Music Informatics Coursework 1: Beat Tracking

INTRODUCTION

This project implements beat tracking with dynamic programming, as described in Ellis, 2007[1]. While there are state of the art deep learning models to solve the problem of beat tracking, the Ellis approach breaks down the problem into logical steps (onset detection, tempo estimation, then beat tracking). The multi-step, white box approach has several advantages: (i) it requires little to no training data, (ii) beats detected by the model are highly explainable, and (iii) it serves as a useful exercise to practice and think about music information retrieval from an analytical perspective.

After getting a set of detected beats, a naive downbeat detection logic is also applied to estimate downbeats.

This report details the approach taken, evaluates its performance, and discusses its merits and room for improvement.

THEORY

Ellis, 2007 proposed a method for beat tracking based on the idea that the problem of identifying beats can be formulated as a cost function, and then minimised algorithmically. The cost function takes into account 2 costs: (i) the strength of onsets, and (ii) the temporal consistency of the beats. In other words, beats are more likely to occur where it is an onset, and when it is consistent with the tempo of the song. Formally, it is defined as:

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

where {ti} is the sequence of N beats found by the tracker, O(t) is the onset strength envelope, and F(Δt, τp) is a function that measures the temporal consistency between beats given τp as the target tempo.

In the paper, a squared-error function is used for F:

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

and it was found that α=680 was the optimal setting for the weight of the temporal consistency term in the cost function.

Based on the definition of the cost function above, we can see that there is a sequential dependency in this approach - the estimated tempo is dependent on the onset strength envelope, and the beat tracking component is dependent on both the estimate tempo and onset strengths. The fact that onset strength envelope is used twice in the steps also implies that the overall approach is highly dependent on the onset detection function. Furthermore, in the temporal consistency term, a global tempo is assumed, which implies that the tempo throughout each piece of music must be relatively consistent.

IMPLEMENTATION

The Ellis beat tracking approach has 3 steps:

1. Onset strength envelope
2. Tempo estimation
3. Beat tracking based on (1) and (2)

For onset strength, a simple perceptual model was used. The steps described in the paper are as follows:

1. Resample the audio to 8kHz
2. Obtain a Short-time Fourier Transform spectrogram using a window of 32ms and hop size of 4ms.
3. Map a filter of 40 Melbanks to the spectrogram to obtain a Mel-spectrogram, so that the frequency scale are aligned with human auditory perception.
4. Calculate the time-derivative along each band, and perform half-wave rectification on the derivative signal
5. Sum up the positive differences across all frequency bands
6. Pass the signal through a high-pass filter with 0.4Hz cutoff
7. Apply a gaussian convolution with a window of 20ms on the signal
8. Normalize the signal by dividing the result over its standard deviation

In the code implementation, some changes and assumptions were made. Step 6 was skipped, since it was not clear how to perform a frequency-based cutoff over dB-based signal. In Step 7, the standard deviation off the gaussian window was not indicated in the paper. A value of 10 was arbitrarily chosen based on the resultant onset strength envelope.

For tempo estimation, an autocorrelation method was described in the paper. To reflect a known human bias towards tapping close to 120 BPM, a perceptual weighting window was applied so that periodicity peaks further from this bias were down weighted. Formally, the tempo period strength was defined by:

$$TPS(\tau) = W(\tau) \sum_{t} O(t) O(t - \tau)$$

where W(τ) is a Gaussian weighting function on a log-time axis:

$$W(\tau) = \exp\left\{-\frac{1}{2}\left(\frac{log_2 \tau / \tau_0}{\sigma_\tau}\right)^2\right\}$$

Given the subjective nature of tempo, the paper also incorporated the idea that ground-truth tempos could fall within 0.5 or 0.33 times

of the estimated tempo. To account for this, they proposed 2 further TPS functions that resamples TPS to half or one-third of its original length:

$$TPS2(\tau) = TPS(\tau) + 0.5TPS(2\tau) + 0.25TPS(2\tau - 1) + 0.25TPS(2\tau + 1)$$

$$TPS3(\tau) = TPS(\tau) + 0.33TPS(3\tau) + 0.33TPS(3\tau - 1) + 0.33TPS(3\tau + 1)$$

They found that choosing the tempo at the maximum point of TPS2+TPS2 performed better for tempo estimation, using a value of $\sigma\tau = 0.9$ octaves.

