4 Representation of Signals III: Cepstral Analysis, Automatic Speech Recognition I: Fundamentals and Feature Extraction

4.1 Overview

- In this exercise, we first recapitulate the **cepstral analysis technique** (lecture notes p. 86ff.) and give interpretations.

  There are several applications for cepstral analysis, such as pitch detection (see source-model in Fig. 6) or de-reverberation of speech signals. Cepstral analysis is also used for the extraction of *features* for speech recognition engines.

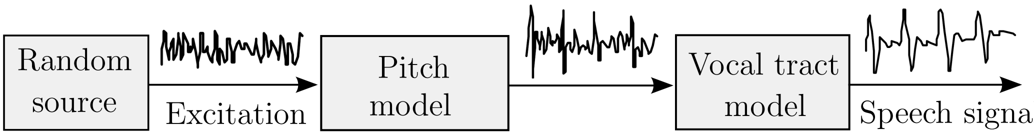


Fig. 6: Source-model for speech production [Vaseghi, 2000]

- This exercise together with exercise 5 also contains a brief introduction to **automatic speech recognition** (ASR) used in commercial products.

4.2 Cepstral Analysis and Liftering

4.2.1 General

The cepstrum has a very interesting property in conjuction with linear time invariant (LTI) systems:

For given real input $x[k]$, impulse response $h[k]$, and output $y[k]$ with the respective DTFTs $X(e^{j\Omega})$, $H(e^{j\Omega})$, $Y(e^{j\Omega})$, we have

$$\begin{aligned} y[k] &= h[k] * x[k] \\ Y(e^{j\Omega}) &= H(e^{j\Omega}) \cdot X(e^{j\Omega}) \\ \ln Y(e^{j\Omega}) &= \ln H(e^{j\Omega}) + \ln X(e^{j\Omega}) \\ c_y[n] &= c_h[n] + c_x[n], \end{aligned} \tag{14}$$

where $c_x[n]$ is the inverse DTFT of $\ln X(e^{j\Omega})$ and is called *complex cepstrum* (note: the first part of the word '*spectrum*' is simply reversed):

$$c_x[n] = \frac{1}{2\pi}\int_0^{2\pi} \ln X(e^{j\Omega}) e^{j\Omega n} d\Omega \tag{15}$$

$$= \frac{1}{2\pi}\int_0^{2\pi} \ln |X(e^{j\Omega})| e^{j\Omega n} d\Omega + j\frac{1}{2\pi}\int_0^{2\pi} \arg\{X(e^{j\Omega})\} e^{j\Omega n} d\Omega. \tag{16}$$

The index n in the cepstral domain is also called *quefrency* (reverse of '*frequency*').

Using this transformation, the *convolution* $y[k] = h[k] * x[k]$, shown above, can be transformed into a *sum* of signals. Note that the complex cepstral transformation is invertible. Moreover, for real time-domain signals, the complex cepstrum $c_x[n]$ is indeed always real as well (even component carries magnitude information, odd component carries phase information, see lecture notes).

These properties make the cepstrum an interesting tool for separating signal components (that are non-overlapping in the cepstral domain), such as *deconvolution.*

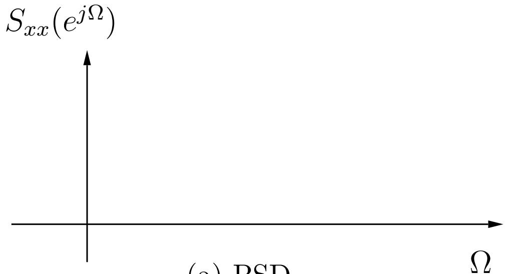
For signal analysis, the *real cepstrum* is often used:

$$c_{x,r}[n] = \frac{1}{2\pi} \int_0^{2\pi} \ln |X(e^{j\Omega})| e^{j\Omega n} d\Omega. \tag{17}$$

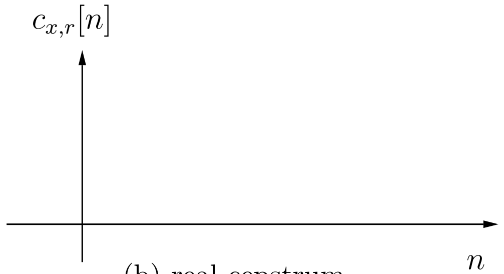
It corresponds to only the first term of the defining equation of $c_x[n]$. We can interpret it as a Fourier series expansion of the logarithmic magnitude spectrum, which describes periodic components of the magnitude spectrum.

