

$$\begin{aligned}
S &\rightarrow M_{b,s} \\
M_{b,s} &\rightarrow B_s \dots B_s \text{ (} b \text{ times)} \\
B_s &\rightarrow SB \dots SB \text{ (} s \text{ times)} \mid r \\
SB &\rightarrow r
\end{aligned}$$

Fig. 2. The PCFG’s rules, where b is the number of beats per bar, s is the number of sub-beats per beat, and r is any rhythmic pattern.

cally salient notes—in particular long or low notes—tend to align with strong positions higher up in the metrical tree. McLeod and Steedman [12] show that a lexicalized PCFG (probabilistic context-free grammar; LPCFG), which is able to draw on a wider rhythmic context, achieves state-of-the-art results for metrical alignment. Their model also has the property of incrementality (it performs its analysis in a single pass of the input and can output its top hypothesis at any time). This potentially allows it to be adapted to work on real-time tasks such as live accompaniment, unlike most other methods. However, they note that it can suffer from a lack of training data and performs poorly on unseen or uncommon rhythms. This is a significant issue: the training data required must be fully aligned with a metrical structure, which is uncommon. We build upon their method here, improving performance given limited training data.

2. PROPOSED METHOD

Our proposed method improves upon McLeod and Steedman’s [12] LPCFG’s performance on unseen rhythms by using online learning to create a piece-specific grammar that can be learned and used incrementally. The piece-specific grammar assigns a greater probability to a metrical alignment which results in rhythmic repetition (or near-repetition) occurring in similar metrical positions throughout a piece of music. Repetition is a well-known aspect of music, and much work has been done on musical pattern recognition and motif extraction (e.g. [13, 14, 15]). Notably, [16] investigates the link between musical patterns and meter, showing that pattern extraction can be used for meter detection, although, to our knowledge, it has not before been used explicitly for alignment.

The original LPCFG is described in Section 2.1, and the application of the new grammar is presented in Section 2.2. Code is available at www.github.com/apmcleod/met-align.

2.1. Lexicalized PCFG

2.1.1. The Grammar

The grammar itself, introduced in [17], is based on a relatively simple PCFG, whose rules are in Figure 2. The grammar rules create a tree similar in form to a metrical structure, with the addition of the terminal symbol r , which represents any rhythmic pattern. A beat may only be rewritten by a rhythmic pattern if it contains either a single note or a rest for its entire duration. The grammar is based on the principle that long notes are heard as rhythmically stressed, and aligning a detected stress with learned metrical stress patterns will allow for the detection of the underlying metrical structure of a given musical performance. However, a PCFG makes a strong independence assumption which is inappropriate for music. Specifically, a note can only be heard as “long” in comparison to the lengths of the surrounding notes, but a PCFG is unable to make such comparisons.

To solve this problem, the grammar is lexicalized, which refers to the assignment of a head to each node representing the most musically-important note (here, the longest note) beneath it in the tree. A head is written as $(d; s)$, where d and s represent the duration

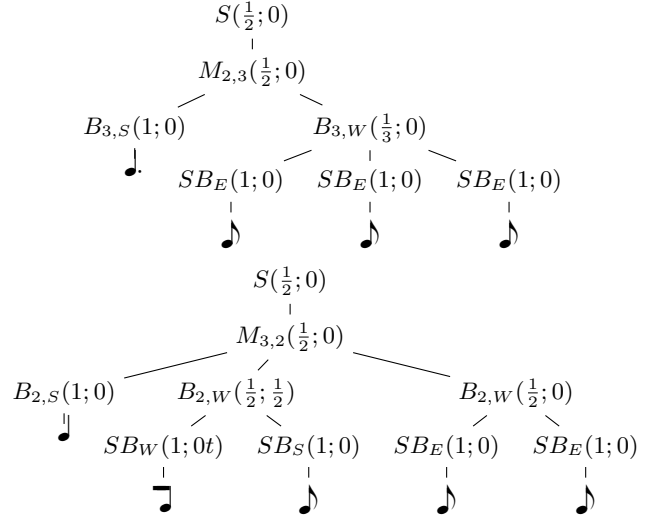


Fig. 3. Two possible parse trees of the rhythm ♪.♪♪ including lexicalization. Top: $\frac{6}{8}$, $p = 1.37 \times 10^{-3}$; bottom: $\frac{3}{4}$, $p = 6.78 \times 10^{-5}$.

