Music Informatics - Assignment 1

Brendan O'Connor

March 2021

1 Introduction

For this assignment I chose to stick with the deterministic signal processing methods discussed in the lectures. The beat detection algorithm presented here is based on the analysis of onsets derived from a combination of onset detection functions (ODFs). Auto-correlations of these functions were used to determine the tempo of the music, while a pulse-train signal of the tempo's frequency was cross-correlated with the ODF to locate the first beat. A set of rules allowing for different degrees of tolerance, inertia and reactivity were devised to allow the beat detector to adapt to tempo fluctuation, human performance and silent beats. Section 2 describes in detail how each stage of the beat tracking process was designed. Section 3 presents the results and discusses the pitfalls in beat detection, accompanied with some reflection and suggestions on how the algorithm can be improved upon.

The file `beatDetection_Walkthrough.ipybn` accompanying this submission displays a concise walk-through of the algorithm, and compliments Section 2, providing the reader with visualisations of the data as well as code blocks. Some of the steps featured in this notebook were not used in the final draft of the beat detection algorithm, but did prove to be useful in verifying and understanding how the data was analysed and how the beat detector reacted during development. The file `beatDetection_DatasetIteration.ipybn` cycles through the entire dataset and presents distributions of scores for the various evaluation metrics. Finally, the file `beatDetection_OneLineFunction.ipybn` presents the algorithm in its fourth cell as the requested one-line function `beats = beatTracker(inputFile)`.

2 Implementation

Please refer to the **beatDetection_Walthrough.ipybn** file as you read through this section. The Ballroom dataset provided for this assignment was sampled 44.1kHz. ODFs (which included root-mean-square, high-frequency content, spectral flux, complex domain, rectified complex domain, phase deviation, and weighted phase deviation) were derived using the material covered in lectures, and based of FFT framerates of 100fps, which although an unusual number to choose in computer science, proved to be useful for the transparent relation between sample durations and seconds - a good enough reason for the purposes of experimental research. ODF values were standardised to unit variance and a mean of 0 in order to normalise their distributions and make threshold determinations more manageable between them. It was empirically found that different onsets seemed to perform best for only certain types of music and there was no ODFs that performed noticeably better than the rest. In particular however, Spectral Flux was usually favoured because of its ability to harnass information from the frequency domain, reliance on large changes of energy across the frequency bins that would normally be less salient through other means, and generally well-distinguished onsets that it generated. The first derivative of spectral flux was used in the final ODF.

Irrespective of the choice in ODFs, the remainder of the algorithm remains the same. First the selected ODF is compared against a designated threshold to remove lower-level noise in the signal and derive a series of distinguishable peaks. Neighbouring onset peaks that are closer than 50ms to one another are filtered out, leaving the strongest peak remaining (Dixon, 2001). Again this removes further noise and reduces the chance of mistaking neighbouring peaks for a beat-marking onset.

The research of Laroche (2003); Peeters (2006) was consulted to develop tempograms, where it was suggested that a combination of FFTs and ACFs were used to compliment one another. Peeters (2006) suggests that the analysis window for the STFT should be capable of representing at least two measures (in the case of 120bpm, this would be 8 seconds), with a hop size of 0.5 seconds. However, by using an onset signal that was sampled at 100 frames per second and a window duration of 8 seconds, the tempo resolution was not high enough to represent tempo accurately enough. It made more sense to perform an FFT over the entire song, providing us with more frequency bins and therefore better resolutions. Of course the downside is that one would get only a single tempo value for the whole song. The auto-correlation method however, did not pose to be too much of an issue to implement, and yielded several candidates for the sample-delay value *tau*, representing the ideal bpm. The best *tau* value was selected by choosing the strongest correlation that lay between the reasonable pulse values of 60-200bpm (in most cases there was only one value within this range anyway). This frequency range was suggested by Alonso, David, and Richard (2004) and provides a higher upper bound to take into consideration some of the faster-paced music in the dataset.

In order to determine which of the initial onsets are the first downbeat, an adaption of the pulse-train method (Alonso et al., 2004; Scheirer, 1998) was used, where the onset function is cross-correlated with a pulse-width modulation signal with a frequency equivalent to the *tau* value and a narrow duty-cycle (the width of which was empirically determined to be 10 samples long (equating to 50ms left and right of the focused sample). This is computationally efficient as only the indices of the offsets are evaluated for cross-correlation. The onset index that causes the strongest correlation between the two signals is considered to be the first downbeat in the track.