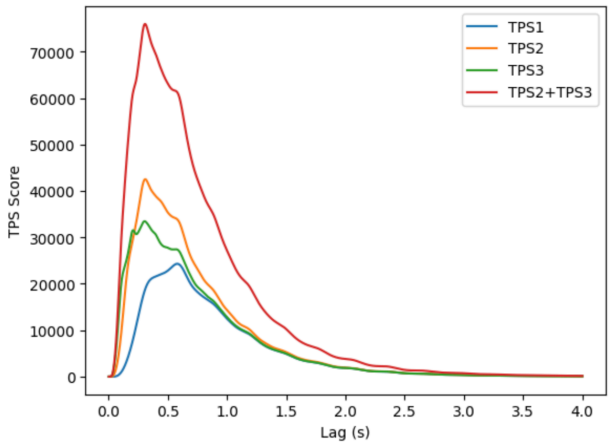

Figure 1: TPS curves across lag times for each TPS function.

In Figure 1 above, the chosen tempo for TPS2+TPS3 (around 0.30s) differs from that which have been chosen for TPS (around 0.58s) matched the actual annotated tempo.

The code implements tempo estimation exactly as described in the paper.

In the paper, the Matlab code for the dynamic programming search algorithm for beat tracking is provided. In the code implementation, this code is converted into Python. The DP search algorithm essentially iterates through each time frame, calculating the optimum cost C based on the onset strength and tempo consistency score at each frame. The index which contains the highest cumulated value of C is selected, and a backtrace is performed on that index, tracking its set of beats back to the start.

While the Ellis paper stops at beat detection, this project attempts to implement downbeat detection is performed over the identified beats. A naive approach is taken, based on the idea that downbeats occur where most energy is present. Again, the onset strength envelope provides a fair indicator of the energy levels in the audio, and is used for downbeat detection. Another important factor in downbeat detection is the metrical structure of the piece. In this implementation, we assume that the metrical structure is known in advance. The logic for downbeat detection is described below:

1. Get the onset strength at all detected beats

2. Given the metric level, sum up the onset strength for each possible set of downbeats. For example, with a metric of 4/4, there would be 4 possible sets of downbeats with indices {0, 3, 7, 11, 15, 19, …}, {1, 4, 8, 12, 16, 20, …}, {2, 5, 13, 17, 21, …}, and {3, 6, 14, 18, 22, …}.

3. Select downbeats based on the set of downbeats with the highest total onset strength.

EVALUATION

The beat tracking approach described above is then evaluated on the full Ballroom dataset. Before looking into the details of evaluation, we first look at the nature of the dataset.

Dataset

The Ballroom dataset consists of 680 audio files, across 10 categories of Ballroom styles. Most styles are annotated in 4/4 metre, except for Viennese Waltz and Waltz, which are in 3/4 time. The audio files capture the first 30 seconds of each piece. To remove interference from the introduction of each song, we trim all beats to remove the first 5 seconds before evaluation.

Tempo Estimation Metrics

category	actual_tempo	est_tempo	tempo_diff	actual_tempo_std	actual_vs_est_tempo_ratio
ChaChaCha	122.303650	188.042909	65.739259	1.949749	0.717728
Jive	166.114688	167.777585	1.662897	3.021075	0.990305
Quickstep	204.292185	198.678329	-5.613856	5.252537	1.029239
Rumba-American	125.478501	180.688129	55.209628	2.708514	0.724243
Rumba-International	100.032235	196.746798	96.714563	1.894954	0.508678
Rumba-Misc	94.170421	186.866710	92.696289	1.927862	0.503550
Samba	100.334682	195.326529	94.991847	1.479404	0.514994
Tango	127.463714	159.533016	32.069302	3.327960	0.836288
VienneseWaltz	178.089268	177.667092	-0.422176	5.085680	1.002425
Waltz	85.879105	173.758324	87.879219	2.933090	0.494296

Table 1: Tempo metrics by dance category

In general, tempo estimation worked well for most dance categories. From Table 1 above, looking at the tempo differences (tempo_diff), we could estimate the tempo for Jive, Quickstep, and Viennese Waltz to closely match the actual annotated tempo. For Rumba-International, Rumba-Misc, Samba, and Waltz, we were able to obtain almost exactly 2-times of the annotated tempo, as quantified by actual_vs_est_tempo_ratio, which measures the actual tempo/the estimated tempo. As for ChaChaCha, Rumba-American, and Tango, the tempo ratio is neither 0.5 nor 0.66, implying that there might be a fair mix of match, 2-times, and 1.5-times the annotated tempo.