and starting location of the longest note, each relative to the current node’s duration. An additional “ t ” is added if the note is tied into from a previous node. The heads provide a vital context which is missing from a standard PCFG. Following this head calculation, we assign each beat and sub beat a strength of either strong, even, or weak, (represented by subscripts of S , E , and W), based on its head and the heads of its siblings. Heads are ranked by duration (with tied notes being weaker than untied notes) and then by starting location. If all of the siblings’ heads are equal, the nodes are assigned equal strength. Otherwise, the nodes among a group of siblings with the largest head are assigned strong strength while the other nodes are assigned weak strength. Two example parse trees of the rhythm ♪.♪♪ (in $\frac{6}{8}$ and $\frac{3}{4}$ time) are shown in Figure 3. Notice the tied note in the second beat of the $\frac{3}{4}$ tree, and how it affects the surrounding heads and strengths. The probability of a parse tree is calculated using the standard LPCFG equations as described in [18], with an additional conditioning on the meter type and Good-Turing smoothing [19]. For example, the probability of the transition from the second beat of the $\frac{3}{4}$ tree from Figure 3 is the product of Equations (1)–(3). This results in probabilities of 1.37×10^{-3} for the $\frac{6}{8}$ tree and 6.78×10^{-5} for the $\frac{3}{4}$ tree given our training data.

$$P(B_{2,W} \rightarrow SB_W SB_S \mid M_{3,2}, (\frac{1}{2}; \frac{1}{2})) \quad (1)$$

$$P((1; 0t) \mid M_{3,2}, SB_W, (\frac{1}{2}; \frac{1}{2})) \quad (2)$$

$$P((1; 0) \mid M_{3,2}, SB_S, (\frac{1}{2}; \frac{1}{2})) \quad (3)$$

2.1.2. The Parser

The parser for the grammar is an HMM. Each state S_i contains a list of the times of the tatum of the i th bar (represented by $S_i.t$) and a metrical structure, which marks which of those tatum correspond to beats and sub beats. A tatum is the lowest-level pulse of a piece of music, and we implicitly model four tatum per sub beat. The transition function of the HMM is given by Equation (4), where $T(S_i)$ represents the tempo at the i th bar, and $P(S_i.t)$ is the probability of the state’s tatum—in particular how evenly spaced they are, calculated as the standard deviation of the time between pulses at each

level of the metrical tree. Both probabilities are modelled by Gaussians with learned standard deviations, similar to how Gaussians are used by Raphael [1] to model tempo deviation for rhythm parsing.

The emission function of the HMM is given by Equation (5). The first term measures how well each observed note’s onset aligns with a tatum (calculated as a Gaussian around each tatum location), and $P(\text{rhythm})$ is the probability of the resulting rhythmic parse tree (which is deterministic given a note set and a state) according to the learned LPCFG. Notice that each parse tree is monophonic, although our input is polyphonic. In practice, we perform voice separation [20] as preprocessing, creating one tree per voice per bar. $P(\text{rhythm})$ is then the product of the probabilities of each monophonic tree in a bar. The decoding of the model involves a modified Viterbi algorithm [21] with beam search, as detailed in [12].

