Music Informatics Coursework 1: Beat Tracking for Ballroom Dance Music

Pedro Pereira Sarmento

Queen Mary University of London

March 9, 2020

1 Overview of the task

Our task is to implement and test a beat tracking system for ballroom dance music. Specifically, we are asked to develop a procedure that returns the *tactus* (or primary metrical level) and also the downbeat (the beat at the beginning of each measure), of a given audio excerpt. To this extent, we implemented and built upon the solution proposed by Ellis which utilizes dynamic programming [2] for the problem of beat tracking. Furthermore, this implementation was supported by the guidelines presented in the module's textbook [3], as cited in the files attached to this submission (*.ipynb* and *.py* with the same implementation).

2 Background

The notion of what constitutes rhythm in music is indispensable for a better understanding of the beat tracking problem. Referring to the medium-scale temporal characteristics of music (excluding the ones that concern the evolution of individual notes), rhythm can be perceived as a composition of: **pulse** (equally spaced sequence of perceived accents in time), **metre** (the structure that best fits the pulse) **tempo** (the rate of the primary pulse) and **timing** (the occurrence of an event with respect to its position in the structure). Of these, the concept of beat is derived from the pulse, often from what is known as the *tactus*, or primary pulse, and it is linked with *the rate at which one claps or taps along with the music* [3]. Furthermore, a downbeat is considered as the beat at the beginning of each measure.

In typical rhythm-focused music information retrieval applications, rhythm is usually based on onset times. Thus, the taks of onset detection, the estimation of the beginnings of the attack transients, is considered as an important preliminary step towards the inference of higher-level musical features, such as periodicities and accentuations in the signal, and, ultimately, rhythmic properties. A standard approach in onset detection consists in first determining an onset detection function (also known as novelty function), subsequently applying some post-processing steps (e.g. peak-picking). To this extent, several different signal processing approaches might be undertaken, namely: energy-based onset detection, spectral-based onset detection, phase-based onset detection and complex domain onset detection [3].

Regarding beat tracking algorithms, the last thirty-five years have witnessed the establishment of increasingly accurate solutions. In his master thesis, Ritcher presents an historical overview of the most influential work [4]. As with onset detection, beat tracking state-of-the-art approaches use deep learning techniques, (e.g. temporal convolutional neural networks [1]).

3 Implementation

In this assignment, an implementation of the algorithm presented in [2], leveraging dynamic programming for the task of beat and downbeat tracking, was carried out. This system is driven by an objective function that seeks to maximize both the *onset strength* at every hypothesized beat time (where the onset strength function is derived from the music audio by some suitable mechanism), and the consistency of the inter-onset-interval (assuming some pre-estimated constant tempo). This procedure is based on the assumptions that beat positions go along with the strongest note onsets and that the tempo is roughly constant. The following subsections consist of the steps taken to determined both the beat sequence and the downbeat sequence. The implementation was done in Python 3.7 using the packages *numpy* and *librosa*, and package *mir_eval* for the evaluation.

3.1 Onset Detection Function

As discussed in section 2, the task of onset function detection is central to this approach. Following the procedure in [3], we used spectral flux by first calculating the STFT of the input signal with a window size of 1024 samples, a hop size of 512 samples, for a sampling rate of 22050 Hz. Next, a logarithmic compression step was applied, considering a factor γ of 100, followed by the calculation of the first derivative of the spectrum and a process of half-wave rectification, excluding drops in energy, ending with a sum across all frequencies. Then, local averaging with a window of size $M = 10$ was performed. Finally, a normalization was done, dividing the onset detection function by its maximum value.

3.2 Tempo Estimation

In this implementation we assumed a pre-estimated constant tempo and its determination was done automatically by exploring the function *librosa.beat.tempo*, passing our onset detection function as an argument. This yields a tempo estimation in BPM (beats per minute) which is later conversed to frames per beat, using the sampling rate from the previously calculated onset detection function.

3.3 Beat Estimation with Dynamic Programming

Assuming the onset detection function referred in section 3.1 and the tempo estimation from section 3.2, we follow the approach proposed by Ellis whereas the generated beat sequence corresponds to rhythmic, regular patterns present in the perceived onsets [2]. This is expressed by the following objective function:

$$C\left(\{t_i\}\right) = \sum_{i=1}^{N} O\left(t_i\right) + \alpha \sum_{i=2}^{N} F\left(t_i - t_{i-1}, \tau_p\right) \tag{1}$$

Here, from the subset of the onset detection function $O\left(t_i\right)$, composed of onset values from the indexes that correspond to the beat sequence t_i, we subtract a consistency function $F(\Delta t, \tau)$ weighted by a consistency factor α (Eq. 2).

$$F(\Delta t, \tau) = -\left(\log \frac{\Delta t}{\tau}\right)^2 \tag{2}$$

Intuitively, the consistency function measures how similar two consequent beats from the beat sequence are with respect to τ, the number of frames per beat retrieved from the tempo estimation step in section 3.2. In order to estimate the best beat sequence, the best score for time t is the local onset strength, summed with the best score to the preceding beat time τ that maximizes the sum of that best score and the consistency from that time, presented in Eq. 3. Furthermore, throughout the process we keep track of the actual preceding beat that yielded the best score (Eq. 4).

$$C^*(t) = O(t) + \max_{\tau=0\ldots t} \left\{\alpha F\left(t-\tau, \tau_p\right) + C^*(\tau)\right\} \tag{3}$$