To test the assumption of a global tempo, we also calculated the standard deviation of the annotated tempo (actual_tempo_std). This was done by taking the standard deviation of the time between all consecutive beats, and then converting it to beats per minute. For most genres, the global tempo assumption held well, since the tempo within the piece did not vary by much, with a standard deviation of around 2-3 BPM. However, the tempo for Quickstep and Viennese Waltz did vary by an average of 5 BPM. Despite this, the tempo estimation still performed well on both categories, suggesting that a 5BPM standard deviation in tempo might not be too significant.

Beat Estimation Metrics

In general, our beat estimation scores fairly with a 0.69 f-measure, as seen in Table 2. Furthermore, if we consider metrics performance at various metric levels, the beat estimation has a maximum Cemgil score of 77% and AMLc/t scores well at 73% and 80% respectively. This shows that the approach might not be that great at identifying all the exact annotated beats, but could perform well at various metric levels.

f_measure	0.691739
cemgil_score	0.616754
cemgil_max	0.766748
goto	0.351003
p_score	0.628919
info_gain	0.450553
CMLc	0.338683
CMLt	0.361343
AMLc	0.733653
AMlt	0.799406

Table 2: Average score for each evaluation metric, across the entire dataset.

category	f_measure	cemgil_score	cemgil_max	goto	p_score	info_gain	CMLc	CMLt	AMLc	AMlt
ChaChaCha	0.744341	0.698917	0.900818	0.378378	0.655933	0.615170	0.376904	0.376904	0.905782	0.924745
Jive	0.805505	0.744120	0.783073	0.800000	0.801633	0.596082	0.782899	0.787621	0.832258	0.845704
Quickstep	0.800684	0.677446	0.691100	0.658537	0.775256	0.309473	0.588135	0.725622	0.588736	0.727579
Rumba-American	0.530732	0.435910	0.524309	0.142857	0.525581	0.221858	0.190700	0.230383	0.261645	0.384727
Rumba-International	0.633239	0.575757	0.863047	0.000000	0.481913	0.533375	0.000000	0.000000	0.860725	0.913166
Rumba-Misc	0.608150	0.530005	0.798144	0.000000	0.471751	0.442856	0.000000	0.000000	0.793465	0.874130
Samba	0.578510	0.536050	0.799040	0.000000	0.450347	0.526200	0.000000	0.000000	0.805965	0.814548
Tango	0.660283	0.569721	0.639434	0.465116	0.656711	0.345560	0.460095	0.476707	0.527455	0.570994
VienneseWaltz	0.956947	0.870953	0.872921	0.923077	0.949657	0.495267	0.899389	0.938662	0.899389	0.939067
Waltz	0.519559	0.430845	0.649865	0.000000	0.458602	0.285704	0.000000	0.000000	0.570314	0.731899

Table 3: Average score for each evaluation metric, grouped by each dance category.

Using f-measure as a metric, the beat estimated scores well for VienneseWaltz (0.96), Jive (0.81), and Quickstep (0.80), while it does poorer on Waltz (0.52), Rumba-American (0.53) and Samba (0.58). When considering the scores at various metric levels, VienneseWaltz, ChaChaCha, and Rumba-International have high maximum Cemgil scores, and AMLc/t scores. However, it performs very poorly for Rumba-American.

Looking into the individual audio files for Rumba-American, a few things are of note. First, the Rumba-American category only has 7 examples, and do not affect the overall metric much. 3 pieces from GloriEstefan _MiTierra and AnaBelen_Veneo performed the worst, had odd tempo estimations that were about 1.4 times the original tempo. The GloriEstefan pieces had long 16s intros, where the percussive beat was only heard 10s into the piece.

