Meter Detection in Symbolic Music Using a Lexicalized PCFG

Andrew McLeod
University of Edinburgh
A.McLeod-5@sms.ed.ac.uk

Mark Steedman
University of Edinburgh
steedman@inf.ed.ac.uk

ABSTRACT

This work proposes a lexicalized probabilistic context free grammar designed for meter detection, an integral component of automatic music transcription. The grammar uses rhythmic cues to align a given musical piece with learned metrical stress patterns. Lexicalization breaks the standard PCFG assumption of independence of production, and thus, our grammar can model the more complex rhythmic dependencies which are present in musical compositions. Using a metric we propose for the task, we show that our grammar outperforms baseline methods when run on symbolic music input which has been hand-aligned to a tatum. We also show that the grammar outperforms an existing method when run with automatically-aligned symbolic music data as input. The code for our grammar is available at https://github.com/apmcleod/met-detection.

1. INTRODUCTION

Meter detection is the organisation of the beats of a given musical performance into a sequence of trees at the bar level, in which each node represents a single note value. In common-practice Western music (the subject of our work), the children of each node in the tree divide its duration into some number of equal-length notes (usually two or three) such that every node at a given depth has an equal value. For example, the metrical structure of a single $\frac{3}{4}$ bar, down to the quaver level, is shown in Fig. 1. Additionally, the metrical structure must be properly aligned in phase with the underlying musical performance so that the root of each tree corresponds to a single bar. Each level of a metrical tree corresponds with an isochronous pulse in the underlying music: bar, beat, and sub-beat (from top to bottom). There are theoretically further divisions further down the tree, but as these three levels are enough to unambiguously identify the meter of a piece, we do not consider any lower.

The task is an integral component of Automatic Music Transcription (AMT), particularly when trying to identify the time signature of a given performance, since there is a one-to-one relationship between time signatures and metrical structures. In music, each successive bar may have a different metrical structure than the preceding one; however, such changes in structure are not currently handled by our model, and are left for future work. Our grammar can only be applied to pieces in which each bar is of equal length, has the same number of equal-length beats, and each beat has the same number of equal-length sub-beats. That is, it can be applied to any piece where the metrical tree structure under each node at a given level of the tree is identical. In this work, we evaluate our grammar only on the simple and compound meter types $\frac{2}{X}$, $\frac{3}{X}$, $\frac{4}{X}$, $\frac{6}{X}$, $\frac{9}{X}$, and $\frac{12}{X}$ (where X can be any of 1, 2, 4, 8, or 16), and leave more uncommon and irregular meters for future work. Those interested in asymmetric meter detection should refer to [1].

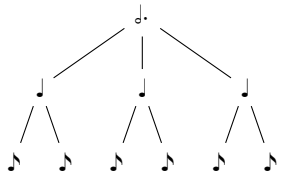

Figure 1. The metrical structure of a $\frac{3}{4}$ bar.

Our grammar is designed to be run on symbolic music data such as MIDI. In this work, we present two experiments: one where the tatum—the fastest subdivision of the beat (we use demisemiquavers, or 32nd notes)—is given, and another where a beat-tracker is used as a preprocessing step to automatically detect some tatum (not necessarily demisemiquavers). Note that ideally, the beat-tracker would be run jointly with our grammar, as beat and meter are intrinsically related; however, we leave such a joint model for future work.

Thus, the task that our grammar solves is one of identifying the correct full metrical structure, composed of: (1) meter type (the number of beats per bar and the number of sub-beats per beat), (2) phase (the number of tatums which fall before the first full bar), and (3) sub-beat length (the number of tatums which lie within a single sub-beat).

