Massachusetts Institute of Technology
Department of Electrical Engineering & Computer Science

6.345 Automatic Speech Recognition
Spring, 2003

Issued: 02/21/03
Due: 03/05/03

Assignment 3
Signal Representation

This assignment is intended to familiarize you with the principles of short-time Fourier analysis (Part I), and cepstral analysis (Part II) as applied to speech. The following tasks (marked by **T**'s) should be completed during your lab session. The answers to the questions (marked by **Q**'s) should be handed in on the due date.

To start the lab enter the following command at the UNIX prompt:

```
% start_lab3.cmd
```

The lab will primarily be controlled with the Lab 3 window. This window contains a set of panels containing layout options (these are called Program options in the Lab 3 window), and a set of utterance buttons. To display an utterance using a particular layout, first select the layout option and then click on the button of the utterance you wish to display. The lab will also make use of the Spectrum Analyzer control window. From this window you can alter many of the features of the spectral analysis such as the window length, window type, etc.

Part I: Short-Time Fourier Analysis

Laboratory Exercise

T1: In this part of the lab you will examine the spectral characteristics of two different windows for performing short-time Fourier analysis. An easy way to do this is to apply the windows to a waveform of constant amplitude, $x[n] = K$, and then alter the characteristics of the window to examine the effects.

To do this, select the Waveform and Spectra layout and the constant utterance, which contains a waveform of constant amplitude, from the Lab 3 window.

For a given window length, compare both the width of the main lobe and the peak amplitude of the side lobes (defined as the difference in amplitude between the main lobe and the largest side lobe), for the **Rectangular** versus **Hamming** window type. You can specify the window by using the window type control in the Spectrum Analyzer. You may want to overlay several plots in the spectrum display window to facilitate comparison. Zooming in on the low-frequency region of the spectra will make it easier to measure the height and width of the main and side lobes.

For a given window type, also examine the effect of different window lengths on the width of the main lobe and the peak amplitude of the side lobes. Use window lengths of 100, 300, and 500 points. You can specify these in the Spectrum Analyzer by using the size (sec) control. Remember that the signal is sampled at 16kHz and that you will need to calculate what the window length (in seconds) is for the above point-lengths.

Do not use the keypad to enter the window size. If you type on the keypad and **Num Lock** is on, the Spectrum Analyzer will die. After entering the the new size in seconds, hit **Enter** or **Return** to see the change in the spectrum window.

T2: In this part of the lab you will investigate the effects of applying different windows to a perfectly periodic waveform. From the results of the previous part, you should be able to predict (or explain) the distortions that these windows may create on the magnitude spectrum.

While in the same layout, select the synthetic-ah utterance, which is a synthetically generated vowel with a perfectly periodic excitation function.

Compare the magnitude spectra for the same cases as before (i.e., **Hamming** and **Rectangular** windows of length **100**, **300**, and **500** points each).

Q1: Determine the fundamental period of the vowel from the waveform. How and under what conditions can this value be determined from the magnitude spectra?

Q2: For the spectra computed with the Hamming windows, what is the general effect of increasing the window size?

Q3: Why are the spectra computed with the rectangular windows ragged-looking?

T3: Repeat **T2** with a naturally-spoken utterance.

Select the Waveform, Spectrogram, and Spectra layout, which is the same as the previous layout except for the addition of a spectrogram plot. The spectrogram shows time on the horizontal axis and frequency on the vertical axis. The degree of blackness of the display represents the energy. The variation in amplitude with frequency for a particular point in time corresponds to the STFT of the speech signal centered at this point. (The STFTs for this spectrogram were calculated with a 6.7ms Hamming window).

Choose either natural-female, an utterance spoken by a female, or natural-male, an utterance by a male.

Place the cursor in various vowel and fricative portions of the utterance and observe the various spectra by holding down the right mouse button while the cursor is within the waveform window and selecting the xspectrum option.