4.2.2 Preparation

a, Sketch a typical shape of a short-time power spectral density of a speech signal (voiced). What shape of the corresponding real cepstrum do you expect? $\boxed{\text{T}}$



(a) PSD



(b) real cepstrum

b, Roughly, two areas can be distinguished in the real cepstrum over quefrency. These two areas correspond to two different parts of the well-known speech production model (Section 1.1.1 of the lecture notes). Give a brief interpretation.

▷ ____

c, In which part of the real cepstrum do male and female voices mainly differ?

▷ ____

Which part is therefore potentially interesting for (speaker independent) *speech recognition*? Which one is interesting for (text independent) *speaker recognition*?

▷ ____

4.2.3 Experiments with complex and real cepstrum

M

a, To experiment with speech signals, an m-file `ceps_test_prep.m`, for block processing is provided in the directory `./SHARED_FILES/spsa/Exercise4`. You should use this file as a starting point for the following problems. As the first step in this m-file, the speech input is segmented into short blocks of length 20ms. For simplicity, the blocks are non-overlapping.

What might be the reason for a block length of 20ms?

▷ ____

During the execution of the m-file, you will see a window with four subplots. The uppermost subplot shows the current input signal block. The second subplot shows the corresponding short-time log power spectral density (log PSD). Before the log PSD is calculated, the current input signal block is weighted by a Hamming window.

What is the purpose of this weighting?

▷ ____

You can switch to the next input signal block and its analysis by pressing a button. Do the shapes of the short-time PSDs in subplot 2 correspond to point a, in your preparations (pitch frequency, formant frequencies)?

b, Now, we add a calculation of the real cepstrum in `ceps_test_prep.m` according to Eq. (17). In the m-file you will find a line that is prepared for that calculation. The result should then appear in subplot 3. Can you find the pitch frequency in this subplot (for parts with voiced excitation)?

▷ ____

Optional:

c, Next, we add a calculation of the complex cepstrum in `ceps_test_prep.m` according to Eq. (33) and visualize it in subplot 4.

d, We now implement a backward transformation of the complex cepstrum for each signal block so that the original short-time PSD can be reconstructed in subplot 4 (The complex cepstrum is invertible).

e, Finally, to remove undesired information on the pitch frequency (depending on the speaker) from the desired information on the formant frequencies (depending on the actual content of the speech), we cut the complex cepstrum on a suitable point (see preparations b, and c, and also the result of experiment b,). This filtering in the cepstral domain is called *liftering* (the first part of the word ‘filtering’ is reversed). The result should be shown in subplot 4.

4.3 Automatic Speech Recognition I

In this section and in exercise 5, we give a brief overview of the major techniques used in most of the current state-of-the-art automatic speech recognition (ASR) systems.

4.3.1 Some Fundamentals of Pattern Recognition

A pattern is a function (signal) measured from the physical world using some sensor.

- For **isolated patterns**, recognition means simple *classification*: Every incoming pattern is considered as one entity. It is assigned to one particular *class* out of several possible ones. The assignment of a pattern is independent of all other patterns.
  Examples: Isolated word recognition, recognition of isolated syllables, simple optical character recognition.
- In the case of a **sequence of patterns** where a sequence of classes has to be found, the classification is often generalized to a *decoding process*. Here, it is possible to take into account dependencies between the occurrence of patterns (context).
  Example: Continuous speech recognition.

