CMRM Homework Assignment No. 2

December 10, 2021

1 Description

The main topic of this homework is template based chord recognition. The goal is to propose and implement a metric able to evaluate the performance of the template based chord recognition algorithm. The proposed metric is then used to evaluate the performance on 4 different songs. Finally, the evaluation is repeated for different values of the algorithm parameters. Use a Jupyter Notebook to implement the code and explain in the report what you have implemented step by step.

1.1 Question 1

Implement the template based chord recognition algorithm. Define a function that takes as input the path to a wav file and returns the estimated chords sequence labels.

The output must be a list

$$\boldsymbol{\lambda}^{\text{pred}} = [\lambda_0^{\text{pred}}, \lambda_1^{\text{pred}}, ..., \lambda_{N-1}^{\text{pred}}] \tag{1}$$

where each element λ_n^{pred} is the predicted chord label for the time frame n. The length of the list depends on the feature rate, i.e., both on the window length and hop size used for the chromagram computation and on the downsampling factor, if feature downsampling is applied.

The chord templates to be considered are the major triads and the minor triads, leading to a total of 24 templates.

Once the function is defined in the notebook, test the function on the wav file `Beatles_LetItBe.wav`, available in `/data/wav/` folder, and print or plot the output.

Explain in the report the idea behind the template-based chord recognition algorithm and detail each step implemented in the code, including pre processing and post processing phases.

1.2 Question 2

Write a function to load and preprocess a reference annotation (or ground truth) file, saved in CSV format. The function should take as input the path to a CSV file and produce as output a list of ground truth chord labels, after suitable pre processing. The output must be a list

$$\boldsymbol{\lambda}^{\text{gt}} = [\lambda_0^{\text{gt}}, \lambda_1^{\text{gt}}, ..., \lambda_{N-1}^{\text{gt}}] \tag{2}$$

where each element λ_n^{gt} is the ground truth chord label for the time window n. The length of the list must be adapted to match the the feature rate.

The reference annotations stored in the CSV file are given in the form of labelled segments, each specified as a triplet $(\text{start}, \text{end}, \lambda)$ where start and end are expressed in seconds. To load the CSV file check the `csv` library (distributed with Python) or `Pandas` library (needs to be installed).

In the preprocessing step you should

- convert the segment-based annotation into a frame-based label sequence adapted to the feature rate used for the chroma sequence;
- convert the labels used in the annotation file to match the chord labels used for the chord recognition algorithm in terms of enharmonic equivalence (i.e., **Db** = **C#**);

- reduce the chord label set used in the annotation to match the chord labels set used for the chord recognition algorithm (i.e., **DMaj6** → **D**); for this step you can choose any reduction strategy.

Once the function is defined in the notebook, test the function on the CSV file `Beatles_LetItBe.csv`, available in `/data/csv/` folder, and print or plot the output.

Explain in the report each step of the preprocessing phase, focusing in particular on the reduction strategy of the chord label set.

1.3 Question 3

Propose a metric for evaluating the template based chord recognition algorithm. A metric

$$m = f(\boldsymbol{\lambda}^{\text{pred}}, \boldsymbol{\lambda}^{\text{gt}}) \tag{3}$$

is a scalar number that expresses how good is the algorithm in performing the task of chord recognition.

The proposed metric should have higher values when the chord recognition algorithm is able to predict correctly the ground truth chords most of the times, lower values if the chord recognition algorithm often fails at recognising the chords.

Write a function that takes as input the list of predicted chord labels, the list of ground truth chord labels and computes the proposed metric value. The two input lists must have same length and the output must be a scalar value.

Once the function is defined in the notebook, test the function on the two lists of predicted and ground truth chord labels computed in Question 1 and Question 2.

In the report write a formal definition of the proposed metric and thoroughly explain the idea behind the proposal. Can you imagine a musically informed strategy that weights differently mismatch errors of the chord recognition algorithm?

