Instructions: This assignment is intended to introduce you to signal processing to the extent that it is used for typical speech recognition methods. Additionally, the quality of these features will be tested with the implementation of one of the oldest methods for speech recognition: Dynamic Time Warping (DTW). Please ask for help if you feel stuck!
Please submit answers to all underlined questions below in the writeup as a PDF file via the course Canvas site. The tex file used to create this PDF has also been provided so that you can enter your answers directly into this document. Additionally, please run through and execute all of the cells of the ipython notebook (including the parts you added which are marked by `# TODO`) and submit the notebook via Canvas.
Lateness policy: Late submissions submitted up to Thursday 4/30/20 7:00 pm (the "late deadline") will receive a 20% reduction in credit. Submissions that are later than this will receive some non-zero credit, but we cannot guarantee how much. We cannot guarantee that we will carefully grade or give feedback on submissions after the "late deadline". In addition, you have four free "late days" that you can use throughout the term to extend homework (not project) deadlines without penalty. If you are using any of your "late days" for this assignment, mention that in the comment while submitting this assignment. You should state how many "late days" you are using and how many remaining "late days" you have. The number of late days used for any given homework must be an integer.
Collaboration: You are encouraged to discuss assignments and any aspect of the course material with others, but any material you submit (writeup, code, figures) should be produced on your own.

*** Before starting this homework, you'll need to make sure you have python3 and the following packages on your system: `numpy, scipy, jupyter, matplotlib, pysoundfile`. You may use any installer of your choice. Alternatively, here are the steps you can follow to create a python3 environment and install the required packages using Miniconda.

(a) Install Python 3.7 using miniconda.

- Linux users:
  i. `wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh`
  ii. `bash Miniconda3-latest-Linux-x86_64.sh`
- Mac users:
  i. `wget https://repo.anaconda.com/miniconda/Miniconda3-latest-MacOSX-x86_64.sh`
  ii. `bash Miniconda3-latest-MacOSX-x86_64.sh`

(b) Create and activate a python3 conda environment

i. `source ~/miniconda3/bin/activate` (the appropriate path to the miniconda installation)
ii. `conda create -n new-py3-env python=3` (feel free to use an environment name of your choice in place of `new-py3-env`)
iii. `conda activate new-py3-env`

(c) Install the required packages after activating the environment

- For numerical computing facilities in python: `conda install numpy scipy`
- For using jupyter notebooks with python: `conda install jupyter`
- For the plotting functionality in python: `conda install matplotlib`
- For using the `soundfile` library within python: `pip install pysoundfile`

(d) Additional downloads from Canvas: `hw2.ipynb` notebook file, `aurora_FMS_15739A.wav`, zipped data: `tidigits.zip` from canvas.

Once everything is set up, you can open a notebook with the command: `jupyter notebook hw2.ipynb`. Make sure that the kernel is set to Python 3 in the notebook. Note that you need to activate the environment (if not already activated) before opening the notebook. Use the commands b.(i) followed by b.(iii) to activate. You can deactivate the environment using the `conda deactivate` command. Please let us know if you need help!

0. Please fill out this questionnaire. This is purely to help us calibrate assignment load and let us know what may need to be made clearer in class. Your responses will receive a small amount of credit, independent of the actual answers.

   (i) Did you collaborate on this assignment, and if so, with whom?
   Yes, with Hanqi Zhang

(ii) Approximately how many hours did this assignment take to complete?
12 hours approximately

(iii) On a scale of 1 to 5 (where 1 = trivial and 5 = impossible), what was the difficulty of this assignment?
4

(iv) On a scale of 1 to 5 (where 1 = useless and 5 = essential), how useful has this assignment been to your understanding of the material?
4

1. Mel-frequency Cepstral Coefficients (MFCCs)

   (a) This part allows you to experiment with MFCCs, and introduces you to code to perform feature extraction in python.

   (b) Follow the instructions given in Part 1 of the `hw2.ipynb` and answer the questions below.

   - What does the first cepstral coefficient $c[0]$ represent? (i.e. can you describe it in words, by examining the equation for computing the cepstrum given in the lecture slides) Optionally (for extra credit): What about the second cepstral coefficient $c[1]$?

     By looking at the formula:
     $C[0] = \frac{1}{N} \sum_{k=0}^{N-1} X_{mel}[k] * 1$
     where $X_{mel}[k]$ is just the log magnitude of the signal(after mel warping). So C[0] is just the energy or power of the voice signal.

   - How does the recovered signal compare to the original recording? Does the difference make sense? Repeat for smaller numbers of cepstral coefficients by lowering 'ncoeffs'. What's the smallest number such that the re-synthesized utterance is still intelligible?