2. EXISTING WORK

Most of the early work in the field of meter detection involved rule-based, perceptual models. Longuet-Higgins and Steedman [2] present one which runs on monophonic quantized data and uses only note durations, which was later extended by Steedman [3] to incorporate melodic repetition. Both models were evaluated on full metrical structure detection on the fugues from Bach's Well-Tempered Clavier (WTC). Longuet-Higgins and Lee [4] describe a somewhat similar model, also to be run on monophonic quantized data, though only a few qualitative examples are

presented in evaluation, and the model is unable to handle syncopation. Spiro [5] proposes a rule-based, incremental model for quantized data, combined with a probabilistic n-gram model of bar-length rhythmic patterns, and evaluated on full metrical structure detection on a small corpus of 16 monophonic string compositions by Bach. This remains one of the only successful models for meter detection to use a grammar thus far, though similar grammars have been used for rhythmic and tempo analysis where the meter is given [6–8]. While these rule-based methods show promise, and we base some of our model's principals on them, a more flexible probabilistic model is preferred.

Brown [9] proposed using auto-correlation for meter detection, in which a promising, though limited, evaluation on meter type and sub-beat length detection was shown for 17 quantized pieces, Meudic [10] later proposed a similar model also using auto-correlation on quantized MIDI data for the same task. Eck and Casagrande [11] extended this further, and were the first to use auto-correlation to also calculate the phase of the meter (though phase detection results are limited to synthetic rhythms). They were also the first to do some sort of corpus-based evaluation, though only to classify the meter of a piece as duple or compound. Though auto-correlation has performed well for partial metrical structure detection, there is still a question about whether it can detect the phase of that meter, and no work that we have found has yet done so successfully from real symbolic music data.

Inner Metric Analysis (IMA) was first proposed for music analysis by Volk [12], though only as a method to analyse the rhythmic stress of a piece, not to detect the meter of that piece. It requires quantized MIDI with labeled beats as input, and it involves identifying periodic beats which align with note onsets. Thus, detecting metrical structure and phase using IMA is a matter of classifying the correct beats as downbeats; it is used by De Haas and Volk [13], along with some post-processing, to perform meter detection on quantized MIDI data probabilistically. We were unable to run their model on our data, though they evaluate the model on two datasets, testing both duple or triple classification as well as full metrical structure detection (including phase). However, as the datasets they used are quite homogeneous—95% of the songs in the FMPop corpus are in $\frac{4}{4}$, and 92% of the songs in the RAG corpus [14] are in either $\frac{2}{4}$ or $\frac{4}{4}$ time—we have decided not to include a comparison in this work.

Whiteley et al. [15] perform full metrical structure detection probabilistically from live performance data by jointly modeling tempo, meter, and rhythm; however, the evaluation was very brief, only testing the model on 3 bars of a single Beatles piano performance, and the idea was not used further on symbolic data to our knowledge. Temperley [16] proposes a Bayesian model for the meter detection of unquantized, monophonic MIDI performance data. The general idea is to model the probability of a note onset occurring given the current level of the metrical tree at any time with Bayes' rule. This is combined with a simple Bayesian model of tempo changes, giving a model which can detect the full metrical structure of a performance. Temperley [17] extends this model to work on polyphonic data, combining it into a joint model with a Bayesian voice separator and a Bayesian model of harmony. This joint model performs well on full metrical structure detection on a corpus of piano excerpts, and we compare against it in this work.

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

Figure 2. The grammar rules which form the basis of the PCFG. The subscript b is the number of beats per bar, while s is the number of sub-beats per beat. The terminal symbol r can refer to any rhythmic pattern.

3. PROPOSED METHOD

For our proposed method, we were careful to make as few assumptions as possible so it can be applied to different styles of music directly (assuming enough training data is available). It is based on a standard probabilistic context free grammar (PCFG; presented in Section 3.1) with added lexicalization as as introduced in Section 3.2. The inference procedure is described in Section 3.3.

The basic idea of the grammar is to detect patterns of rhythmic stress in a given piece of music with the grammar, and then to measure how well those stress patterns align with metrical stress patterns. We use note length to measure rhythmic stress in this work, assuming that long notes will be heard as stressed. This assumption is based on ideas from many of the rule-based methods presented above, and works well; however, there are many other factors of musical stress that our grammar does not capture, such as melody and harmony, which have been found to be helpful for meter detection [18], and will be incorporated into our grammar in future work.

