CMRM Homework Assignment No. 2 (HWA2)

December 12, 2022

1 Description

The main topic of this homework is key detection based on chromagrams. The goal is to implement an algorithm that is able to detect the key of music tracks, and compare the performance of different types of chromagrams. The proposed algorithm is used to estimate the key of 4 different tracks belonging to different musical genres. Use the Jupyter Notebook named `Homework.ipynb` to implement the code, and explain in a report, step by step, what you have implemented.

1.1 Question 1

Define a function that computes, plots, and returns the chromagram considering as input the audio signal. The function will be named `compute_chromagram`. You find a template in the notebook. In particular:

- use the function `librosa.feature.chroma_stft` or `librosa.feature.chroma_cqt` for computing the chromagram according to the input argument `chroma_type`;
- use the function `librosa.display.specshow` for plotting the chromagram if the input argument `print_chromagram` is `True`.

Once the function is defined, test it on the wav file:

`LynyrdSkynyrd_SweetHomeAlabama.wav`,

available in the `/data/audio/` folder, and print both `stft` and `cqt` outputs. Remember to provide comments in the notebook for each step.

Explain in the report the idea behind both `stft` and `cqt` chromagrams. In particular, address the following points:

- How is the `stft` chromagram defined? (provide just an intuitive explanation, no formulas)
- How is the `cqt` chromagram defined? (provide just an intuitive explanation, no formulas)
- What are the main differences between the two approaches? Can you mention pros and cons?

1.2 Question 2

Load the `ground_truth.xlsx` file using the `pandas` library. You find such a file in the `/data/` folder. It contains the correct keys for all the tracks that we are going to analyze in this homework. You will need to add `pandas` and `openpyxl` to your conda environment in order to solve this point. Once you have loaded the file, print the key associated to the song `LynyrdSkynyrd_SweetHomeAlabama.wav`. Then, perform the following steps:

1. Define a list `chroma_labels` containing all the chromas (e.g., `C`, `C#`, etc.). Starting from this list, define a new list `keys` containing all the names for the considered scales, adding to the chromas the words `major` or `minor`. In this application, we will thus consider only the 24 basic keys. The list will have thus 24 entries, and will be of the type:

   `keys = ['C major','C# major',...,'B major','C minor','C# minor',...,'B minor'].`

2. Compute the prominance $\mathcal{P}[c]$ of each chroma c in the track. Given the chromagram $C[c, n]$ (consider, for this question, only `librosa.feature.chroma_stft`), with $n \in \mathbb{R}$ being the time index and $c \in [0 : 11]$ being the chroma index, this can be achieved performing the following pooling operation:

   $$\mathcal{P}[c] = \sum_n C[c, n]\,. \tag{1}$$

   Then, create a list named `chroma_vals` for collecting all the $\mathcal{P}[c]$ as:

   $$\texttt{chroma\_vals} = [\mathcal{P}[0], \ldots, \mathcal{P}[11]].$$

3. Define a dictionary `key_vals` that has as keys `chroma_labels` and as values `chroma_vals`.

Print or plot the intermediate results. Provide comments in the report on the performed steps. What is the purpose of the pooling operation?

1.3 Question 3

In order to guess the key of the audio track that we are processing, we may think to consider the correlation between `chroma_vals` and some profiles for major and minor scales. These profiles can be the simple binary templates, similar to the ones we have considered in class, i.e.,

```
maj_profile = [1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1],
min_profile = [1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0],
```

or more refined templates, such as perceptually-based templates. For example, we may consider the the following perceptual templates:

```
maj_profile = [6.35, 2.23, 3.48, 2.33, 4.38, 4.09, 2.52, 5.19, 2.39, 3.66, 2.29, 2.88],
min_profile = [6.33, 2.68, 3.52, 5.38, 2.60, 3.53, 2.54, 4.75, 3.98, 2.69, 3.34, 3.17],
```

which have been obtained starting from a "probe-tone" experiment conducted with expert listeners. In order to compute the key associated to the song, perform the following steps:

1. Compute the Pearson correlation coefficients between the `chroma_vals` in `key_vals` and the binary major and minor profiles. You can use, for example, the `np.corrcoef` function.