     The recovered signal lose some detail information especially from the vocal cords. And this make sense because the cepstral features are computed by taking the Fourier transform of the warped logarithmic spectrum.And the formant filtering is seperated with the low frequency on using cepstral features. By only taking the first 13(or more)features, it will capture the majority of the filter detail though it sounds like a 'robot'.

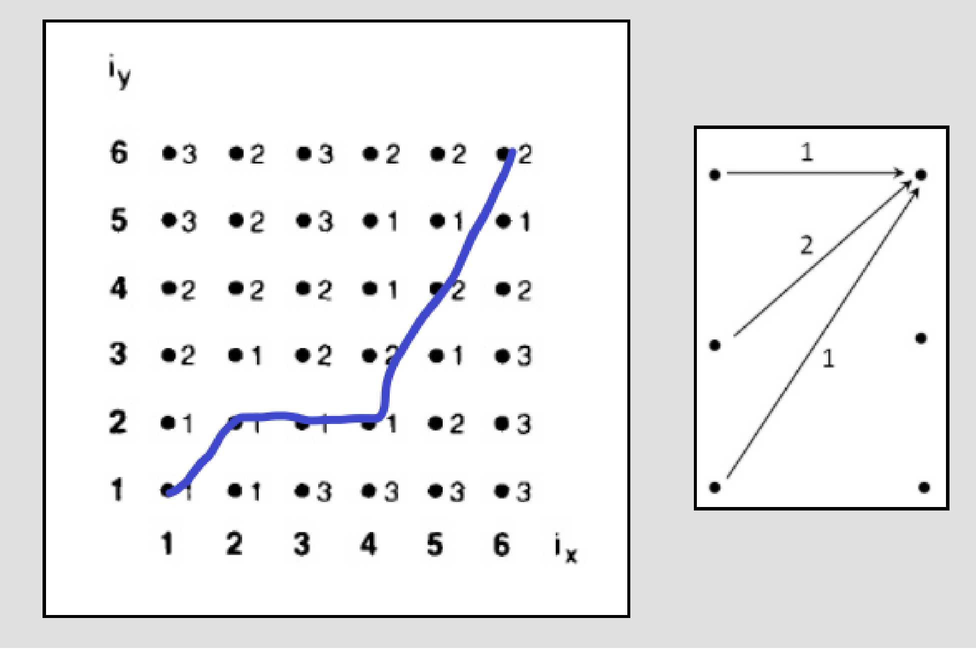
     According to my experiment, I could use 3-5 number of cepstral coefficients and the recovered signal is still intelligible.

   - Which types of speech sounds do you expect to be better represented by long-window MFCCs and which by short-window MFCCs? For purposes of this question, short-window MFCCs are ones generated with a window size of 64 samples, and long-window MFCCs are generated with a 1024-sample window. The long-window version is the standard one used in speech recognizers. Optionally (for extra credit): Can you suggest a way of using this information to improve the performance of a speech recognizer?

     The window should not be longer if it break the relative stationary of the underlying process. And it should not be shorter if it unable to capture the low frequencies.
     When analyzing signal, we should consider small window because speech signals behave as quasi stationary in short period and vowels,fricatives will be more distinguishable with shorter window.
     However, longer window will emphasize more on the entity. It it better at capturing the features of the language. For example the emotion or gender or other features.
     Therefore, I think we should use long window when predict speakers identity while using short window when predict the words.

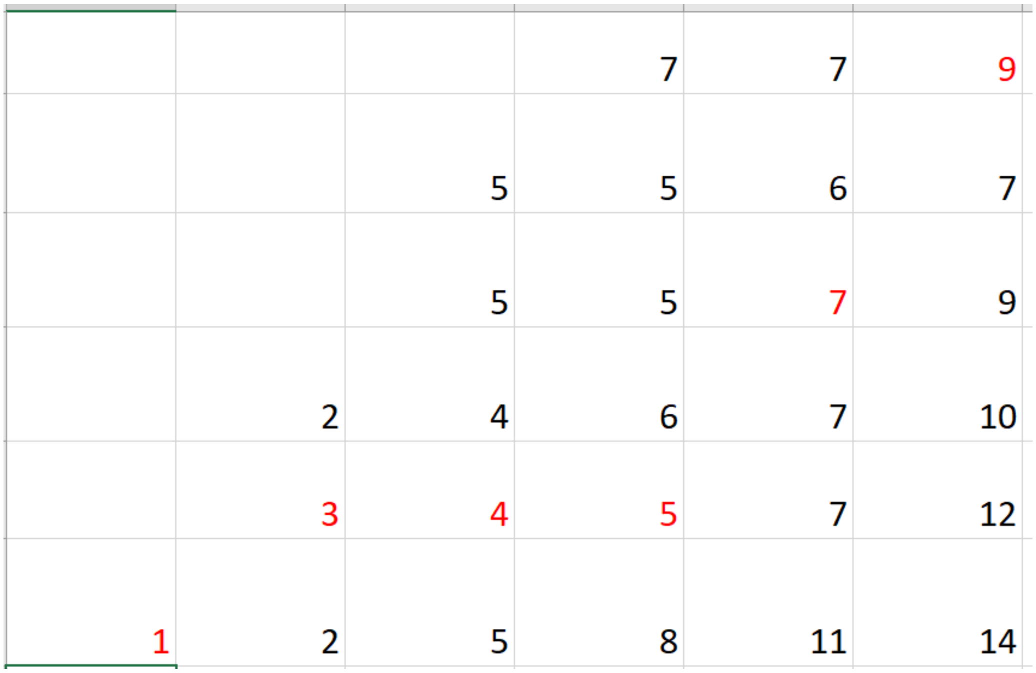
2. Dynamic Time Warping (DTW):

