State-of-the-art Beat Tracking

Sevag Hanssian

MUMT 621, Winter 2021

March 30, 2021

madmom

madmom is an open-source Python MIR library from the Institute of Computational Perception at the Johannes Kepler University,[1] containing implementations of the many beat tracking works of Sebastian Böck and colleagues

1. Sebastian Böck et al. 2016. "madmom: a new Python Audio and Music Signal Processing Library." In *Proceedings of the 24th ACM International Conference on Multimedia,* 1174–1178. Amsterdam, The Netherlands, October. https://doi.org/10.1145/2964284.2973795. https://github.com/CPJKU/madmom.

State-of-the-art results

The RNNBeatProcessor[2] → DBNBeatTrackingProcessor[3] algorithm achieved the best results in the last* **M**usic **I**nformation **R**etrieval **E**valuation e**X**change audio beat tracking challenge[4]

*: 2020 did not include beat tracking submissions

2. Sebastian Böck and Markus Schedl. 2011. "Enhanced Beat Tracking with Context-Aware Neural Networks." In *Proceedings of the 14th International Conference on Digital Audio Effects, DAFx 2011.* September.

3. Sebastian Böck, Florian Krebs, and Gerhard Widmer. 2015. "Accurate Tempo Estimation Based on Recurrent Neural Networks and Resonating Comb Filters." In *Proceedings of the 16th International Society for Music Information Retrieval Conference,* 625–631. Málaga, Spain: ISMIR, October. https://doi.org/10.5281/zenodo.1416026.

4. https://www.music-ir.org/mirex/wiki/2019:MIREX2019_Results

State-of-the-art results

Algorithm	F-Measure	Cemgil	Goto	McKinney P-score	CMLc	CMLt	AMLc	AMLt	D (bits)	Dg (bits)
SB1	52.1352	41.2588	17.5115	60.2857	27.3747	35.2810	33.7466	47.4101	1.3555	0.3861
CD2	33.6605	26.2949	6.9124	45.1828	9.8778	13.1236	17.9898	29.4810	0.8081	0.1172
CD1	30.3524	23.4505	3.2258	41.9394	4.9259	6.5798	14.7409	26.8531	0.7077	0.0888
GBK1	39.8601	29.7131	12.9032	54.7260	22.8996	35.5078	36.3348	53.4312	1.5584	0.3712

(a) SMC (Sound and Music Computing at University of Porto) dataset results

Algorithm	F-Measure	Cemgil	Goto	McKinney P-score	CMLc	CMLt	AMLc	AMLt	D (bits)	Dg (bits)
SB1	58.6525	50.4149	0.0000	57.7060	6.1384	37.6827	6.4029	37.9650	0.6934	0.4972
CD2	49.6137	43.2195	0.0000	50.4279	4.7274	29.7076	4.9440	30.3144	0.5948	0.3769
CD1	42.0000	36.5974	0.3106	45.2193	3.2504	20.6707	4.4265	24.5303	0.5121	0.2770
GBK1	25.2988	16.9298	0.0000	21.1892	1.2915	4.5931	4.4308	13.6719	2.3046	0.4271

(b) MAZ (Chopin Mazurka) dataset results

Figure: MIREX 2019 audio beat tracking results

Step 1: RNNBeatProcessor

Traditional beat tracking algorithms use onsets, or the beginning of musical events,[5] as "potential beats," upon which further post-processing is applied to get beat locations[6]

The first algorithm, the RNNBeatProcessor[7], replaces traditional onset detection with a bidirectional Long Short-Term Memory (BLSTM) recurrent neural network to output frame-by-frame beat activations

---

5. Juan Bello et al. 2005. "A Tutorial on Onset Detection in Music Signals." *Speech and Audio Processing, IEEE Transactions on* 13 (October): 1035–1047. https://doi.org/10.1109/TSA.2005.851998.

6. Daniel Ellis. 2007. "Beat Tracking by Dynamic Programming." *Journal of New Music Research* 36 (March): 51–60. https://doi.org/10.1080/09298210701653344; Masataka Goto. 2002. "An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds." *Journal of New Music Research* 30 (September). https://doi.org/10.1076/jnmr.30.2.159.7114; Adam Stark, Matthew Davies, and Mark Plumbley. 2009. "Real-time beat-synchronous analysis of musical audio." September.

7. Böck and Schedl 2011.

BLSTM

Böck and Schedl (2011)'s justification for using BLSTM architecture:

- Most basic approach of neural networks is the multilayer perceptron (MLP) forming a feed forward (causal) neural network (FNN)
- Cyclic connections form a recurrent neural network (RNN), which can use past values but suffers from the vanishing gradient problem
- Long Short-Term Memory (LSTM) solves this problem. If the past *and* future of input are both important, Bidirectional LSTMs add a new layer presented to the network in reverse temporal order

Step 1: RNNBeatProcessor

- Originates from BLSTM onset detection algorithm[8]
- Modified to "suit the needs for audio beat detection by modifying the input representation and adding an advanced peak detection stage"[9]

