

Assignment 1 for 11-751

Speech recognition and understanding

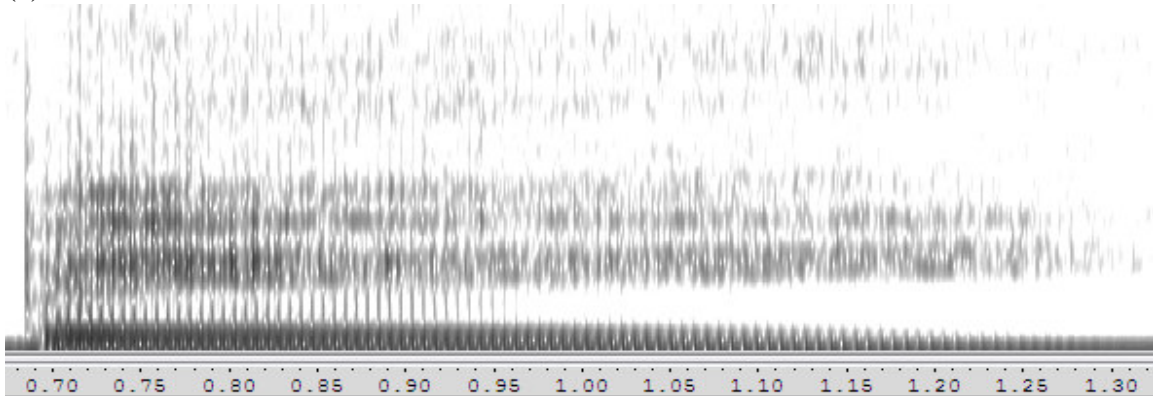
Due on September 17 before class

Teaching assistant: Wilson Tam (yct@cs.cmu.edu)

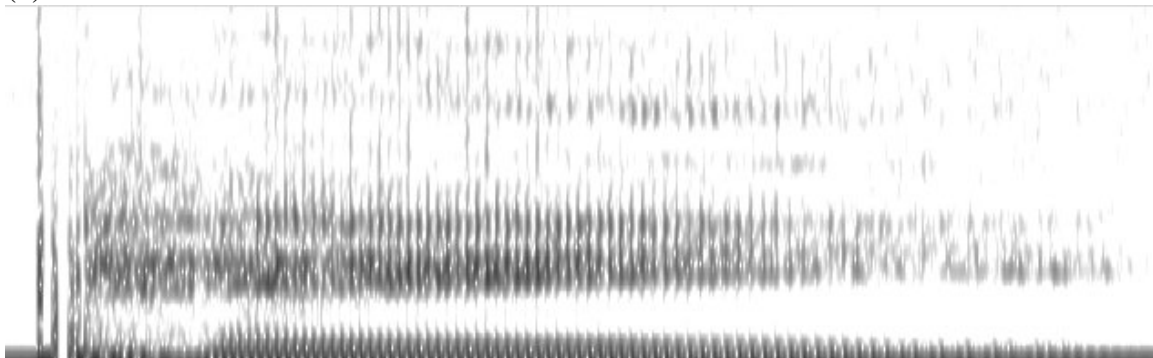
Problem 1: Speech basics

- 1.1 What factors affect the performance of speech recognition?
- 1.2 What is fundamental frequency (F0)? What are formants? What is the relationship between F0 and formants?
- 1.3 What is a spectrogram? How is a spectrogram produced? What are the important information in a spectrogram for speech recognition?
- 1.4 Suggest one feature to distinguish adult male and female voice.
- 1.5 What role does the filterbank play in feature extraction? What is the implication of the non-linear frequency scale in human perception?
- 1.6 What is the bandwidth of a telephone channel? When spelling English letters over the telephone line, which pairs of letters are the most confusable? Explain.
- 1.7 What is a vowel triangle? What are the implications?
- 1.8 Label the two spectrograms below as either “pay” or “bay,” explain your choices.

(a)



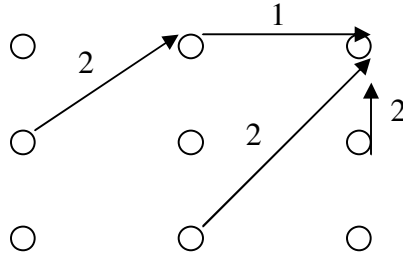
(b)



- 1.9 Suggest strategies for robust endpoint detection.