$$P(S_i|S_{i-1}) = P(T(S_i)|T(S_{i-1}))P(S_i.t) \quad (4)$$

$$P(N_i|S_i) = P(N_i|S.t)P(\text{rhythm}) \quad (5)$$

2.2. Piece-specific Grammar

As noted in previous work, one of the main drawbacks of the LPCFG is that it suffers from a lack of data for training its grammar probabilities. In particular, while it performs well for more common time signatures with meter types it has seen often in the training data, it tends to struggle when it has only seen a given meter type once or twice during training. Here, we present a method to measure a given metrical alignment’s well-formedness without relying on any training data, while retaining the model’s incrementality. To do so, we use online learning with a piece-specific grammar (which we will refer to here as a local grammar) that encourages the model to align rhythmic repetition (or near-repetition) with the metrical structure.

During alignment, each hypothesis is initially assigned an empty local grammar. We add each parsed tree to its hypothesis’s local grammar, updating the grammar’s transition probabilities accordingly. After the first, we calculate the probability of each tree (the $P(\text{rhythm})$ term in the emission function from Equation (5)) as a weighted product of its probability given the global grammar G and its probability given its hypothesis’s current local grammar G_{local} (before that tree is added to the local grammar), as shown in Equation (6). Here, α is used to control the influence of the local grammar. A grid search on our training data resulted in a value of $\frac{1}{2}$.

$$P(\text{rhythm}) \propto P(\text{rhythm}|G)P(\text{rhythm}|G_{\text{local}})^\alpha \quad (6)$$

The local grammar drives the model towards a metrical alignment in which some level of the metrical tree aligns with a rhythmic repetition. However, it cannot make a distinction between different phases of the alignment—the global grammar must do that. For example, imagine a piece which has a repeated bar-length pattern of five notes. The local grammar will drive the model towards an alignment whose bar length is correct and whose beats and sub beats align consistently with the underlying pattern, but it will not be able to distinguish which of the five notes occurs on the downbeat. Still, this should eliminate a significant source of errors from the previous version of the model, and drive it towards an at least partially correct alignment, particularly for pieces with uncommon rhythmic patterns which are not seen by the global grammar during training.

Another potential approach could be to run the original model as usual (without the local grammar), and then use the local grammar to re-rank the resulting hypotheses, although this would no longer be incremental. Our results are slightly worse when performing this re-ranking, likely because the incremental processing allows the model

to filter out incorrect hypotheses sooner, thus freeing up space in the beam for more potentially good metrical alignments.

3. EVALUATION

3.1. Metric

For our evaluation metric, we use metrical F-measure as introduced by [12]. It tries to match each ground truth bar, beat, and sub beat node with those given by a metrical alignment. To count as a match, the beginning and end points of two nodes must match within 70 ms¹, though the level is not required to match. For example, a ground truth beat may match with an aligned sub beat, as long as their beginning and end points match. Unmatched ground truth and aligned nodes count as false negatives and false positives respectively.

3.2. Corpora

For training our proposed model as well as the original LPCFG [12], we use the miscellaneous corpus, released by [11], which contains a live performance portion used to train our beat tracking HMM (22 pieces) and a metronomic portion used to train our LPCFG probabilities (45 pieces). The corpus contains (mostly common practice era) pieces by various composers, and is quite small for the purpose of training the LPCFG for best performance. However, since we are investigating the local grammar’s performance when there is a lack of training data, we do not supplement it with additional data.

We evaluate the models on two corpora: (1) Bach, containing 63 metronomic MIDI files of Bach compositions—the 48 fugues from books one and two of the Well-Tempered Clavier (BWV 846–893; from www.musedata.org), and his 15 inventions (BWV 772–786; from www.imslp.org); and (2) piano-midi, containing 261 MIDI files of various common practice era composers from www.piano-midi.de. We ignore those containing time signature changes or irregular meters (time signatures besides $\frac{2}{x}$, $\frac{3}{x}$, $\frac{4}{x}$, $\frac{6}{x}$, $\frac{9}{x}$, or $\frac{12}{x}$), bringing the total down to 216. These MIDI files are pseudo-live performance: note velocities and tempo curves were manually edited by their creator to emulate live performance.