3.1 PCFG

The context-free grammar shown in Fig. 2 is used to construct a rhythmic tree quite similar to the metrical tree from Figure 1 above. Each bar of a given piece is first assigned the start symbol S, which can be rewritten as the non-terminal $M_{b,s}$ (representing the meter type), where b is the number of beats per bar and s is the number of sub-beats per beat (2 for simple meters and 3 for compound meters). For example, $M_{4,2}$ represents a meter in $\frac{4}{4}$ time, and $M_{2,3}$ represents a meter in $\frac{6}{8}$ time.

A non-terminal $M_{b,s}$ is rewritten as b beat non-terminals B_s. Each beat non-terminal B_s can be rewritten either as s sub-beat non-terminals SB or as the terminal r, representing the underlying rhythm of the beat. A beat may only be rewritten as r if it contains either (1) no notes or (2) a single note which lasts at least the entire duration of the node (the note may begin before the beat, end after the beat, or both). A sub-beat SB must be rewritten as a terminal r, representing the underlying rhythm of that sub-beat.

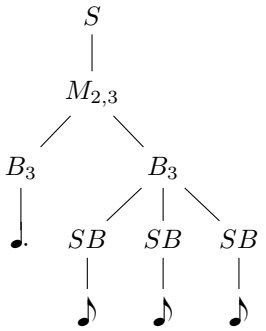

Figure 3. An example of the rhythmic tree of a $\frac{6}{8}$ bar with the rhythm ♩. ♫.

An example of the rhythmic tree of a single $\frac{6}{8}$ bar with the rhythm ♩. ♫ is shown in Figure 3. Here, the first beat is been rewritten as a terminal since it is the only note present.

3.2 Lexicalization

One downside of using a PCFG to model the rhythmic structure is that PCFGs make a strong independence assumption that is not appropriate for music. Specifically, in a given rhythm, a note can only be heard as stressed or important in contrast with the notes around it, though a standard PCFG cannot model this. A PCFG may see a dotted quarter note and assume that it is a long note, even though it has no way of knowing whether the surrounding notes are shorter or longer, and thus, whether the note should indeed be considered stressed.

To solve this problem, we implement a lexicalized PCFG (LPCFG), where each terminal is assigned a head corresponding to its note with the longest duration. Strong heads (in this work, those representing longer notes) propagate upwards through the metrical tree to the non-terminals in a process called lexicalization. This allows the grammar to model rhythmic dependencies rather than assuming independence as in a standard PCFG, and the pattern of strong and weak beats and sub-beats is used to determine the underlying rhythmic stress pattern of a given piece of music.

This head is written $(d; s)$, where d is the duration of the longest note (or, the portion of that note which lies beneath the node), and s is the starting position of that note. The character 't' is added to the end of s if that note is tied into (i.e. if the onset of the note lies under some previous node). In the heads, d and s are normalized so that the duration of the node itself is 1. Thus, only heads which are assigned to nodes at the same depth can be compared directly. A node with no notes is assigned the empty head of $(0; 0)$.

Once node heads have been assigned, each beat and sub-beat non-terminal is assigned a strength of either strong (S), weak (W), or even (E). These are assigned by comparing the heads of siblings in the rhythmic tree. If all siblings' heads are equal, they are assigned even strength. Otherwise, those siblings with the strongest head are assigned strong strength while all others are assigned weak strength, regardless of their relative head strengths.

Head strength is determined by a ranking system, where

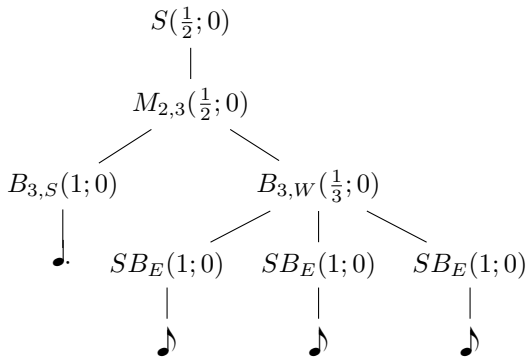

Figure 4. An example of the rhythmic tree of a $\frac{6}{8}$ bar with the rhythm ♩. ♫ including strengths and lexicalization.

heads are first ranked by d such that longer notes are considered stronger. Any ties are broken by s such that an earlier starting position corresponds to greater strength. Any further ties are broken such that notes which are not tied into are considered stronger than those which are.