Furthermore, when comparing the annotated beats to the onset strength envelope, we find that the annotated beats aren't always at the peaks of the onset strength envelope. As seen in Figure 2 below, for GloriaEstefan_MiTierra-06, the onset peaks are not more distinct at beats than at non-beats, which might explain why the model was unable to pick out the correct tempo, and seemed to simply estimate most onset peaks as beats.

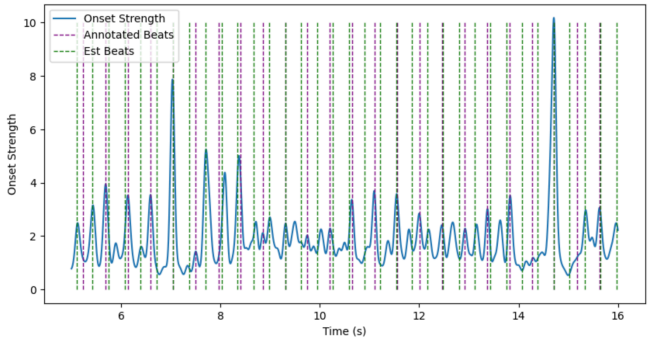

Figure 2: Annotated VS Estimated beats along the Onset Strength Envelope for GloriaEstefan_MiTierra-06 (Rumba-American).

Comparing this to a Rumba-International piece, GloriaEstefan_MiTierra-01, in Figure 3, we see that most of the onset strength peaks fall within the annotated beats, allowing for a more accurate tempo estimation, and consequently, better estimated beats.

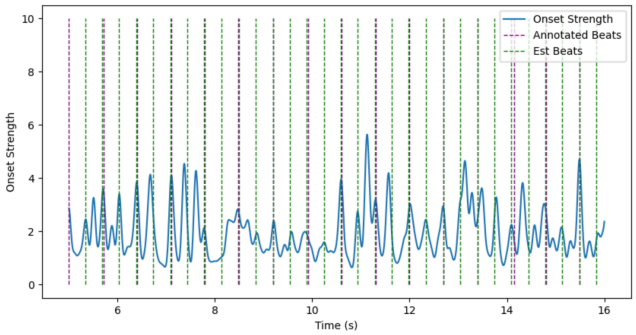

Figure 3: Annotated VS Estimated beats along the Onset Strength Envelope for GloriaEstefan_MiTierra-01 (Rumba-International).

Downbeat Estimation Metrics

f_measure_downbeat	0.290554
cemgil_score_downbeat	0.257330
cemgil_max_downbeat	0.365534
goto_downbeat	0.153295
p_score_downbeat	0.384788
info_gain_downbeat	0.637430
CMLc_downbeat	0.177989
CMLt_downbeat	0.178732
AMLc_downbeat	0.445822
AMlt_downbeat	0.451529

Table 4: Average score for downbeat estimation, across the entire dataset

As shown in Table 4 above, the downbeat estimation performed poorly, with an F-measure of 0.30. Accounting for different metrical levels, the maximum Cemgil score was 37%, and AMLc/t were both 45%. This indicates that then downbeat estimation could be slightly better than chance, since picking a set of downbeats at random would theoretically yield a 25% accuracy. The downbeat metrics are also constrained by the beat estimation metrics. When looking at AMLt, this means that we initially identified 80% of beats at all metric levels, and then identified 45% of down beats at all metric levels.

category	f_measure_db	cemgil_score_db	cemgil_max_db	goto_db	p_score_db	info_gain_db	CMLc_db	CMLt_db	AMLc_db	AMlt_db
ChaChaCha	0.134180	0.125036	0.198292	0.072072	0.204396	0.743413	0.080438	0.080438	0.244115	0.246074
Jive	0.068118	0.062378	0.089179	0.116667	0.187714	0.805035	0.166287	0.167212	0.237304	0.244468
Quickstep	0.361380	0.308133	0.462969	0.329268	0.471522	0.557124	0.443642	0.449291	0.637350	0.643933
Rumba-American	0.271201	0.231234	0.300144	0.142857	0.525230	0.397142	0.224490	0.224490	0.280390	0.280390
Rumba-International	0.215745	0.197977	0.290990	0.000000	0.271667	0.678754	0.000000	0.000000	0.301744	0.301744
Rumba-Misc	0.284051	0.248393	0.368844	0.000000	0.334854	0.639435	0.000000	0.000000	0.416128	0.416128
Samba	0.291804	0.271407	0.401498	0.000000	0.382506	0.678212	0.000000	0.000000	0.418087	0.418087
Tango	0.209330	0.178972	0.387472	0.139535	0.346367	0.551598	0.165947	0.165947	0.423540	0.423540
VienneseWaltz	0.811901	0.740519	0.740963	0.800000	0.818249	0.745257	0.817065	0.817065	0.891527	0.910604
Waltz	0.309619	0.256457	0.382805	0.000000	0.447013	0.465833	0.000000	0.000000	0.484185	0.498334

