CMSC5707 Assignment 1, submission deadline: 23:59, 16 Oct 2022

(Late submission penalty: 15% marks reduction of this assignment for every 24 hours)
Please zip all into one zip file. Submit your zip file to the CUHK Blackboard system (https://blackboard.cuhk.edu.hk/)
and follow the naming format: [Asg-X][SID][Name].zip
For example, [Asg-1][1234567890][Chan Tai Man].zip

Programming Question (50% of assignment 1)

Build a speech recognition system (total 100 points)

In part 1-4, use MATLAB, Python, or any language you prefer. Write you own code. **Do not use any signal processing library functions** such as FFT or autocorrelation in this part. But string/file manipulation functions are allowed.

In part5, you may use any Matlab/Octave/python library functions you prefer.

CUHK has now provided licenses for all students to install Matlab in their own computers. You can check the details on ITSC website at https://www.itsc.cuhk.edu.hk/all-it/procurement-support/campus-wide-software/matlab-and-simulink/. In addition, you can use the Matlab online at https://matlab.mathworks.com/.

If you choose to use Python instead, please attach a README file containing package dependencies and instructions together with the submission.

1. (5 points) Recording of the template data as set A

   Use your own sound recording device (e.g. mobile phone, Windows Sound Recorder, https://www.goldwave.com/ software etc.) to record the 6 different numbers of your student ID (If you student ID is 1155034567, you may record 0,1,3,4,5,6,7 from small to large value) and name these files as s1A.wav, s2A.wav, etc. You are free to choose any 6 different numbers which exist in your ID. Each word should last for around 0.7 seconds (depending on your speaking speed, some variation, e.g. a deviation of +/- 0.2 seconds is acceptable) and you can use http://audio.online-convert.com/convert-to-wav to convert your file to .wav if necessary. You may use any spoken language (Cantonese/Mandarin/English etc.) to read out those numbers/words. (Hint: remember to record the speech in a quiet environment, it will give you much better result)

   These files are called set **A** as the templates of our speech recognition system. You may use any sampling rate (Fs) and data width. However, the typical values are Fs = 44100 Hz (or Lower) and data width = 16 bits. Be careful that your .wav file may be in stereo format; choose the left or right channel (column 1 or column 2) of the matrix representing your data in MATLAB during processing.

   To submit: wav files.

2. (5 points) Recording for testing data as set B

Repeat the above recording procedures of the same numbers, and save the files as s1B.wav, s2B.wav, etc. They are used as the testing data in our speech recognition system.

To submit: wav files.

3. (5 points) Plotting
   a. Select one of your recorded numbers/words (in set A or B) as x.wav, use the computer to load the file and plot the time domain signals.
   HINT: You may use "audioread" and "plot" in MATLAB or OCTAVE. Type "help audioread" / "help plot" in MATLAB to learn how to use them.
   To do and submit: Plot "x.wav" and save it in a picture file "x.jpg", submit x.jpg.
4. (35 points) Signal Analysis
   a. From "x.wav" of the previous question, write a program to find the start (T1) and stop (T2) locations in time (ms) of your recorded sounds automatically. Depending on the environment when you are recording your sounds, you need to alter the parameters (number of zero crossing and energy level thresholds etc) of your system to obtain a reasonable T1 and T2 time locations. See the end point detection algorithm in chapter 5 (http://www.cse.cuhk.edu.hk/~khwong/www2/cmsc5707/5707_05_speech_rec.pptx) of my lecture notes.
   To do and submit: Plot x.wav as x_wav.jpg and show T1 and T2 as vertical bars in the figure. Submit the figure.
   b. Extract one segment called **Seg1** (20 ms duration of your choice of time location) inside the voiced vowel part (the part with a repetitive waveform not noise) of "x.wav" between T1 and T2. Usually, the voiced part is where you are uttering a vowel and usually has higher energy. And **Seg1** can be saved as an array/vector in MATLAB/OCTAVE/python. You may choose the segment manually by inspection and hardcode the locations in your program.
   To do and submit: Mark the starting and ending position of the voiced part in the diagram x_wav.jpg. Submit the picture.
   c. Find and plot the Fourier Transform (Energy VS Frequency) of **Seg1**. The energy is equal to $\left|\sqrt{real^2 + imaginary^2}\right|$. Write your Fourier transform code without using the FFT library. The horizontal axis is frequency, and the vertical axis is energy. Label the axes of the plot.
   To do and submit: Submit your code in a file called fourier.txt, and submit the plot as fourier_x.jpg.
   d. Find the pre-emphasis signal (**Pem_Seg1**) of **Seg1** if the pre-emphasis constant is 0.95. Write your own code for emphasis.
   To do and submit: Submit your code in a file called pre-em.txt. Save the original segment **Seg1** (as the top part) and **Pem_Seg1** (as the bottom part) in a picture called Pem_x.jpg. Submit the picture.
   e. Find the LPC-10 parameters (the order of LPC for **Pem_seg1** is 10). You should write your autocorrelation code, but you may use the inverse function (inv) in MATLAB/OCTAVE to solve the linear matrix problem.