An example of the rhythmic tree of a single $\frac{6}{8}$ bar with the rhythm ♩. ♫ including strengths and lexicalization is shown in Figure 4.

3.3 Performing Inference

Each of the LPCFG rule probabilities are computed as suggested by Jurafsky and Martin [19], plus additionally conditioning each on the meter type. For example, the replacement $\{M_{2,3}(\frac{1}{2}; 0) \rightarrow B_{3,S}(1; 0)\ B_{3,W}(\frac{1}{3}; 0)\}$ is modeled by the product of Equations (1), (2), and (3). Equation (1) models the probability of a transition given the left-hand-side node's head, while Equations (2) and (3) model the probability of each child's head given its type and the parent's head.

$$p(M_{2,3} \rightarrow B_{3,S}\ B_{3,W} \mid M_{2,3}, (1/2; 0)) \quad (1)$$

$$p((1; 0) \mid M_{2,3}, B_{3,S}, (1/2; 0)) \quad (2)$$

$$p((1/3; 0) \mid M_{2,3}, B_{3,W}, (1/2; 0)) \quad (3)$$

The meter type conditioning ensures that the model not prefer one meter type over another based on uneven training data. Specifically, each initial transition $S \rightarrow M_{b,s}$ is assigned a probability of 1. The actual probability values are computed given a training corpus using maximum likelihood estimation with Good-Turing smoothing as described by Good [20]. If a given replacement's head, as modeled by Equations (2) and (3), is not seen in the training data, we use a backing-off technique as follows. We multiply the probability from the Good-Turing smoothing by a new probability equation, where the meter type is removed (again with Good-Turing smoothing). This allows the grammar to model, for example, the beat-level transitions of a $\frac{9}{8}$ bar using the beat-level transitions of a $\frac{3}{4}$ bar. Note that this does not allow any cross-level calculations where, for example, the beat level of a $\frac{9}{8}$ bar could be modeled by the sub-beat level of a $\frac{6}{8}$ bar, though this could be a possible avenue for future work.

The grammar was designed to be used on monophonic melodies, so we use the voices as annotated in the data. Afterwards, only rhythmic information is needed. That is, the grammar uses onset and offset times for each note, and no pitch or velocity information.

The first step in the inference process is to create multiple hypothesis states, each with probability 1, and each corresponding to a different (meter type, sub-beat length, phase) triplet, which are treated as latent variables. Meter type corresponds to the specific $M_{b,s}$ which will be used throughout the piece for that hypothesis (there is currently a constraint that pieces do not change time signature during a piece). Sub-beat length corresponds to the length of a sub-beat of that hypothesis state. This differentiates $\frac{2}{4}$ time from $\frac{2}{2}$ time, for example. Phase refers to how long of an anacrusis a hypothesis will model. That is, how many tatums lie before the first full bar.

Each state's rhythmic trees are built deterministically, one per voice per bar while that voice is active, throwing out any anacrusis bars. A state's final probability is the product of the probabilities of each of the trees of each of its voices. After running through a full piece, the states are ordered by probability and the metrical structure corresponding to the most likely state's (meter type, sub-beat length, phase) triplet is picked as the model's guess.

One final optimization is made, related to the "rule of congruence" as noted by Longuet-Higgins and Steedman [2], and further described perceptually by Lee [21]. That is, with few exceptions, a composer (at least of classical music), will not syncopate the rhythm before a meter has been established. This means that if the meter has not yet been established, and the underlying rhythm does not match with the metrical structure of a hypothesis state based on its (meter type, sub-beat length, phase) triplet, we should be able to remove it. In practice, we allow up to 5 mismatches before eliminating a metrical hypothesis state. In tests, setting this value to anything from 2 to 20 makes no difference, just the lower the value the faster the program becomes (and the less room for error there is in the case of a particularly adventurous composer). For full details on the implementation of this rule, see Appendix A.