2. Define a dictionary `key_dict` considering as keys `keys` and as values the correlation coefficients. We define the most likely key as the key associated to the maximum correlation coefficient. Assign this key to `best_key` and the associated correlation coefficient to `best_corr`. Print both the results. Compare it to the ground-truth that you have printed earlier. Do they match? Is the algorithm able to detect the correct key?

3. Go over again step 1 and step 2 but, this time, consider the perceptual major and minor profiles. What about the key now? Does it match the ground-truth?

Print or plot the intermediate results. Provide comments in the report on the performed steps. In particular, address the questions mentioned in step 1 and step 2, and explain the reasons why the algorithm does or does not work fine in the two cases.

1.4 Question 4

Define a function that provides an estimate of the musical key considering as inputs the chromagram of the audio file and the profiles for major and minor scales. The function will be named `detect_key` and will contain the operations that you have carried out earlier in Question 1.2 and Question 1.3. You find a template in the notebook. In particular:

- compute the prominance of each chroma in the song;
- define the 12 chroma labels (in a list) and the 24 key names (in a dictionary);
- compute the Pearson correlation coefficient between the key profiles and the prominance of each chroma;
- define a new dictionary associating all the correlation coefficients to the keys;
- compute the most likely key (i.e., `best_key` with its correlation coefficient being `best_corr`);
- compute the second best key, and assigned it to `alt_key`. This must be determined checking whether exists or not a correlation coefficient that is $> 75\%$ of the best correlation coefficient. Assigned this correlation coefficient to `alt_best_corr`.
- return `best_key`, `best_corr`, `alt_key`, and `alt_best_corr`.

Test the function, and check that the results are equivalent to the ones obtained in Question 1.3 (when considering the `stft` chromagram as input). Provide comments in the report.

1.5 Question 5

Apply the functions `compute_chromagram` and `detect_key` for computing the key of all the tracks, namely

- `LynyrdSkynyrd_SweetHomeAlabama.wav`,
- `Beatles_HereComesTheSun.wav`,
- `Vivaldi_AllegroNonMoltoRV297.wav`,
- `ElvisPresley_BlueChristmas.wav`,

which are all contained in the folder `/data/audio/`. Analyze how the chromagram type and the key profiles affect the performance of the algorithm. In particular, apply both binary and perceptual profiles to both the two types of chromagram (i.e., `stft` and `cqt`). How does the `cqt` chromagram behave with the binary profile? And with the perceptual profile? And then, how does the `stft` chromagram behave with the binary profile? And with the perceptual profile? Are the different versions of the algorithm able to detect the correct key? Print and plot everything, and provide comments in the notebook. Moreover, considering just the perceptual profiles and helping yourselves with the plots of the chromagrams, can you explain why sometimes `cqt` (or `stft`) performs better than `stft` (or `cqt`)? Can you think of a strategy to improve the performance of the algorithm?

2 Files to be Delivered

You can address this assignment on your own or forming a group of two people. You are required to deliver the following files:

1. a report, containing all answers to the questions and comments to the code. Include you surname/surnames in the title of the report (e.g., `Rossi.docx` or `Rossi_Bianchi.docx`). You can use whatever editor, even LaTeX, and, in general, you can provide a pdf file (rather than a doc file);
2. the filled `Homework.ipynb` file. This is already divided into different sections and cells according to the questions that you are required to solve. In order to ease the solution, the notebook is provided with some guidelines in the form of comments. Rename the notebook with your surname/surnames (e.g., `Rossi.ipynb` or `Rossi_Bianchi.ipynb`). Please, add comments to the code, and plot or print all intermediate results. It is suggested to add titles, axis labels, and/or legends to the plots.

Zip the report, the notebook, and the `data` folder. Name the zip file using your surname/surnames (e.g., `Rossi.zip` or `Rossi_Bianchi.zip`). The zip must be turned in by **Wednesday Dec 21** (before midnight). As far as grades and general rules are concerned, please, refer to those described in Homework Assignment No. 1. **Only one student for group must load the zip file on WeBeep**.