ELEC-E5520 Speech and language processing methods

Cepstrum assignment, spring 2019

In this exercise, a brief report of cepstral studies is made. In exercises 1-2, the same speech material is used as in the LPC-exercises. In addition, exercise 3 uses new speech material (the material you have already recorded).

Report all requested figures and values clearly. Scale the axes of all figures so that the data relevant to the given questions can be easily observed. Include all MATLAB-codes that you wrote in answering to the exercises.

Exercise 1

1.1 Segment a vowel segment of your choosing (approx. 30 ms) from the speech material using a Hamming window. Similarly to the LPC exercise, it is preferred to choose as stationary segment as possible. Draw magnitude spectrum of the segment and try to estimate first three formant frequencies.

1.2 Change (lifter) the signal in cepstral domain so that the formants become more easily observable in the spectral domain. Draw the new magnitude spectrum of the liftered signal. Estimate formant frequencies again using the new presentation.

1.3 Study how the number of cepstral coefficients affects the spectrum. How many coefficients are required to describe the essential formant structure? What happens when low order coefficients (c_0, c_1,…) are removed?

1.4 Estimate fundamental frequency of the vowel from its cepstral representation (draw a figure).

Exercise 2

2.1 Compute 10th order LPC prediction coefficients for the vowel segment. Use these coefficients to compute complex cepstrum (CC) using the equation provided in lecture slides. Transform complex cepstrum into real cepstrum (see the slides).

2.2 Compute real cepstrum (RC) of the vowel segment directly from time-domain signal using standard cepstral transformation. Compare this representation graphically to the one obtained in 2.1 and report any differences. How the order of the LPC affects the results? Try orders 7, 10, 13 and 25.

Exercise 3

3.1 Choose a 30 ms frame corresponding to vowel /a/ in any selected word, where it has been clearly pronounced. Compute a so-called template for the vowel by computing the cepstrum for the frame and selecting the first m coefficients c(1),…,c(m), where m is a number of your choosing from range 10-30. Do not include the first cepstral coefficient c(0) that describes signal energy. Do the same for vowel /i/ in any selected word, where it has been clearly pronounced.

3.2 Process a test signal (any word including at least five phonemes including the vowels /a/ and /i/) by sliding a 30 ms long window with 15 ms steps across the signal and computing cepstrum from each window position similar to the template computation above. Compute the Euclidean distance between the signal frame and the /a/ and /i/ templates for each window position. Draw a figure with three subplots, where the top panel shows the spectrogram of the speech signal, whereas the two lower panels show the distance curves for /a/ and /i/ vowel templates, temporally aligned with the spectrogram.

3.3 Use distance curve of the /a/ template to estimate which three phone segments of the test signal are closest to the /a/ template (in terms of cepstral distance). Report these phones and the corresponding minimum cepstral distances. Repeat the procedure for the distance curve of /i/ vowel. Is it possible to recognize vowels from the test sample using this approach? How well would the templates work for generic vowel recognition?

3.4 Repeat steps 3.1 – 3.3 using Mel-cepstral coefficients (MFCCs) instead of normal cepstrum. How do the distance curves of MFCCs differ from standard cepstrum? Add white noise to the signals (e.g., by using randn() command of MATLAB) and study how this affects the vowel classfication accuracy with cepstrum and MFCC.

3.5 **S**tudy speaker independent classification by selecting the vowel templates from one speaker and the test signal from another speaker. Suggest better pre-processing methods and distance measures for cepstrum.

Exercise 4

4.1 Divide your speech data into training and testing words. Compute the MFCCs for each frame in each sample by sliding a 30 ms window through the samples. Segment the training samples into phonemes (by hand). Compute *the average of the MFCCs* for all samples of each phoneme and store it as that phoneme's template "A". Select *the MFCCs of the center frame* for all samples of each phoneme and store it as that phoneme's template "B". For phonemes /a/, /m/, /k/ and /s/, plot template A and B in the same graph and report any visual differences of A and B.

4.2 For the MFCCs of each frame in each test sample, measure the distance to each training template and select the closest template as a classification result. Segment the test samples into the correct phonemes (by hand). Report the amount of correctly classified frames for each phoneme using first the template A and then B. For which phonemes A performs better and for which B?

4.3 Discuss the weaknesses of the template-based classifiers (A and B). Suggest several ways for improving the classifier, be creative. Discuss why they could be better.

4.4 *Optional exercise for improving your grade:* Implement and test one or more ways to improve the classifier. Report the results and discuss the pros and cons of your method.