The patterns that belong to one class should be

- 'similar' to each other
- 'different' from patterns that belong to other classes

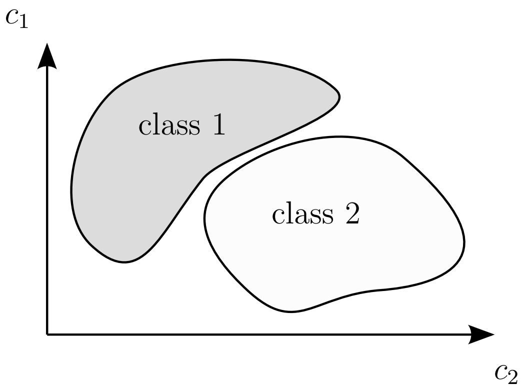


Fig. 7: Simple example: feature space spanned by two features c_1 and c_2, i.e., vectors $\mathbf{c} = [c_1, c_2]$.

The similarity between two given patterns is determined using so-called *features*.

- Features are certain quantities, derived from the patterns. The features should be chosen such that they contain in a compressed way all the information that is necessary for reliable classification of the patterns at hand.

- This way, each pattern of interest can be represented by one vector **c** in a so-called *feature space* (see *Fig.* 7).

From *Fig.* 7 it is clear that a certain 'clustering' in the feature space is a necessary condition for a successful classification.

From the above considerations, we can immediately derive the *basic structure of a system for pattern classification* as shown in *Fig.* 8.

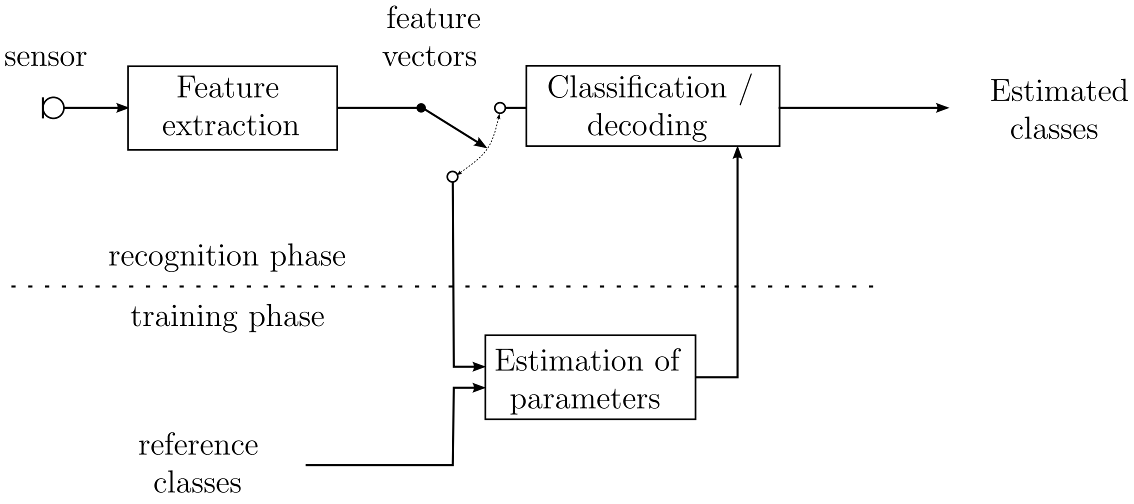


Fig. 8: Basic structure of a system for classification.

In the case of speech recognition the process can be summarized as follows: An unknown speech signal $s(k)$ is converted by a front-end signal processor into a sequence of feature vectors $\boldsymbol{C} = \langle \boldsymbol{c}(1), \boldsymbol{c}(2), \ldots, \boldsymbol{c}(T) \rangle$. Each of these vectors is a compact representation of the short-time speech spectrum covering a period of typically 10ms. Thus, a typical 10-word utterance might have a duration of around 3s and would be represented by a sequence of $T = 300$ acoustic vectors. The utterance consists of a sequence of words $\boldsymbol{w} = \langle w_1, w_2, ..., w_n \rangle$ and the task of the ASR system is to determine the word sequence $\boldsymbol{w}$ with the highest probability, given the observed acoustic sequence $\boldsymbol{C}$.