Q4: Why is the level of the high-frequency portions of the spectra computed with the rectangular windows higher than that for the corresponding Hamming spectra?

Homework Exercise

Q5: A speech signal is sampled at a rate of 16,000 samples/sec (16 kHz). A 10 msec window is applied to the speech signal for short-time spectral analysis, and the window is advanced by 40 samples each time. Assume that only radix-2 FFT algorithms are available to compute the discrete Fourier transform (DFT).

(a) How many samples are there in the selected segment of speech?

(b) What is the frame rate of the short-time spectral analysis, i.e., what is the duration (in msec) between each DFT computation?

(c) What is the *minimum* size of the DFT such that no time-aliasing will occur?

(d) What is the spacing in Hz between DFT samples for the size of the DFT as determined from part **(c)**?

The digital signal now needs to be passed through a telephone-based speech recognition system that assumes a sampling rate of 8 kHz.

(e) What pre-processing to the signal is necessary to accommodate this new sampling rate?

(f) Assuming the same 10 msec window size, how many samples are there in the selected segment of speech?

(g) To maintain the same frame rate as in part **(b)**, how many samples should the DFT window be advanced?

(h) How would the answers to part **(c)** and **(d)** change?

Part II: Cepstral Analysis of Speech

Laboratory Exercise

In this part of the lab, you will first examine some of the properties of the complex cepstrum and then show how the cepstrum can be manipulated to obtain either the vocal tract information or the excitation information. You will first study these issues using synthetic utterances whose properties are known exactly. Afterwards, you can examine natural speech from the utterances provided.

The (Complex) Cepstrum

You will first examine some of the properties of the complex cepstrum using two synthetic utterances. The first consists of an impulse at the origin and a scaled impulse 20 samples later. The second is a synthetic /a/, the same utterance used in Part I.

T4: In the Lab 3 window select the layout Waveform Only and then select the utterance impulse-pair.

Compute the (minimum-phase) complex cepstrum by holding down the right mouse button in the waveform window and choosing complex cepstrum from the menu that appears. You will be prompted for the DFT order N (i.e., 2^N point DFT) in a dialog box. Enter a reasonable number between 5 and 12 and hit return. A new window with the complex cepstrum will appear.

Make sure you understand how the complex cepstrum is computed. Measure and verify that the *valid* impulses in the complex cepstrum have the correct amplitude. You may want to rescale the vertical axis to better see the cepstrum. Do this by holding down the right mouse button in the cepstrum window and choosing vertical fixed zoom from the menu. You can restore the window by selecting vertical auto-zoom.

Vary the size of the DFT and observe the effects it has on the extent of aliasing.

Q6: Can you explain why the zeroth value for the complex cepstrum is nonzero? Make some measurements to verify your hypothesis.

T5: Select the layout Waveform Only and the utterance synthetic-ah. Mark out a region on the waveform that encloses several pitch periods.

Compute the cepstrum (zero-phase complex cepstrum) by holding down the right mouse button in the waveform window and choosing cepstrum from the menu. Use a 10th order DFT. You might want to try this on some real speech too.

Q7: Measure the fundamental frequency of voicing (F0) of the synthetic vowel from the cepstrum. Verify your measurement from the time waveform.

Recovering Vocal Tract Information

T6: Using the Waveform and Spectra layout, select the synthetic utterance synthetic-ah.

In the Spectrum Analyzer, set the Analysis type to DFT, the Window type to Hamming, and the size sec to include several pitch periods (i.e., 0.025 seconds). In the spectrum window, save the spectrum as a reference spectrum.

Compute the cepstrum by holding down the right mouse button in the waveform window and choosing cepstrum from the menu. Use a 10th order DFT. Look at the cepstrum and familiarize yourself with the locations of the peaks in it.

In the Spectrum Analyzer, change the Analysis type to CEPST. This option will *recompute* the spectra after performing cepstral liftering operations on the signal. The liftering operations are controlled by the following parameters:

- Liftering: selects high-pass liftering, low-pass liftering, or none.
- Cep. cut (sec): gives the nominal cutoff quefrency.
- Cep. trans: gives the duration of a quefrency transition region between zero and full power.