Problem 2: Dynamic Time Warping (DTW)

- 2.1 What kind of errors can a speech recognizer make? Can word error rate (WER) be higher than 100%? Explain.
- 2.2 Assuming the decoded sentence is on the x axis and the reference sentence is on the y axis, draw the path constraint diagram for WER computation. Indicate which path corresponds to which type of error.
- 2.3 Write down the dynamic programming recursion formula for the following constraint. Use $D(i,j)$ to indicate the accumulated cost and $d(i,j)$ to indicate the local cost.



- 2.4 Write down the pseudo-code to perform DTW based on the above path constraints. Your code should return the cost of the optimal path and the path sequence $\{(i,j)\}$. Assuming the boundary condition applies.
- 2.5 Write down the decision rule for isolated word recognition using DTW. Assume you have N templates.
- 2.6 What is the time complexity of DTW?
- 2.7 Pruning is a common strategy to speed up the DTW procedure. Describe the implementation of beam search in DTW.
- 2.8 Describe how DTW can be extended to continuous speech recognition.

Problem 3: Signal processing

- 3.1 What is aliasing? How to avoid its effect?
- 3.2 What is the difference between waveform, spectrum and cepstrum? Describe the x and y axis of each representation.
- 3.3 Explain one method to extract pitch from a wave signal
- 3.4 Consider the simplest discrete digital high-pass filter: $y(n) = 2x(n) + 4x(n-1) + 3x(n-3)$
What is the impulse response of this filter? What is the transfer function of it? If the Z function of the output is $Y(z)$, what is the Z transform of the input?

Problem 4: Digital signal processing using Matlab

Matlab is a popular tool for providing visual representations of mathematical functions. This problem will hopefully show that there is nothing magic about the front-end DSP in a speech recognition system: you can easily reproduce the effects by using a program like Matlab. Bring up Matlab by typing “matlab” at the prompt. (If it's not installed in your machine, please ask around first, otherwise contact me).

We're going to create a 0.25s-long signal sampled at 1000 Hz, so we set t to steps of 0.001 from 0 to 0.25. (This is just the Matlab FFT demo, by the way.)

```
>> t = 0:.001:.25;
```

Next we can form a signal containing 30 Hz and 100 Hz. Let's do it step by step.

```
>> x30 = sin(2 * pi * 30 * t);
```

```
>> plot(t,x30), title('Pure 30-Hz signal [time domain]')
```

```
>> x100 = sin(2 * pi * 100 * t);
```

```
>> plot(t,x100), title('Pure 100-Hz signal [time domain]')
```

```
>> x = x30 + x100;
```

```
>> plot(t,x), title('Pure 30-and-100-Hz signal [time domain]')
```

Now we add some random noise with a zero mean and standard deviation of 2 to produce a noisy signal y:

```
>> y = x + 2 * randn(size(t));
```

```
>> plot(t,y), title('Noisy 30-and-100-Hz signal [time domain]')
```

4.1 Print the noisy 30-and-100-Hz signal to a ps file by typing "print" at the prompt. You can append to this file using the -append flag. Clearly, it is difficult to identify the frequency components from looking at the original signal; that's why spectral analysis is so popular. Finding the discrete Fourier transform of the noisy signal y is easy; let's take the 256-point fast Fourier transform (FFT):

```
>> Y = fft(y,256);
```

The power spectral density, a measurement of the energy at various frequencies, is found using the complex conjugate function:

```
>> PY = Y .* conj(Y) / 256;
```

To plot the power spectral density, we must first form a frequency axis:

```
>> f = 1000/256*(0:127);
```

which we do for the first 128 points. (The remainder of the 256 points is symmetric.) We can now plot the power spectral density:

```
>> plot(f,PY(1:128)), title('Power spectral density [frequency domain]')
```

```
>> xlabel('Frequency (Hz)')
```

4.2 Print this plot out.

4.3 What do you get from the real and imaginary parts of the spectrum?

4.4 Try plotting the power spectral density of the original signal without the noise. What does noise represent in speech?

4.5 Use the help functions in Matlab to figure out how to produce the log and inverse FFT of the spectrum. Print them out. What is the final result in speech research terminology?