X = \{x_0, x_1, x_2, \ldots x_{N-1}\}$,
     (a) Root-mean-squared (RMS)

$$X_{\text{RMS}} = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1} x_i^2}$$

     (b) Peak-to-average-ratio (PAR)

$$X_{\text{PAR}} = \frac{\arg\max_i |x_i|}{X_{\text{RMS}}}$$

     (c) Zero crossings (ZCR)

$$X_{\text{ZCR}} = \frac{1}{N-1}\sum_{i=1}^{N-1} \begin{cases} 1 & \text{if } (x_i \cdot x_{i-1}) < 0 \\ 0 & \text{else} \end{cases}$$

     (d) Median absolute deviation (MAD)

$$X_{\text{MAD}} = \underset{i}{\text{median}}\left(\left|x_i - \underset{j}{\text{median}}(x_j)\right|\right)$$

     Hint: numpy has a built-in `numpy.median()` function!

- Output the data to a comma separated value (CSV) text file in the format:

  ```
  filename1,RMS1,PAR1,ZCR1,MAD1\n
  filename2,RMS2,PAR2,ZCR2,MAD2\n
  ...
  filename128,RMS128,PAR128,AC128,MAD128\n
  ```

  Concretely, the beginning and ending of the file should be:

  ```
  music_wav/bagpipe.wav,0.063492,8.149929,0.191660,0.031769
  music_wav/ballad.wav,0.029699,7.320233,0.039395,0.012695
  ...
  speech_wav/voice.wav,0.070688,4.163124,0.082435,0.031982
  ```

  To pass our automated grading system, the format of your file must match this exactly. The order of filenames must match the order in the `music_speech.mf` file.

2. Upload your CSV file 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` file), the CSV file, 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 about 1 hour. Grading scheme:

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