In the next section, we consider the feature extraction more closely. The decoding process will be considered in exercise 5.

Question: Why are the incoming patterns *not* directly compared with some reference signals? (One could argue that there is only a finite set of possible patterns due to the digital representation.)

▷ ____

▷ ____

1.	*Spectrum of windowed block* $\tilde{s}(m)$ *of speech signal* $S(\mu, n) = \sum_{m=0}^{M-1} \tilde{s}(n, m) e^{-j\frac{2\pi}{M}m\mu}$ ($\mu = 0, \ldots, M-1$, M is length of DFT, n is current block number)
2.	*Mel-frequency spectrum* $\tilde{c}(\nu, n) = \sum_{\mu=0}^{M-1} \lvert D(\nu, \mu) S(\mu, n)\rvert^2$ ($D(\nu, \mu)$ are triangular frequency bins according to mel scale, ν is index of bin)
3.	*Mel-frequency cepstrum coefficients (MFCCs)* $c(q, n) = \sum_{\nu=1}^{N} \log(\tilde{c}(\nu, n)) \cos \frac{\pi q(2\nu+1)}{2N}$ ($q = 1, \ldots, N$ is quefrency, N is number of bins)

Table 1: Calculation of the MFCCs

4.3.2 Extraction of Features

The first step of the front-end parameterization stage for any possible set of features is to divide the input speech into blocks and from each block derive a smoothed spectral estimate. The spacing between blocks is typically 10ms, and blocks are normally overlapped to give a longer analysis window (about 25ms). Also the speech signal is often pre-emphasized by applying high-frequency amplification to compensate for the attenuation caused by the radiation from the lips. After an application of some window function (e.g., a Hamming window) the spectral estimates may be computed via linear prediction (see exercises 3 and 4) or the Fast Fourier Transform (FFT) and there are several additional transformations that can be applied in order to generate the final acoustic vectors which should just contain the essential information needed in the following components of the ASR.

Mel-Frequency Cepstral Coefficients A typical set of features are the *mel-frequency cepstral coefficients* (MFCCs). This method is currently the state of the art and a short overview is given in *Table* 1. Note that there are several slightly different definitions of the MFCCs in the literature.

4.3.3 Experiments on MFCCs

M

Experiment 1.
In the directory `./SHARED_FILES/spsa/Exercise4`, you will find the file `mfcc_test.m` which illustrates the steps involved in the MFCC calculation.

First, the utterance ‘A huge tapestry hung in her hallway’ with a sampling rate of 16kHz is loaded, and the signal is plotted in a figure. By pressing a button, you will move on to the next steps (Spectrogram, MFC, and MFCC domains). Also note the explanations appearing on the screen. Compare the results with those of Section 2.

As in most practical systems, the MFCC feature vectors $\boldsymbol{c}(n)$ in the program are built by 12 MFCC coefficients, and extended by the short-time signal energy.

In state-of-the-art systems, these 13 features are often further extended by the differentials (approximated by linear regression) of all the coefficients, finally giving a feature space with 39 dimensions in which the feature vector changes its position every 10ms.

Experiment 2.
Using the same file `mfcc_test.m` we can also see the 'liftering effect' similarly as in Section 2. Both, in the MFCC, and in the frequency domain, the contours become smoother.

Experiment 3.
Using another file `mfcc_vocals_test.m` you will see *clusters* of features derived from different vocals according to *Fig.* 7. The signal is actual recorded speech. The figure shows a 2-dimensional subspace of the actual 13-dimensional feature space.

Experiment 4.
Compare the clusters in the feature space for two different speakers by changing the name of the data file in `mfcc_vocals_test.m` from `vocals_herbert.mat` to `vocals_robert.mat`.