   (a) Complete the DTW exercise shown below using the following move set. Find the minimum distance, and the corresponding path, through the grid of frame distances using the move set below. Submit the marked-up grid with the best path (or one of the best paths, in case of ties) highlighted and minimum distance marked.



i_y						
6	3	2	3	2	2	2
5	3	2	3	1	1	1
4	2	2	2	1	2	2
3	2	1	2	2	1	3
2	1	1	1	1	2	3
1	1	1	3	3	3	3
	1	2	3	4	5	6 i_x

DP table or total distance table:



			7	7	9
		5	5	6	7
		5	5	7	9
	2	4	6	7	10
	3	4	5	7	12
1	2	5	8	11	14

The minimum distance is 8.

(b) For this part you will use a basic DTW-based recognizer to do single digit recognition. You will use a data set consisting of 100 training utterances and 100 test utterances of isolated digits (10 each of 0-9, where ”0” is always pronounced ”oh” and not ”zero”). The data and scripts are to be downloaded from the canvas site.

i. Finish the implementation of the ‘dtw’ function by implementing the recursion step (marked `# TODO`) for the move set introduced in class:

See the code in notebook.

ii. Additionally, experiment with (at least) one of the following:

- altering the ’dist’ function
- changing the move set

- increasing the number of templates
- coming up with your own extension!

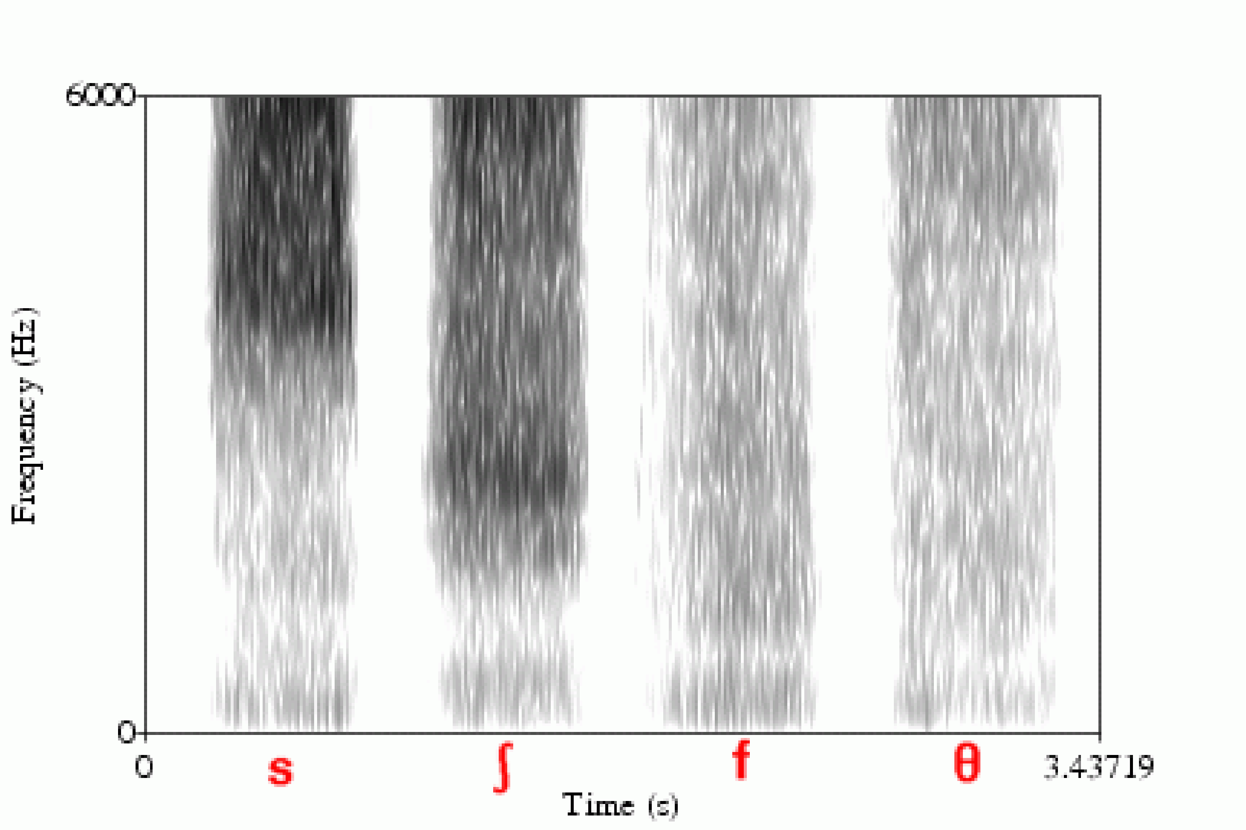
iii. Describe in 1-2 paragraphs what you did, the results you obtained, and any other comments about your experiments (e.g. the trade-off between performance and efficiency, types of errors, etc.).