While a large corpus of actual live performance is ideal, a thorough evaluation of that type is left for future work. In previous work [12] a small subset (13 pieces) of CrestMusePEDB [23], which contains live performance MIDI, was used. Here, we do not do so, as the size of the subset is quite small which may lead to unclear results. Extending the evaluation to the full CrestMusePEDB corpus requires additional annotation because only beats and downbeats are labelled explicitly, not sub beats. The Vienna 4x22 piano corpus [24], which contains MIDI performance data of four compositions, each played by 22 different musicians, has the annotations we need. However, although this corpus is large (88 MIDI files total), the models’ scores on the 22 performances of a single composition are highly correlated (since their rhythmic patterns are identical), and it is difficult to derive any conclusion from its results.

For all corpora, we run the voice separation model from [20] with default settings as a preprocessing step.

3.3. Results

We compare our new method against two baselines: Temperley’s Bayesian model [11], and the original LPCFG model [12] without the new local grammar. The results are shown in Table 2, where it can be seen that the new local grammar leads to a consistent two to

¹The 70 ms window is taken from a widely used beat tracking metric [22].

Model	Bach	piano-midi
Temperley [11]	67.65	54.80
LPCFG [12]	77.67	44.91
+local (This work)	79.90	47.24

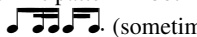

Table 2. The average metrical F-measure of Temperley’s method [11], the original LPCFG [12], and our newly proposed model (+local) on each corpus.

	$SB_S SB_W$	$SB_E SB_E$	$SB_W SB_S$
B_S	0.2571	0.6838	0.0591
B_E	0.0744	0.8989	0.0267
B_W	0.0905	0.5795	0.3300

Table 3. The LPCFG’s learned probabilities for every possible beat to sub beat transition in the context of a $\frac{4}{x}$ bar.

three point increase over the original LPCFG on both corpora. In the piano-midi corpus, the local grammar leads to an approximately normally distributed increase in performance among composers with more than 5 pieces ($\mu = 3.5$, $\sigma = 3.7$), ranging from -6 (Beethoven) to +13 (Tchaikovsky). On the Bach corpus, both versions of the LPCFG model outperform Temperley’s, while the opposite is true of the piano-midi corpus. All three models perform worse on the piano-midi corpus compared with the Bach corpus, which is not too surprising: the tempo curves applied to the piano-midi pieces make them much more difficult to align with a metrical structure than the flat tempo of the Bach corpus—not to mention that corpus’s wider variety of composers and musical styles. It was noted in previous work that the LPCFG requires more data of a given musical style than Temperley’s model [12], and the corpus’s variety—combined with the fact that we have used very little training data here on purpose—helps to explain why Temperley’s model outperforms the two LPCFG models on the piano-midi corpus.

Table 3 gives an intuition for why the wider context provided by the LPCFG might provide an advantage over a simpler model, when beat tracking is accurate. It shows the LPCFG’s learned probability of every possible beat to sub beat transition in a $\frac{4}{x}$ bar. The distributions are significantly different depending on the strength of the beat. Only through its context is the LPCFG able to learn such distributions, and it is precisely in cases like this that it adds value.

A specific example of a case in which the local grammar improves the metrical alignment dramatically is the 12th movement of Schumann’s *Kinderszenen*, Op. 15. It is in $\frac{2}{4}$ time, and contains a repeated bar-length rhythmic pattern in both the right and the left hands throughout the piece:  (sometimes ). This pattern, particularly its second beat, is quite uncommon in our training data, so the global grammar struggles with it. The local grammar, however, is able to recognize it. In the left hand, the pattern tends to begin on the downbeat of each bar, while in the right hand, the pattern usually begins on beat two. Nonetheless, each alignment is common enough for the local grammar to recognize, and regardless of what beat the pattern begins on, the trees from the beat level down are identical. Figure 4 shows the first three bars of the piece, along with the metrical alignment and F-measure of Temperley’s model (top), and the LPCFG without (middle) and with (bottom) the new local grammar. Temperley’s model predicts a $\frac{3}{8}$ meter which begins in phase, although drops in and out of phase throughout the piece due to syncopation and tempo changes. Without the local grammar, the LPCFG struggles, and predicts a $\frac{6}{16}$ meter, achieving a metrical F-measure of 0. With the local grammar, however, the model finds