4. EVALUATION

4.1 Metric

To evaluate our method, instead of just checking whether the top hypothesis' metrical structure is fully correct or not, we wanted some measure of partial correctness. For instance, if the correct time signature is $\frac{4}{4}$, a guess of $\frac{2}{4}$ should achieve a higher score than a guess of $\frac{6}{8}$. With that in mind, we propose the following metric.

For each of the three levels of the guessed metrical structure, an exact match with a level of the correct metrical structure is counted as a true positive, while a clash—when all of the nodes in a level of the guessed structure cannot be made by some integer multiple or division of nodes from each of the levels of the correct structure—is counted as a false positive. After all three levels have been tested, each of the correct metrical structure's levels which were not matched count as a false negative. Precision, recall, and F1 can all be computed based on the resulting true positive, false positive, and false negative totals.

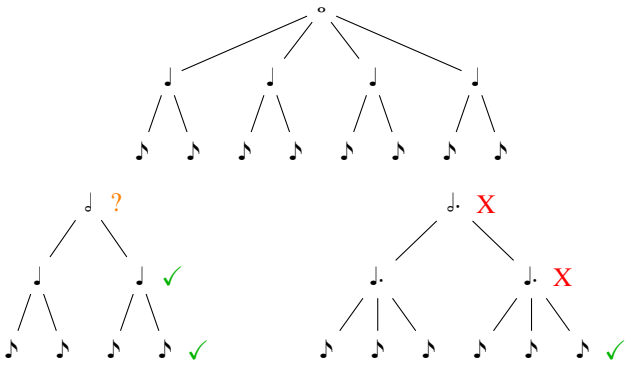

Figure 5. Top: The metrical structure of a $\frac{4}{4}$ bar. If $\frac{4}{4}$ is the correct time signature, a guess of $\frac{2}{4}$ with the correct phase (bottom-left) would give $P = 1.0$, $R = 0.67$, and $F1 = 0.8$. A guess of $\frac{6}{8}$ with the correct phase (bottom-right) would give $P = 0.33$, $R = 0.33$, and $F1 = 0.33$.

Examples of this metric are illustrated in Fig. 5. Given a correct time signature of $\frac{4}{4}$, and assuming that the phase of the guessed metrical structure is correct, if the guessed time signature is $\frac{2}{4}$, there are only 2 true positives; however, the bar-level grouping for $\frac{2}{4}$ does not clash with the metrical structure of $\frac{4}{4}$, so it is not counted as a false positive. There is, however, 1 false negative from the bar level of the $\frac{4}{4}$, giving values of $P = 1.0$, $R = 0.67$, and $F1 = 0.8$. If $\frac{6}{8}$ is guessed instead, the sub-beat level again matches, giving 1 true positive. However, both the beat level and the bar level clash (since 1.5 beats of a $\frac{4}{4}$ bar make a single $\frac{6}{8}$ beat, and $\frac{3}{4}$ of a $\frac{4}{4}$ bar gives a $\frac{6}{8}$ bar), giving 2 false positives and 2 false negatives. This gives values of $P = 0.33$, $R = 0.33$, and $F1 = 0.33$. Much lower, and rightfully so.

For evaluation on a full corpus, true positives, false positives, and false negatives are summed throughout the entire corpus to get a global precision, recall, and F1.

4.2 Data

We report our results on two main corpora: (1) the 15 Bach Inventions, consisting of 1126 monophonic bars (in which a single bar with two voices counts as two bars), and (2) the much larger set of 48 fugues from the Well-Tempered Clavier, containing 8835 monophonic bars. These two corpora contain quantized MIDI files, hand-aligned with a demisemiquaver (32nd note) tatum, and we present results using both this hand-aligned tatum and an automatically-aligned tatum. The notes in each file are split into voices as marked in the corresponding scores. We present additional evaluation in the automatically-aligned case using the German subset of the Essen Folksong Collection [22], 4954 pieces in total, consisting of 66356 monophonic bars.