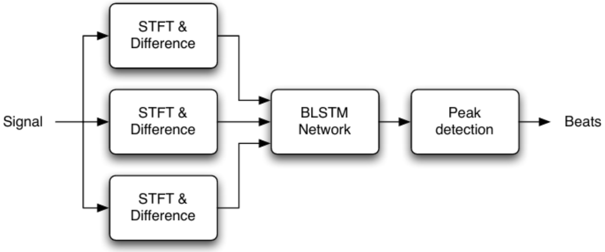


Figure: RNNBeatProcessor architecture

8. Florian Eyben et al. 2010. "Universal Onset Detection with Bidirectional Long Short-Term Memory Neural Networks." In *Proceedings of the 11th International Society for Music Information Retrieval Conference,* 589–594. Utrecht, Netherlands: ISMIR, August. https://doi.org/10.5281/zenodo.1417131.

9. Böck and Schedl 2011.

Step 2: DBNBeatTrackingProcessor

DBNBeatTrackingProcessor:[10]

- Use Böck and Schedl (2011)'s RNNBeatProcessor with multiple RNNs for different, heterogenous music styles
- Add an additional dynamic Bayesian network stage[11] which jointly infers the tempo and the beat phase from the beat activations of the RNN stage
- More efficient implementation of the DBN proposed by Böck, Krebs, and Widmer (2015)

10. Sebastian Böck, Florian Krebs, and Gerhard Widmer. 2014. "A MULTI-MODEL APPROACH TO BEAT TRACKING CONSIDERING HETEROGENEOUS MUSIC STYLES." In *Conference: 15th International Society for Music Information Retrieval Conference (ISMIR).* October.

11. Nick Whiteley, Ali Taylan Cemgil, and Simon J. Godsill. 2006. "Bayesian Modelling of Temporal Structure in Musical Audio." In *Proceedings of the 7th International Conference on Music Information Retrieval,* 29–34. Victoria, Canada: ISMIR, October. https://doi.org/10.5281/zenodo.1415138.

Step 2: DBNBeatTrackingProcessor

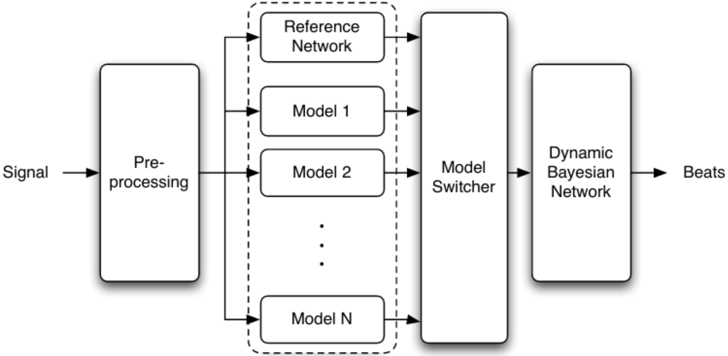
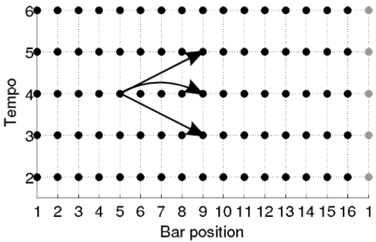


Figure: Multi-model RNN + DBN architecture

Bar-pointer model

Popular model that jointly models tempo and bar position, called the *bar pointer model*[12] is inefficient: "these algorithms share the problem of a high space and time complexity because of the huge state-space in which they perform inference"[13]

1. Non-linear tempo resolution matching humans
2. Only allow tempo transitions on beat locations



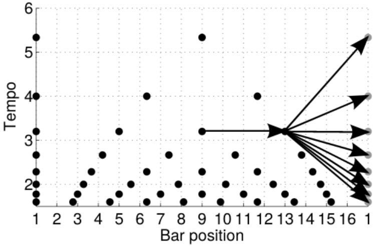


Figure: More efficient tempo-bar model

12. Whiteley, Cemgil, and Godsill, 2006.
13. Böck, Krebs, and Widmer, 2015.

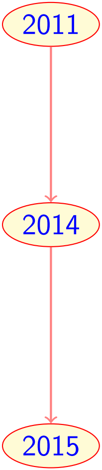
Complete example

Suggested usage from madmom docs:[14]

```
>>> proc = DBNBeatTrackingProcessor(fps=100)
<madmom.features.beats.DBNBeatTrackingProcessor object at 0x...>
>>> act = RNNBeatProcessor()('tests/data/audio/sample.wav')
>>> proc(act)
array([0.1 , 0.45, 0.8 , 1.12, 1.48, 1.8 , 2.15, 2.49])
```

14. https://madmom.readthedocs.io/en/latest/modules/features/beats.html

Conclusion



Böck and Schedl (2011) introduced state-of-the-art algorithm for beat tracking with a BLSTM recurrent neural network

Böck, Krebs, and Widmer (2014) improved the first algorithm by training multiple RNNs on different music styles and combining it with a dynamic Bayesian network for the final beat inference

Böck, Krebs, and Widmer (2015) introduced a new discretisation and tempo transition model that can be used as a drop-in replacement for variants of the bar pointer model