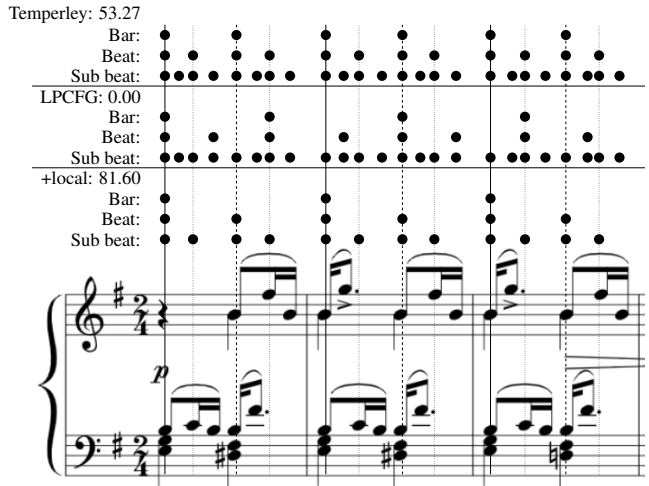


Fig. 4. The metrical alignment and F-measure of Temperley’s model (top), and the LPCFG without (middle) and with (bottom) the new local grammar on the first three bars of the 12th movement of Schumann’s *Kinderszenen*, Op. 15.

the bar-length pattern, and aligns it properly with a $\frac{2}{4}$ meter (with some minor phase errors), achieving a metrical F-measure of 81.60. The global grammar assigns the $\frac{6}{16}$ alignment a log probability of -1971.34 and the $\frac{2}{4}$ alignment a log probability of -2098.52. The local grammar, however, assigns the $\frac{6}{16}$ alignment a log probability of -1556.49 and the $\frac{2}{4}$ alignment a log probability of -1214.07. The weighted log probabilities in the +local model then become -1833.06 for the $\frac{6}{16}$ alignment and -1803.70 for the $\frac{2}{4}$ alignment.

This example, combined with our new model’s performance increase over the basic LPCFG, shows that leveraging rhythmic repetition for metrical alignment is a useful strategy. Such repetition can be detected directly, with no training data, and leads directly to increased performance on the task.

4. CONCLUSION

We have presented an improvement on an existing method for metrical alignment of MIDI performance, which cites a lack of training data and unseen rhythms as the source of many of its errors. To that end, our method uses online learning to adapt to unseen or uncommon rhythms and styles based on occurrences of patterns and rhythmic repetition in the music. Specifically, it uses a piece-specific lexicalized PCFG (LPCFG) which assigns a greater probability to rhythmic patterns that occur in metrical positions where they have previously been seen in a given piece. We have shown that rhythmic repetition is indeed useful for the task, and that our method offers increased performance without requiring any additional training data.

The current version of our model weighs the global and piece-specific grammars according to a parameter α . In future work, we intend to have the value of α change dynamically. For example, if a piece seems to match the rhythms from the global grammar well, α should be adapted to rely more on the global grammar. On the other hand, if a piece seems particularly repetitive, the model should rely more heavily on the local grammar. Furthermore, there seems to be room for improvement on the beat tracking portion of our model, given the drop in performance between the two corpora, and future work will attempt to improve beat-tracking to close that gap.