We use leave-one-out cross-validation within each corpus for learning the probabilities of the grammar. That is, for testing each song in a corpus, we train our grammar on all of the other songs within that corpus. We also tried using

Method	Inventions	Fugues
$\frac{4}{4}$	0.58	0.45
PCFG	0.61	0.63
LPCFG	**0.63**	**0.80**

Table 1. The F1 of each method for each corpus using a hand-aligned tatum.

Method	Inventions	Fugues	Essen [22]
Temperley [17]	**0.58**	0.63	0.60
LPCFG+BT	0.55	**0.72**	**0.74**

Table 2. The F1 of each method for each corpus using an automatically-aligned tatum.

cross-validation across corpora by training on the Inventions when testing the Fugues and vice versa; however, that led to similar but slightly worse results, as the complexities of the rhythms in the corpora are not quite similar enough to allow for such training to be successful. Specifically, there is much more syncopation in the Fugues than in the Inventions, and thus our grammar would tend to prefer to incorrectly choose meters for the Inventions which would result in some syncopation.

4.3 Results

With hand-aligned input, the LPCFG is evaluated against two baselines. First, a naive one, guessing $\frac{4}{4}$ time with an anacrusis such that the first full bar begins at the onset of the first note (the most common time signature in each corpus). Second, the PCFG without lexicalization (as proposed in Section 3.1, with Good-Turing smoothing and rule of congruence matching).

With automatically-aligned input, we evaluate against the model proposed by Temperley [17], which performs beat tracking jointly with meter detection. For direct comparison, we use the fastest beat given by Temperley's model as the tatum, and we call this version of our grammar an LPCFG with beat tracking (LPCFG+BT). It would be better to perform beat tracking and meter detection jointly, as in Temperley's model; however, we leave such joint inference for future work. The automatic alignment presents some difficulty in computing our metric, since a metrical hypothesis generated from an incorrectly aligned input may move in and out of phase throughout the course of a piece due to beat tracking errors. Therefore, we evaluate each meter based on its phase and sub-beat length relative only to the first note of each piece, thus avoiding any misalignments caused by subsequent errors in beat tracking.

The results for hand-aligned input can be found in Table 1, where it can be seen that the LPCFG outperforms all baselines, quite substantially on the fugues. The results for automatically-aligned input are shown in Table 2. Here, the LPCFG+BT outperforms Temperley's model on the fugues and the Essen corpus, but is outperformed by it on the inventions.

For both hand-aligned and automatically-aligned input, it is surprising that the LPCFG (and LPCFG+BT) does not perform better on the inventions, which are simpler compositions than the fugues. The reason for this lack of improvement seems to be a simple lack of training data, as can be seen in Table 3, which shows that as the number of training pieces for each meter type increases, the performance of the LPCFG improves dramatically, though the LPCFG+BT does not follow this trend.

Meter	Inventions			Fugues		
Type	#	LPCFG	+BT	#	LPCFG	+BT
$\frac{6}{X}$	0	—	—	4	0.58	0.83
$\frac{3}{X}$	5	0.60	0.64	7	0.57	0.88
$\frac{2}{X}$	0	—	—	9	0.89	0.76
$\frac{4}{X}$	8	0.71	0.55	26	**0.90**	0.66
All	15	0.63	0.55	48	0.80	0.72

Table 3. The F1 of the LPCFG and LPCFG+BT split by meter type. Here, # represents the number of pieces in each meter type, and meter types which occur only once in a corpus are omitted.

During automatic tatum alignment, beat-tracking tends to quantize rare patterns (which may occur only a single meter type) into more common ones, allowing the grammar to identify the rhythmic stress of a piece without having to parse too many previously unseen rhythms. This helps in the case of not enough training data, but can hurt when more training data is available. These rare patterns tend to be strong markers of a certain meter type, and if enough training data is available to recognize them, such quantization would remove a very salient rhythmic clue. Therefore, for both hand-aligned and automatically-aligned input, more training data in the style of the inventions should continue to improve its performance on that corpus.