Extra credit for the best DTW-recognizer. Note that this is a "cheating" experiment in that we are tuning the recognizer on the test set. In a "real" experiment, we would have a separate development set for tuning, and only test on the test set after fixing the recognizer based on development set performance.)

What I did is change the distance metric to Mahalanobis metric, which I comment out in the code but include a confusion plot at the bottom. The accuracy is not improved unfortunately.
One possibility that the Euclidean distance between pairs that ignore the data regularities. Therefore, it is not optimal in high dimensional space, especially when we want to use more features here. So I try a distance metric called Mahalanobis distancewhich is also commonly used in machine learning problem. Since it accounts for the correlation between features.

The accuracy of using Euclidean distance is 94 percent. While it decrease to 64 percent after changing to Mahalanobis distance. I think the main reason is that the 13 features are actually using the same unit(or metric) and also 13 is not really a high dimension vectors.

3. When spelling a word over a (land-line) telephone, we often find ourselves disambiguating letters; e.g., ”S as in Sam, F as in Frank, B as in boy, D as in dog...”. This is more necessary to disambiguate some letter pairs than others. S and F are a particularly difficult pair. (Note that we don't need to do this when talking face-to-face, even when not looking at our interlocutor.)
Why is it necessary to do this disambiguation, and why are S and F particularly difficult?



The range of human voice is usually between 80Hz to 14000 Hz. While the range from telephone is narrower, usually between 300 Hz to 3000Hz.

It can be seen from the spectrum that S and F are very similar in the range of 80Hz to 3000Hz, therefore, we cannot distinguish between them.

4. When listening to high-pitched singing, it can be harder to distinguish vowels than in low-pitched singing. (For our purposes, we can think about singing as identical to speaking but at specific pitches dictated by the musical score.) For example, if a singer produces an [iy] and an [aa] on a "high C", the two vowels may be indistinguishable. Note that the fundamental frequency of a "high C" is close to 1000Hz.
   Give one explanation for this effect. Note: The spectrum of a vowel at fundamental frequency F will consist roughly of very high values at multiples of F and almost 0 at other frequencies.

   When the fundamental frequency is low, we will have a lot of densely harmonic on the lower range. And the length of the wave in spectrum is smaller. On the spectrogram, it is eaiser to distinguish with different formants.

   While if the fundamental frequency is high, the harmonic becomes multiples of 1000Hz, say 1000Hz, 2000Hz. This makes us difficult to differentiate formants, which is the major difference between vowels.

5. Speech recognizers often (but by no means always) perform better on male voices than on female voices. There are many possible reasons for this, some of them having to do with the different typical vocal tract properties of men and women. (Optional, extra credit) Explain how male vs. female vocal tract properties may cause this effect.

   From what we are told on the class, Man has broader vocal tract, while the range of frequency is narrower. But for woman, the frequency of voice is broader, which makes the recognition harder for similar reason as problem 3. Therefore, the first reason might be the range of frequency or the shape of vocal tract.

   There is some pitch difference between man and woman. Man tends to have a lower pitch than women. Based on this, early in 2001, W. H. Abdulla N. K. Kasabov build separate recognition system that use pitch to differentiate gender first and then pass data into separate learning model(W. H. Abdulla N. K. Kasabov,2001). Now this model has trivial difference between genders. It seems the pitch difference will have effect on the accuracy.

   However, another paper indicates that another major reason is because of the imbalance of the training data.(Tatman,2017) A lot of training data has more man voice compared to woman but they did not balance the data according to gender. So this is another possible reason.

References

[1] Tatman, Rachael (2017), "Gender and Dialect Bias in YouTube's Automatic Captions", *The Proceedings of the First ACL Workshop on Ethics in Natural Language Processing Book*, Association for Computational Linguistics.p.53-59.

[2] W. H. Abdulla N. K. Kasabov (2001), Improving speech recognition performance through gender separation, *In Proceedings of ANNES, pages 218–222Dunedin–New Zealand*,page 218-222