5. REFERENCES

- [1] Christopher Raphael, “A hybrid graphical model for rhythmic parsing,” *Artificial Intelligence*, vol. 137, no. 1-2, pp. 217–238, May 2002.
- [2] Eita Nakamura, Kazuyoshi Yoshii, and Simon Dixon, “Note value recognition for piano transcription using markov random fields,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 9, pp. 1846–1858, Sept. 2017.
- [3] Judith C. Brown, “Determination of the meter of musical scores by autocorrelation,” *The Journal of the Acoustical Society of America*, vol. 94, no. 4, pp. 1953, 1993.
- [4] Benoit Meudic, “Automatic meter extraction from MIDI files,” in *Journées d’informatique musicale*, 2002.
- [5] Douglas Eck and Norman Casagrande, “Finding meter in music using an autocorrelation phase matrix and Shannon entropy,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2005, pp. 504–509.
- [6] W. Bas De Haas and Anja Volk, “Meter detection in symbolic music using inner metric analysis,” in *ISMIR*, 2016, pp. 441–447.
- [7] Sebastian Böck, Florian Krebs, and Gerhard Widmer, “Joint beat and downbeat tracking with recurrent neural networks,” in *ISMIR*, 2016, pp. 255–261.
- [8] Simon Durand, Juan P. Bello, Bertrand David, and Gael Richard, “Robust downbeat tracking using an ensemble of convolutional networks,” *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 25, no. 1, 2017.
- [9] Nick Whiteley, A. Taylan Cemgil, and Simon Godsill, “Bayesian modelling of temporal structure in musical audio,” in *ISMIR*, 2006.
- [10] David Temperley, *Music and Probability*, The MIT Press, 2007.
- [11] David Temperley, “A unified probabilistic model for polyphonic music analysis,” *Journal of New Music Research*, vol. 38, no. 1, pp. 3–18, Mar. 2009.
- [12] Andrew McLeod and Mark Steedman, “Meter detection and alignment of MIDI performance,” in *ISMIR*, 2018, pp. 113–119.
- [13] Darrell Conklin, “Discovery of distinctive patterns in music,” *Intelligent Data Analysis*, vol. 14, no. 5, pp. 547–554, oct 2010.
- [14] Tom Collins, Jeremy Thurlow, Robin Laney, Alistair Willis, and Paul Garthwaite, “A comparative evaluation of algorithms for discovering translational patterns in baroque keyboard works,” in *ISMIR*, 2010.
- [15] David Meredith, “COSIATEC and SIATECCompress: Pattern discovery by geometric compression,” in *Music Information Retrieval Evaluation eXchange (MIREX)*, 2013.
- [16] Ron J. Weiss and Juan Pablo Bello, “Unsupervised discovery of temporal structure in music,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1240–1251, 2011.
- [17] Andrew McLeod and Mark Steedman, “Meter detection in symbolic music using a lexicalized PCFG,” in *Proceedings of the Sound and Music Computing Conference*, 2017, pp. 373–379.
- [18] Daniel Jurafsky and James H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice Hall PTR, 2000.
- [19] Irving J Good, “The population frequencies of species and the estimation of population parameters,” *Biometrika*, pp. 237–264, 1953.
- [20] Andrew McLeod and Mark Steedman, “HMM-based voice separation of MIDI performance,” *Journal of New Music Research*, vol. 45, no. 1, pp. 17–26, Jan. 2016.
- [21] Andrew Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.
- [22] Simon Dixon, “Automatic extraction of tempo and beat from expressive performances,” *Journal of New Music Research*, vol. 30, no. 1, pp. 39–58, Mar. 2001.
- [23] Mitsuyo Hashida, Toshie Matsui, and Haruhiro Katayose, “A new music database describing deviation information of performance expressions,” *ISMIR*, pp. 489–494, 2008.
- [24] Werner Goebel, “Numerisch-klassifikatorische Interpretationsanalyse mit dem ‘Bösendorfer Computerflügel’,” M.S. thesis, Universität Wien, 1999.