Tutorial - Exercise sheet 2

Pattern and Speech Recognition

Introduction

The goal of the current assignment is to make you understand the main informations about feature extraction. We are focusing the study on speech processing. At the end of this exercises sheet, you should

- understand how to analyze a spectrogram associated with speech
- understand the goal of using a window and to choose a proper one
- understand the different stages to get a spectrogram
- understand the different stages to get cepstrum coefficients (bonus exercise)

1 Preparation [1pts]

1.1 Loading and plotting in the waveform

In order to conduct your experiments, you have to prepare your environment.

Ex. 1 — Retrieve the sample from this URL:
`http://www.coli.uni-saarland.de/~slemaguer/teaching/cmu_us_arctic_slt_a0001.wav`

Ex. 2 — Load the sample and plot the waveform.

Ex. 3 — Plot the waveform between 0.55ms and 0.6ms. Can you find a pattern? If it is the case, indicate the pattern on the waveform.

Ex. 4 — Plot the waveform between 2.0ms and 2.1ms. Can you find a pattern? If it is the case, indicate the pattern on the waveform.

2 Spectrum analysis [9pts]

From the previous part, we have seen that time-domain representation are exhibiting some properties. However, the properties cannot be processed easily by a computer. Therefore the objective of this section is to obtain another representation of the signal: the spectrogram.

2.1 A reference plot

Ex. 5 — Plot the spectrogram of the sample (Use the function *specgram* for python or *spectrogram* for matlab). Use the default parameters.

Ex. 6 — Plot the spectrogram from the segment which is starting at 0.55ms and ending at 0.6ms. Is it a voiced or an unvoiced segment? Justify your answer.

Ex. 7 — Plot the spectrogram from the segment which is starting at 2.0ms and ending at 2.1ms. Is it a voiced or an unvoiced segment? Justify your answer.

2.2 Compute spectrum

During this part, you are going to develop your the first part of the feature extraction process. At the end, you will have the representation of the power spectrum of a specific section.

2.2.1 Windowing

Ex. 8 — Implement a function for each following windows and plot all of them. T is a **parameter** of the function. For the plot, we consider $T = 100$.

- The rectangular window is defined

$$w(t) = 1, 0 \leq t < T \tag{1}$$

- the Hamming window is defined by

$$w(t) = \alpha - (1-\alpha)cos(\frac{2 * \pi * t}{T-1}), 0 \leq t < T \tag{2}$$

α is also a **parameter of the function**. For this assignment we consider $\alpha = 0.54$.

The window purpose is to be shifted over the signal. Therefore, to apply this process, we need to define a shift step (named *frameshift*) and a window size (identified by *winsize*). From now, we are considering the following values:

- *frameshift* = 5ms
- *winsize* = 25ms

Ex. 9 — The frame shift is smaller than the window size. What does it implies? Based on this information, which window function should you choose? Justify your choice.

Ex. 10 — Implement the function which is applying the window to the original signal. The frame shift and the window size should not be hardcoded in the function.

2.2.2 Generate a spectrum section

Ex. 11 — We are considering the frame started at 0.55ms, which indexes are you going to use to extract the data you need? (i.e. considering the notation x[ss:ee] what are the values of ss and ee). Justify.

Ex. 12 — Apply the fourier transform (using the function *numpy.fft.fft* for python and *fft* for matlab). We consider $N = 1024$. Plot the results. Are we expecting this results? Justify.

Ex. 13 — Apply the function function *numpy.fft.fftshift* for python or *fftshift* for matlab). Plot the results and compare to the previous one.

Ex. 14 — Which property is exhibited by the previous plot? Remove the useless part and justify your choice.

Ex. 15 — Currently, the x-axis are indexes. What should the x-axis represent? Justify your answer and adapt your code to correct your plot. (Small hint: you have to use the sample rate of the signal which is 48000)

2.2.3 Generate the power spectrum

Generally, we are using the power spectrum which is providing real values instead of complex values provided by the Fourier transform. Furthermore, we generally represent this spectrum in the logarithm domain. The power spectrum $S(f)$ is computed from the spectrum $s(f)$ using the following equation:

$$S(f) = 10 * log10((s(f) * conj(s(f)))^2) \tag{3}$$

Ex. 16 — Implement a function which is computing power spectrum coefficients from the spectrum coefficients

Ex. 17 — Plot the power spectrum.

Ex. 18 — Based on the power spectrum, how will you generate the spectrogram? (Indicate briefly the procedure, you do not have to implement this part)

3 From spectrum to mel-cepstrum [bonus: 2 pts]

We are now considering the power spectrum associated to the frame associated at 0.55ms.

Ex. 19 — Apply the cosine transform (*scipy.fftpack.dct* for python, *dct* for matlab) and plots the results

Ex. 20 — Develop the function which is providing the mel-scaled filterbank. Apply it in its proper place. Plots the results and compare with previous plot.

Submission instructions

The following instructions are mandatory. If you are not following them, tutors can decide to not correct your exercise.

Submission architecture

You have to generate a **single ZIP file** respecting the following architecture:

```
tutorial1_<matriculation_nb1>_<matriculation_nb2>_<matriculation_nb3>
|
+--- source
|      |
|      +------- file 1
|      +------- file 2
|      +------- ...
+--- rapport.pdf
+--- README.txt
```

where

- **source** contains the source code of your project,
- **rapport.pdf** is the report where you present your solution with **the explanations (!)** and the plots,
- **README** which contains group member informations (name, matriculation numbers and emails) and a **clear** explanation about how to compile and run your source code

The ZIP filename has to be :

`tutorial1_<matriculation_nb1>_<matriculation_nb2>_<matriculation_nb3>.zip`

You have two choose between the following languages **python** or **matlab**. Other languages won't be accepted.

Some hints

We advice you to follow the following guidelines in order to avoid problems :

- Avoid building complex systems. The exercises are simple enough.
- Do not include any executables in your submission, as this will cause the e-mail server to reject it.

Grading

Send your assignment to the tutor who is responsible of your group:

- Gerrit Großmann `gerritgr@gmail.com`
- Sébastien Le Maguer `slemaguer@coli.uni-saarland.de`
- Kata Naszádi `b.naszadi@gmail.com`

The email subject should start with `[PSR TUTORIAL 2]`