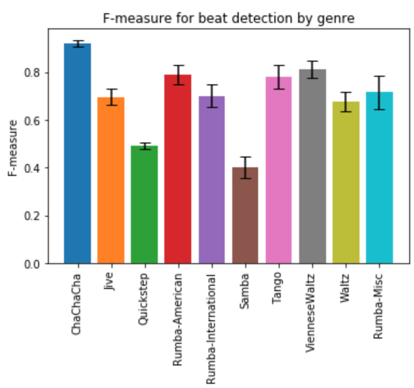
$$P^*(t) = \arg\max_{\tau=0\ldots t} \left\{\alpha F\left(t-\tau, \tau_p\right) + C^*(\tau)\right\} \tag{4}$$

Searching for the beat times that optimize the objective function, we first calculate C^* and P^* for every t (*forward step*). We then search for the largest value of C^*; this result yields the final beat instant t_N (N being the total number of beat times). The optimal beat sequence $\{t_i\}$ is determined following the *back tracking* step, looking backwards via P^*, whereas the previous beat time t_{N-1} is $P^*(t_N)$. Here we leverage dynamic programming, breaking this exponential in N problem into sub-problems, yielding a linear-time operation. This loop is performed until we reach the beginning of the signal, where $P^*(t) = 0$. Finally, we output the beat sequence with the best score, comprised of onset detection function indexes that are further conversed to time stamps.

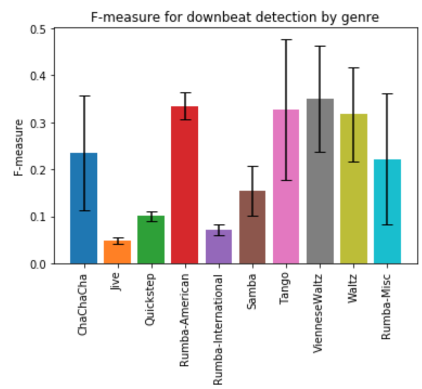
3.4 Downbeat Estimation and Metre Detection

After having determined a beat sequence, we first create an array consisting of multiple hypothesis for both the measure (either 3/4 or 4/4) and the start of first beat within the measure. This step is done by extracting a subset of the beat sequence indexes with values retrieved every fourth or third step, starting from the first, second, third or fourth position, in the case of a 4/4 measure, or from the first, second or third position, in the case of a 3/4 measure, yielding a total of seven hypothetical beat sequence indexes. We then use this indexes to extract a subset of the novelty function, one per each hypothesis. Finally, for each hypothesis we compute the mean and retrieve the one with the highest value. We then get a subset of the beat sequence that matches the best hypothesis.

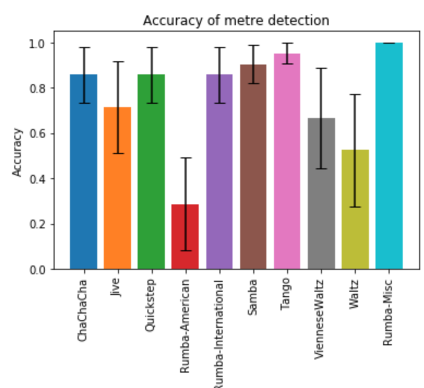
4 Evaluation



(a)



(b)



(c)

Fig. 1: Evaluation process for (a) beat tracking, (b) downbeat tracking and (c) metre detection.

Using a subset of the ballroom dance dataset, consisting of 30 random examples per genre[1], the algorithm was evaluated by computing the F-measure for estimated beat and downbeat sequences using the Python package *mir_eval*, comparing with annotated data. Furthermore, metre detection was assessed by computing an accuracy factor. The results are shown in Figure 1, presenting values of mean and variance of the considered subset. Here we can infere that the beat tracking algorithm (a) is able to achieve satisfactory results . The genre *ChaChaCha* yielded the best results, whilst *Samba* presented the worst. Concerning downbeat tracking (b), the algorithm was not able to produce robust results, despite having estimated good sequences when judged by subjective listening. Finally, meter detection proved to work reliably (c), despite the low results obtained for the genre *Rumba-American*.

5 Limitations

The two main limitations of this approach are *the assumption that tempo is nearly constant throughout the song* (making it less flexible then the Predominant Local Pulse algorithm [3], for example), and t*he premise that beats correspond to strong onsets*. As for downbeat estimation, one limitation is that the procedure assumes that the first estimated beats have strong chances of being correctly predicted, which in most of the cases reveals itself as a false hypothesis. Furthermore, the algorithm is hard coded to only consider 3/4 or 4/4 examples, due to the properties of the ballroom dance data set used in this assignment.

References

[1] Matthew E.P. Davies and Sebastian Böck. "Temporal Convolutional Networks for Musical Audio Beat Tracking". In: *European Signal Processing Conference*. 2019.

[2] Daniel P.W. Ellis. "Beat tracking by dynamic programming". In: *Journal of New Music Research* 36.1 (2007), pp. 51–60. ISSN: 09298215. DOI: 10.1080/09298210701653344.

[3] Meinard Müller. *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. 1st. Springer Publishing Company, Incorporated, 2015. ISBN: 3319219448.

[4] Julius Richter. "Style-Specific Beat Tracking with Deep Neural Networks". Master Thesis. 2019.

[1] Except for the genre *Rumba-American*, which only had seven available examples