Experiment with different values of the above parameters. With the appropriate liftering operation on the signal, you should be able to recover just the log magnitude of the frequency response of the vocal tract without any of the source/excitation information.

Q8: In this example, what is the right liftering operation (high-pass, low-pass, or none) and the right cut-off in order to recover just the vocal tract information?

Recovering Excitation Information

T7: As in **T6**, experiment with different values of the liftering parameters, but this time you want to recover just the source/excitation information.

Q9: In this example, what is the right liftering operation and the right cut-off in order to recover just the source/excitation information?

T8: Try the analysis procedure on some natural speech. Observe the difference in the characteristics of the complex cepstrum for voiced and unvoiced speech.

You may want to compare the cepstrally smoothed log magnitude spectrum with the original spectrum to see which one is preferable for extracting formant frequencies and/or fundamental frequency of voicing.

Q10: On the basis of your observations on *natural* speech, what is the appropriate lifter for recovering vocal tract information? What about excitation information?

Cepstral Coefficients

Cepstral coefficients and their time derivatives are used in many modern-day speech recognition systems. In this part of the lab, you will examine some of these features and hopefully gain some insights into why they are useful.

T9: Select the Waveform, Spectrogram, and Cepstral Coefficients layout and choose the utterance natural-female.

In the cepstral coefficients window, cepstral coefficients $c[0]$ and $c[1]$ (top) and their corresponding time derivatives (bottom) are plotted. The time derivatives in this case are computed by taking the difference of the cepstral coefficients four frames (i.e. 20 ms) after and before the current frame. Study the behavior of these parameters for different speech sounds.

Q11: Based on what you see and what you know about the behavior of these parameters, suggest how these parameters may correlate with acoustic properties of speech sounds. For example, what can you say about the value of $c[1]$ for vowels and fricatives?

Mel-Frequency Cepstral Coefficients

Now you will use MATLAB to resynthesize some utterances using only the Mel-frequency cepstral coefficients (MFCCs) derived from those utterances. This should give you some idea as to what information is contained in the MFCCs and whether or not this information is adequate for speech recognition purposes.

To start MATLAB, type the following at the linux prompt:

```
% start_lab3_matlab.cmd
```

Then type the following at the MATLAB prompt:

```
>>init_lab3
```

This will load three utterances, each sampled at 8kHz, into the variables `unusual`, `pathological`, and `cupcakes`.

T10: Listen to the three utterances using the play function. To listen to `unusual` type:

```
>>play(unusual, 8000);
```

To resynthesize these utterances using their MFCCs, you will use the `resyn_from_mfccs` function. This function works by first inverting the MFCC's to obtain a magnitude spectrum at each frame, then using an iterative algorithm to estimate the phase of each frequency component at each frame. Phase estimation is necessary, because phase is ignored when computing MFCCs.

The first iteration of the iterative part of the algorithm starts with initial phase functions of zero for each frame. These phase functions are combined with the magnitude spectra obtained from the MFCCs to produce complete STFTs. Because the frames at which the STFT is computed overlap, not all sets of STFTs correspond to valid signals. To account for this an algorithm is applied to obtain the valid signal whose STFTs match most closely, in the least squares sense, the created STFTs. The resulting signal is considered the result of the first iteration. The second iteration begins with the combining of the phase function of this signal and the magnitude spectra obtained from the MFCCs. At each iteration, the resulting signal looks more like the original signal in that the magnitude spectra are closer in the least squares sense. The algorithm does not, however, guarantee convergence to the original signal. See the following reference for more information:

> D. W. Griffin and J. S. Lim, "Signal Estimation from Modified Short-Time Fourier Transform", IEEE Trans. Acoustics, Speech and Signal Processing, Vol. 32, No. 2, Apr. 1984.