Table 5: Average scores for downbeat estimation, grouped by each dance category.

When split by dance category, the downbeat estimation only performed well on Viennese Waltz. It performed most poorly on Jive, ChaChaCha, Rumba-American, and Rumba-International.

Interestingly, Jive and ChaChaCha had relatively high beat estimation scores.

Looking into the audio files for Jive, we see that the annotated down beat was often not at the strongest onset strength. For example, in Figure 4 below, for the Macumba-14 track, our model estimated down beats at the 2nd beat, which often had higher onset strength peaks than the annotated down beats. Listening to the audio, the track had distinct snare sounds at every 2nd and 4th beat, which explains why the onset strength was higher at the 2nd beat compared to the first.

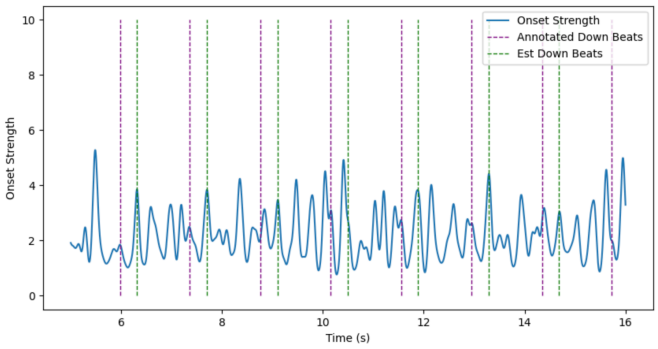

Figure 4: Annotated VS Estimated down beats along the Onset Strength Envelope for Macumba-14 (Jive).

DISCUSSION

The Ellis beat detection approach performs fairly well on the Ballroom dataset, especially since the parameters defined in the paper were tuned on MIREX-2006 beat tracking data, which is a small set of various song genres.

From the results, it is clear that the beat detection component is heavily dependent on both the onset strength envelope and the estimated tempo. With fairly accurate tempo, the detected beats will work well. However, if the tempo is not estimated well, then the beats detected will fall short. While the global tempo assumption seemed like a huge assumption to make, it works fairly well for Ballroom songs, and we observe that a small variance of 5 BPM does not affect the beat tracking performance.

The detected beats and tempo are also are heavily dependent on the strength of onsets, an assumption which might not always hold true. For example, the strongest energy in a song might not always be on the down beat, as observed in the Jive example above. Also, the onset strength envelope might not always have peaks with distinctly higher energy at the beats, especially if songs have quieter percussions. The onset strength peaks also might not always be aligned with the beats when there are many parts in the song, such as brass instruments and vocals.

Due to the limitations above, extending the approach to include downbeat detection did not work very well. The downbeat estimation also required prior knowledge of the song's meter, which might not always be available.

FUTURE IMPROVEMENTS

Since the onset strength envelope seems to be the limiting factor in most of the problems faced, perhaps testing out different onset detection functions would help to improve the performance of the beat tracker. Also, given its sensitivity to noise, it might be helpful to first remove vocals, and split the tracks into different sources (e.g. percussions, brass instruments, string instruments), before running the beat tracking system on each part. The final beats could be determined by a voting or weighted approach, where beats from percussion tracks are given higher weights than those from brass instruments for example.

REFERENCES

[1] Ellis, D. P. (2007). Beat tracking by dynamic programming. *Journal of New Music Research*, *36*(1), 51-60.