1.4 Question 4

Compute the proposed metric for the remaining 3 songs:

- audio `Beatles_HereComesTheSun.wav`, CSV `Beatles_HereComesTheSun.csv`
- audio `Beatles_PennyLane.wav`, CSV `Beatles_PennyLane.csv`
- audio `Beatles_ObLaDiObLaDa.wav`, CSV `Beatles_ObLaDiObLaDa.csv`

all contained in folders `/data/wav/` and `/data/csv/`. Print or plot the metric values.

1.5 Question 5

Analyse how algorithm parameters affect the performance of the templated based chord recognition algorithm.

Given one algorithm parameter (i.e., smoothing filter length L), choose a range of 3 possible values for it (i.e.,$L = [0, 10, 20]$). For each value of the parameter, compute the predicted labels and the correspondent metric value for each song. This produces a list of 3 metric values for each song.

Plot the results for all songs in a figure where the x-axis corresponds to the parameter values, the y-axis corresponds to metric values. An example of the plot is in Figure 1.

Repeat this for at least 3 different algorithm parameters. Note that in every experiment you need to change only one parameter value at a time, while the others must be fixed.

Show the 3 plots both in the Jupyter Notebook and in the report. What considerations can you do from the 3 plots that you have? Are some algorithm parameters affecting the results more than others? Comment the results in the report.

2 Files to be delivered

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`).

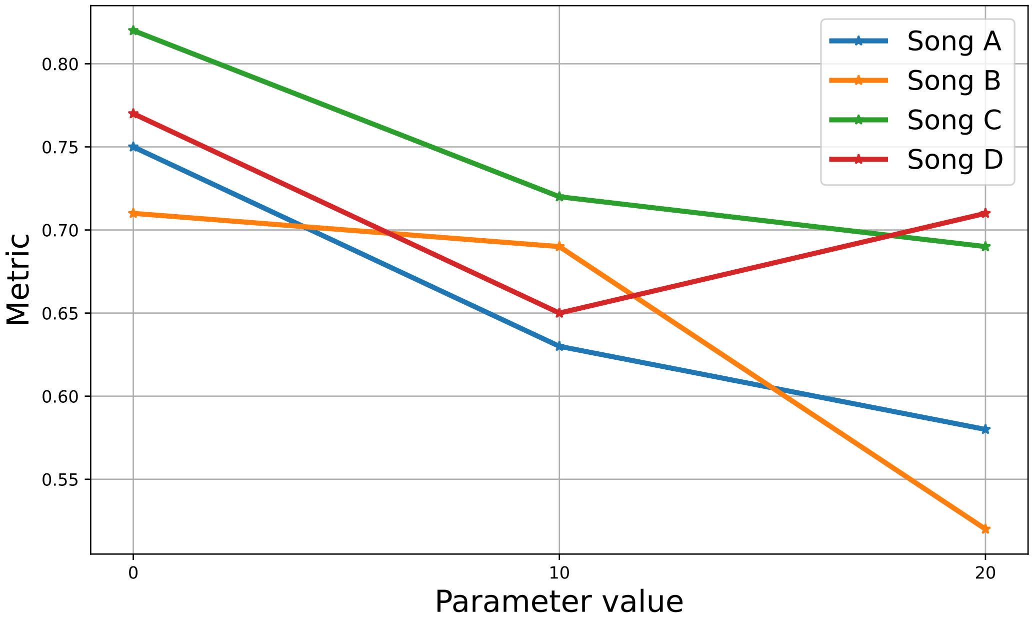

Figure 1: Example of plot for Question 5

2. a Jupyter Notebook file, divided in 5 sections, one for each question. Rename the notebook with your surname/surnames (e.g., `Rossi.ipynb` or `Rossi_Bianchi.ipynb`). Pleas, add comments to the code and plot or print all intermediate results. It is suggested to add titles, axis lables or legends in the plots.

Put both the report, the notebook and the data folder provided in a new folder, named using your surname/surnames and compress the folder in a zip file. **Only one student for group must load the zip file on WeBeep**.