To resynthesize `unusual` with this algorithm using the first 14 MFCCs, type the following at the MATLAB prompt:

```
>>resyn = resyn_from_mfccs(unusual, 14);
```

This will store the resynthesized waveform in `resyn`, which can then be played using the `play` function.

Resynthesize the utterances using various numbers of coefficients and listen to the results. Specifically, resynthesize the waveforms using 2, 14, 25 and 128 MFCCs.

Q12: For each utterance, compare the resynthesized result using different numbers of MFCC's. For how many MFCC's is the overall spectral shape captured well enough to make the utterance intelligible? (You may want to ask a friend who does not know what the utterances are to help you with this.) How many MFCC's are required to capture pitch information?

Q13: Many speech recognizers use the first 14 MFCC's. Using your answer to the previous question, explain why this might be a wise choice.

Q14: Listen to the resynthesized utterances for `unusual` and `cupcakes` using only the first two MFCC's. Which is more intelligible? Why might this be? (Hint: Consider what information is captured in the first two cepstral coefficients and relate this to the spectral characteristics of the phones in the utterance `unusual`. Also consider the number of syllables in each word and how this affects the lexical search space.)

Homework Exercise

Q15: The complex cepstrum, $\hat{x}[n]$, is related to a signal, $x[n]$, by

$$\hat{X}(z) = \log X(z)$$

As we have seen in class, the complex cepstrum of a causal, minimum-phase signal can be generated with the use of DFT's. However, there also exists a recursive relationship for determining $\hat{x}[n]$ directly from $x[n]$.

(a) By differentiating the Z transforms, show that $x[n]$ and $\hat{x}[n]$ are related by

$$n\hat{x}[n] * x[n] = nx[n]$$

(b) Assuming that $x[n]$ is minimum-phase, use this expression to derive the following recursive formula for generating $\hat{x}[n]$,

$$\hat{x}[n] = \begin{cases} 0 & n < 0 \\ \log x[0] & n = 0 \\ \dfrac{x[n]}{x[0]} - \dfrac{1}{nx[0]} \sum_{k=0}^{n-1} k\hat{x}[k]x[n-k] & n > 0 \end{cases}$$

(Use the initial value theorem for $n = 0$.)

Q16: This problem is concerned with some signal processing issues regarding the computation of the Mel-frequency cepstral coefficients (MFCC's) from the Mel-frequency spectral coefficients (MFSC's). We will assume that the length of the speech segment ($s[n]$) is N, the size of the discrete Fourier transform ($S[k]$) is M, and the number of Mel-frequency filters is L. We will also assume that the size of the DFT is sufficiently large such that the effect of aliasing is negligible.

(a) Show that the cepstrum $c[n]$ can be computed as a discrete cosine transform, i.e.,

$$c[n] = \sum_{k=0}^{M-1} \log|S[k]| \cos\frac{2\pi}{M}kn \quad 0 \leq n \leq M-1$$

(A scale factor of $1/M$ is ignored in the above expression.)

(b) The following procedure is typically used to compute the MFCC's:

- The Fourier coefficients $S[k]$ are squared.
- The resultant magnitude-squared spectrum is passed through the Mel-frequency triangular filter banks shown in the lecture handouts.
- The log energy outputs (in decibels) of the filters, $X_k, k = 1, 2, .., L$, collectively form the L-dimensional MFSC vector. Note that there is no MFSC coefficient for $k = 0$. You may find this information important for part (c).

- The MFCC's are then computed by way of a discrete cosine transform.

To carry out the last step, we must treat the MFSC's as the discrete Fourier transform of a real signal. What symmetry properties must be imposed? What is the minimum size of the discrete Fourier transform?

(c) Show that the MFCC's $Y_i, i = 1, 2, .., L$ are given by:

$$Y_i = \sum_{k=1}^{L} X_k \cos[i(k - \frac{1}{2})\frac{\pi}{L}].$$

HINT: The inverse DFT can be computed from any set of equally spaced points on the unit circle.