Homework 2

Tempo estimation, beat/downbeat tracking, and meter recognition of audio and symbolic data

Li Su
Institute of Information Science, Academia, Taiwan
lisu@iis.sinica.edu.tw

In this homework we will implement algorithms for the following tasks: (1) compute the tempo of a song, (2) identify every beat/downbeat position of a song, and (3) identify the meters of a song.

The definitions of beat and downbeat have been mentioned in the course slides. *Meter* refers to the regularity of repeating patterns of music. In a narrow sense, meter here refers to the relationship between beats and bars. For example, time signature 3/4 means that each bar contains 3 beats, and each beat is a quarter note; therefore, its meter is 3-beats. The commonly seen meters in our everyday music are 3-beats, 4-beats or their multiples, while 5-beats and 7-beats are also used sometimes.

Prerequisite:

(1) Download the Ballroom dataset and annotation from:
https://drive.google.com/open?id=1Gk81pTyo65FIkUdR3inlNVWm72cqeaII
(2) Download the SMC beat tracking dataset and annotation:
http://smc.inesctec.pt/research/data-2/
(3) Download the JCS beat and downbeat tracking dataset and annotation:
https://drive.google.com/drive/folders/18OP9LU8YflZtkULOk7qLAZkdBY8cOQfn
(4) Download the madmom library, the state-of-the-art Python library for beat and downbeat tracking:
https://madmom.readthedocs.io/en/latest/
(5) There are also Python resources available for tempo and beat computation:
https://bmcfee.github.io/librosa/generated/librosa.feature.tempogram.html
https://librosa.github.io/librosa/generated/librosa.beat.tempo.html
https://librosa.github.io/librosa/generated/librosa.beat.beat_track.html
(6) Or you could also implement the tempograms by yourself (see the lecture slides).

Task 1: tempo estimation

Q1 (30%): Design an algorithm that estimate the tempo of each clip in the Ballroom dataset. Assume that the tempo of every clip is constant. Note that your algorithm should output two predominant tempi for each clip: T_1 (the slower one) and T_2 (the faster one). For example, you may simply try the two largest peak values in the tempogram of the clip. The tempogram can be computed from `librosa.feature.tempogram`. The evaluation method of tempo estimation is as follows.

We need to compute a "relative saliency of T_1" defined by the strength of T_1 relative to T_2. It is to say, for the tempogram $F(t,n)$, we have the saliency $S_1 = F(T_1,n)/(F(T_1,n)+F(T_2,n))$ for tempo value t at a specific time at *n*. For an excerpt with ground-truth tempo *G*, the *P-score* of the excerpt is defined as

$$P = S_1 T_{t1} + (1 - S_1) T_{t2}$$

$$T_{ti} = \begin{cases} 1 & \text{if } \left|\frac{G - T_i}{G}\right| \leq 0.08 \\ 0 & \text{otherwise} \end{cases}, i = 1,2$$

Another score function is the "at least one tempo correct" (ALOTC) score, defined as

$$P = \begin{cases} 1 & \text{if } \left|\frac{G - T_1}{G}\right| \leq 0.08 \text{ or } \left|\frac{G - T_2}{G}\right| \leq 0.08 \\ 0 & \text{otherwise} \end{cases}$$

Compute the average P-scores and the ALOTC scores of the eight genres (Cha Cha, Jive, Quickstep, Rumba, Samba, Tango, Viennese Waltz and Slow Waltz) in the Ballroom dataset using your algorithm. The above process can all be found in the evaluation routine `mir_eval.tempo.detection`.

Note 1: if you want to use `librosa.beat.tempo` directly, you have to find some ways to let it output two tempi.

Note 2: the ground-truth tempo *G* of each excerpt could be obtained from the labeled beat sequence in the dataset. Given a beat sequence $\mathbf{b} = [\mathrm{b}_1, \mathrm{b}_2, \dots, \mathrm{b}_\mathrm{M}]$, the average tempo (in BPM) could be represented as $\mathrm{mean}(60/\mathrm{diff}(\mathbf{b}))$.)

Q2 (15%): Instead of using your estimated $[T_1, T_2]$ in evaluation, try to use $[T_1/2, T_2/2]$, $[T_1/3, T_2/3]$, $[2T_1, 2T_2]$, and $[3T_1, 3T_2]$ for estimation. What are the resulting P-scores? Discuss the results.

Q3 (5%): Using madmom to estimate tempo on the Ballroom dataset. Evaluate the performance using again the P-score and the ALOTC score. Note that madmom might output more than two tempi with their saliency values; you may select the output tempi with the two highest saliency values as the results. The saliency values should be normalized to 1 before computing the P-score.

Task 2: using dynamic programming for beat tracking

Q3 (30%): Using `librosa.beat.beat_track` to find the beat positions of a song. Evaluate this beat tracking algorithm on the Ballroom dataset. The F-score of beat tracking is defined as $F \coloneqq 2PR/(P+R)$, with Precision, P, and Recall, R, being computed from the number of correctly detected onsets TP, the number of false alarms FP, and the number of missed onsets FN, where $P \coloneqq TP/(TP+FP)$ and $R \coloneqq TP/(TP+FN)$. Here, a detected beat is considered a true positive when it is located within a tolerance of ± 70 ms around the ground truth annotation. If there are more than one detected beat in this tolerance window, only one is counted as true positive, the others are counted as false alarms. If a detected onset is within the tolerance window of two annotations one true positive and one false negative are counted. Evaluation process can be done by `mir_eval.beat`. Similarly, please compute the average F-scores of the eight genres in the Ballroom dataset.

Q4 (15%) Also use this algorithm on the SMC dataset and the JCS dataset. Compare the results to the Ballroom dataset. Could you explain the difference in performance?

Note: since these datasets might be large for your computer, running the JCS dataset is an optional choice for you, and I will take it as bonus points. However, please run them as early as possible in order not to be late in submitting the homework.

Q5 (5%): Use any function in `madmom.features.beats` for beat tracking and downbeat tracking in the Ballroom and the JCS dataset, and for beat tracking for the SMC dataset (note: there is no downbeat annotation in the SMC dataset). For downbeat tracking, also compute the same F-score using tolerance of ± 70 ms. Compare the results to the ones in Q3 and Q4. How much improvement it gains?

Task 3: meter recognition

Q6 (20%): Although `madmom.features.beats` is the state-of-the-art downbeat tracker, one issue is that it assumes the meter of a song is constant. Actually, it assumes the meter a song is only one of the followings: 2-beats, 3-beats, 4-beats, 5-beats, 6-beats, and 7-beats, and the type of meter should be given by the parameter `beats_per_bar`. You may find some clips in the JCS dataset have time-varying meters, and madmom might perform not well in these clips. Could you design an algorithm to detect the instantaneous meter of a song? Test the algorithm on the clips in the JCS dataset, and report frame-wise accuracy. Can such an algorithm help improve the performance using madmom beat and downbeat tracker?

The deadline for this homework is June 16 (Tue).