To do and submit: Submit the code, and the LPC data obtained in a file called lpc10.txt.

5. (50 points) Build a speech recognition system

In this part, you may use any Matlab/Octave/python library functions you prefer. Use the tool at http://www.mathworks.com/matlabcentral/fileexchange/32849-htk-mfcc-matlab to extract the MFCC parameters (Mel-frequency cepstrum) from your sound files. Each sound file (.wav) will give one set of MFCC parameters. See "A tutorial of using the htk-mfcc tool" in the http://www.cse.cuhk.edu.hk/~khwong/www2/cmsc5707/A_tutorial_of_using_the_htk_tool.docx of how to extract MFCC parameters. You may also use python MFCC extraction tools found on the web. Build a dynamic programming DP-based numeral speech recognition system. Use set **A** as templates and set **B** as testing inputs. You may follow the following steps to complete your assignment.

a. Convert sound files in set **A** and set **B** into MFCCs parameters, so each sound file will give an MFCC matrix of size 13 × 70 (number of MFCCs x = 13, and number of frame segments = 70). Because if the time shift is 10ms (frame non-overlapping region), a 0.7 seconds sound will have approximately 70 frame segments, and there are 13 MFCC parameters per frame. Here we use $M(j,t)$, to represent the MFCC parameters, where j is the index for MFCC parameters ranging from 1 to 13, t is the time index for the time segment ranging from 1 to 70. Therefore a (13-parameter) sound segment at time index t is $M(1{:}\,13,t)$.

b. Assume we have two short time segments (e.g. 25ms each), one from the tth (t = 28) segment of sound X represented by 13 MFCCs parameters $M_X(1{:}\,13,t=28)$, and another from the t'th (t' = 32) time segment of sound Y represented by MFCCs parameters $M_Y(1{:}\,13,t'=32)$. The distortion (dist) between these two segments is

$$dist=\sqrt{\sum_{j=2}^{13}[M_X(j,t)-M_Y(j,t')]^2}=\sqrt{\sum_{j=2}^{13}[M_X(j,t=28)-M_Y(j,t'=32)]^2}$$

Note: The first row of the MFCC ($M(j=1,t)$) matrix (also known as MFCC index 0) is the energy term and is not recommended to be used in the comparison procedures because it does not contain the relevant spectral information. Therefore, summation starts from j = 2. Use dynamic programming to find the minimum accumulated distance (minimum accumulated score) between sound X and Y.

c. Build a speech recognition system: You should show an n × n Confusion matrix-table (https://en.wikipedia.org/wiki/Confusion_matrix) as the result, where n is the number of recordings. If your result is correct, we expect you see low-distortion scores along the diagonal of the Confusion matrix-table.

d. You may use the above steps to find the minimum accumulated distance (or distortion) for a sound pair (there should be n pairs, as there are n sound files in set

A and n sound files in set **B**. You just need to select one pair) and draw the path by highlighting the cells in the accumulated distance score (or cost) matrix diagram manually by hand (or by a program).

Procedure: For example, pick any one sound file from set **A** (e.g. the sound of "one") and the corresponding sound file from set **B** (e.g. the sound of "one"), compare these two files using dynamic programming , **plot the optimal path by highlighting the cells in the accumulated distance score matrix diagram manually by hand (or by a program)**.

To do and submit: program code and the n × n Confusion matrix-table of the speech recognition system in part 5.

A check list of what to submit:

1. All your programs with a readme file showing how to run them.
2. All sound files of your recordings.
3. All picture files and data required in the questions.
4. The program code and the n × n Confusion matrix-table of the speech recognition system in part 5.
5. Zip all files into one zip file using the naming format as shown on page 1, and submit to CUHK Blackboard before the deadline.