EEL 6935/ COT 6930 Signal Processing for Machine Learning

Project 3

Due November 6, 2022

References:

1. item Speech modeling and processing: Introduction to Digital Speech Processing by Rabiner and Schafer.
2. Cepstrum: Digital Signal Processing by Oppenheim and Schafer.
3. Paper " Wavelets and Filterbanks" by Vetterli and Herley. IEEE Transactions on Signal Processing, Vol 40, No 9, September 1992, 2207-2231.
4. *Data Driven Science and Engineering* by S. Brunton and N. Kutz. I bought the Kindle version of the book and will use parts of it. Also, look up many wonderful lectures by Stephen Brunton on YouTube.

The overarching goal of this project is to classify speech segments as voiced and unvoiced. You will perform this task using traditional signal processing methods and a neural network. Among the signal processing methods, you will use are time domain methods as described in Introduction to Digital Speech Processing, AR modeling, aka Linear Predictive Coding, cepstrum, wavelets and filter banks, and sparse representations. We have already discussed AR modeling. I will introduce the others in the lectures October 18,20,25. I will discuss them further in the other lectures.

You may use any speech file you like to test your program. I will upload some on Canvas. It is best if you start with a well recorded, noise free recording. Recommended parameters are 16KHz sampling rate and a duration of 3-4 seconds. Use a .wav file for lossless coding.

1. Read an audio file $\mathbf{x}$ and perform preliminary exploration of it.
   - Plot the signal versus time.
   - Divide the signal into frames that are approximately the duration of 5 to 10 times the speaker's pitch period. Frame overlap should be at least 50%. Arrange the frames as the columns of a matrix $\mathbf{X}$.
   - Evaluate the DFT of each frame by using MATLAB's **fX=fft(x,nfft)**.

- Plot the spectral magnitude and the log spectral magnitude for $0 \leq f \leq \frac{F_s}{2}$ of each frame with a **pause(0.3)** between displays to make a movie. You may adjust the amount of pause.
- Plot the spectrogram of the signal.

2. Use time-domain methods described in Chapter 4 of reference 1.

   (a) Use Eqn. 4.6 to find the *Short-time energy* (**STE**) of each frame. Use a Hamming window for each frame.

   (b) Use Eqn. 4.7 to find the *Short-time zero crossing rate* (**STZCR**)of each frame.

   (c) Plot the STE and STZCR versus time in the same figure with the signal as shown in Fig. 4.4.

   (d) Comment on the results.

   (e) Find the *Short-time autocorrelation* (**STACF** of each frame as described in Section 4.2.

   (f) Plot the STACF for each segment. Study your result and comment on your observations. Include only one demonstrative STACF plot in your

   (g) Write a program to determine if a segment is voiced or unvoiced using the STE, STZCR and STACF. The program should consider STE and STZCR thresholds to determine a frame's label. It should also consider the distinguishing properties of the STACF (periodicity) for the label. You may use a vote or a weighted vote of the results of the three measures to make a final decision for a segment's label as *Voiced* or *Unvoiced*.

3. Use cepstral analysis to estimate the fundamental frequency of each frame. If there is a fundamental frequency, record it and the frame number. Claim that frame as voiced. To help you with this algorithm, study MATLAB's program (search in Documentation **Complex Cepstrum — Fundamental Frequency Estimation**.

4. Use eigenvalue decomposition over frames of duration t_d. Save the dominant eigenvalues and their associated eigenvectors as feature vectors. Also explore reconstructing a signal as a linear combination of its eigenvectors, and making approximations to the reconstruction by using the large, i.e. above a threshold, eigenvalues and their associated eigenvectors. All MATLAB variables used below refer to my sample program.

   (a) Following my sample program, divide the signal into 150 ms frames. Feel free to experiment with the frame length to suit your needs. Each frame is a column of matrix **xm**.

   (b) Estimate the autocorrelation matrix of each frame **y=xm(:,k)**by dividing into subframes **ym** and computing **r**.

(c) Find the eigenvalues and eigenvectors **v** of **r**.

(d) Show that the rows and columns) of **v** are orthonormal and **v** is symmetric .

(e) Show that the transpose of **v** is its inverse.

(f) Transform each subframe, i.e. columns of **ym**, by the eigenvectors of the frame. The transforms are in **vym**.

(g) reconstruct columns of **ym** by

- inverse transforming **vym**.
- adding appropriately weighted (weights are the transform values) columns of the eigenvectors.

You should get the same result in both cases and the reconstruction should be perfect. Show this to be true by showing the mean-square-error to be zero.

(h) Reconstruct columns of **ym** using weighted vector sums as above. Start with the eigenvector associated with the largest eigenvalue and move in the direction of decreasing eigenvalues. Each time record and plot the partial sum and compute the mse. Note how many iterations are needed to reconstruct the signal up to an acceptable error. The error must be a measure. Consider "acceptable" reconstruction to be (a) when the mse is below a threshold that shows more that 98 % of the signal energy has been recovered, or (b) when the mse after each iteration shows a considerable decrease.

(i) Plot the largest 3 eigenvalues of each frame versus frame number. Also plot the energy of each frame. Compare the results and observe key frames (plot and determine by inspection) as voiced or unvoiced. Comment on these 4 features as reliable indicators of V/UV.

(j) Optional: Add any creative features using eigenvalues and vectors that may be useful to make the V/U distinction of each frame.

(k) Optional: Find the eigenvectors and eigenvalues of the DFT matrix. Comment on the results.