For tracking the beat throughout the music, Alonso et al. (2004) describe methods that assume the location of the next sample will be the previous beat time plus the *tau* value, accompanied with a 'tolerance window' to anticipate the imperfect nature of human performers and expressive playing. A second, wider tolerance window is also considered, should no onsets be detected in the 'performance tolerance' window. The wider window considers intentional fluctuations in tempo and is asymmetrically distributed to the left and right of the next predicted beat time (Dixon, 2001). The beat tracker will accept the location of a detected onset without considering a change in tempo if the onset is within the its performance tolerance window. If a beat is only detected in the tempo-deviation window, the beat detector accepts the onset as the beat and adjusts its tempo using some constant weight α that combines its reactive state with inertia. If there is no onset detected in either window, the beat is assumed to be silent, but assigned none-the-less. However if the track is finished, the last assigned beats without onsets are retrospectively removed.

3 Results and Discussion

The results of this implementation were evaluated using the `mir_eval` library (provided in the **beatDetection_DatasetIteration.ipybn** file). Results are diverse, but are slightly skewed towards the lower scores, and the average F-score is 0.4. It is clear there is a great deal of room for improvement. We can now discuss some shortcomings of the beat tracker and hypothesize how we can improve upon it. Of course, these observations have not been deduced from the global evaluation metrics, and were more obtained during the development stage, when the beat trackers performance was observed over many different pieces of music. Beat times were saved to csv files and sonified together with the audio in Sonic Visualizer 2 (Cannam, Landone, & Sandler, 2010) for audible evaluation.

Firstly, some more dynamic controls would improve the beat tracker. The use of a tempogram would dynamically inform the beat tracker when it is to expect changes in tempo and could therefore offset its outer tempo-deviation windows appropriately. Threshold values could be automated based on the amount of energy in a local analysis window, allowing less salient onsets to be caught during quiet sections. Lowered thresholds however mean more onset density, and could cause neighbouring onsets to be mistaken for beat-marking onsets. A solution to this would be to have the tolerance windows of the beat tracker to be inversely proportional to the magnitude of the onsets.

It was noticeable that music with stronger offbeats (often accentuated by a snare) overshadowed the downbeats which were sometimes masked by other musical events leading to less salient spectral flux for onset detection. As a result, the beat tracker can sometimes consider this to be half the tempo. One solution might be to divide the audio signal into bands so that frequency bands can have their own onsets for snare and kick drums.

The cross-correlation method used to detect the first downbeat is compared to the first quarter of the music (7.5 seconds roughly). This fraction of the music was specified to avoid significant tempo-drift affecting correlations. However the smaller the cross-correlation segments, the more detrimental our assumption is that there are significant onsets in the beginning of the piece if this turns out to be untrue.

Music of the more orchestral and organically flowing variety typically consist of far-microphone recording techniques and instrumentation that are not heavy in the lower frequencies. For this reason, the HFC-based ODF may be well suited. This music also features generous expressive license in regards to tempo. An analysis of the spectral content could therefore assist the beat tracker in determining large to set its outer tempo-deviation windows.

References

Alonso, M., David, B., & Richard, G. (2004). Tempo and Beat Estimation of Musical Signals. , 6.

Cannam, C., Landone, C., & Sandler, M. (2010). Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proceedings of the international conference on Multimedia - MM '10* (p. 1467). ACM Press. doi: 10.1145/1873951.1874248

Dixon, S. (2001). Automatic Extraction of Tempo and Beat From Expressive Performances. , *30*(1), 39. doi: 10.1076/jnmr.30.1.39.7119

Laroche, J. (2003). Efficient Tempo and Beat Tracking in Audio Recordings. , *51*(4), 226–233.

Peeters, G. (2006). Template-Based Estimation of Time-Varying Tempo. , *2007*(1), 067215. doi: 10.1155/2007/67215

Scheirer, E. D. (1998). Tempo and beat analysis of acoustic musical signals. , *103*(1), 588–601. doi: 10.1121/1.421129