Fig. 6 shows the percentage of pieces in each corpus for which each method achieves 3, 2, 1, or 0 true positives, and further details exactly what is happening on the inventions. The true positive counts correspond with those in our metric, and represent the number of levels of the metrical tree (bar, sub-beat, beat) which were matched in both length and phase for each piece. Thus, more true positives corresponds with a more accurate guess.

The improvement on the fugues is clear for both types of input. On the hand-aligned inventions, however, the naive $\frac{4}{4}$ model gets 40% of the metrical structures of the inventions exactly correct (3 TPs), while the LPCFG gets only 26.67%. The LPCFG gets significantly more of its guesses mostly or exactly correct (with 2 or 3 TPs), and eliminates totally incorrect guesses (0 TPs) completely. This shows that, even though it may not have had enough data yet to classify time signatures correctly, it does seem to be learning some sort of structural patterns from what limited data it has. Meanwhile, on the automatically-aligned inventions, the LPCFG+BT gets slightly fewer of its guesses mostly correct than Temperley's model, but significantly fewer (only one invention) totally incorrect, again showing that it has learned some structural patterns. The slightly lower F1 of the LPCFG+BT on the automatically-aligned inventions is due to a higher false positive rate than Temperley's model.

That our model's performance is more sensitive to a lack

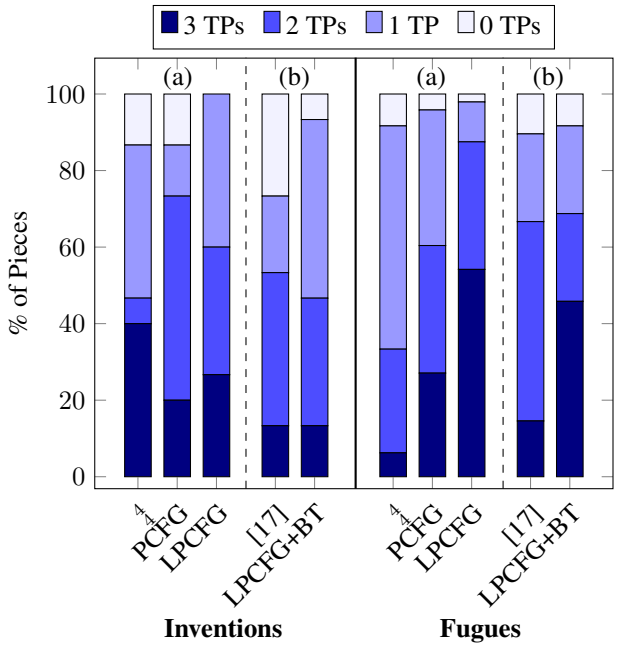

Figure 6. The percentage of pieces in each corpus for which each method's metrical structure guess resulted in the given number of true positives, for (a) hand-aligned and (b) automatically-aligned beats.

Figure 7. The first bar of 15th fugue from WTC book I by Bach (BWV 860).

of training data is further illustrated by the fact that our model outperforms Temperley's substantially on the Essen corpus, where ample training data is available. Indeed, in general, the LPCFG+BT performs worse than Temperley's model when there is a lack of training data, but outperforms it when enough data exists. [1]

A specific case where increased training data would benefit the LPCFG is in the 15th fugue from WTC book I, the first bar of which is shown in Fig. 7. This rhythmic pattern is found throughout the piece, and is a strong indication of a $\frac{6}{8}$ bar, consisting of two even beats, each split into a sub-beat pattern of strong, weak, weak. However, our grammar guesses that this piece is in $\frac{4}{4}$ time simply because it has not seen the transition $\{B_{3,E} \rightarrow SB_S\ SB_W\ SB_W\}$ in a $\frac{6}{X}$ meter type in training. This is indicative of the errors we see throughout the data, showing again that with more training data the results will only improve.

5. CONCLUSION

In this paper, we have proposed an LPCFG for full metrical structure detection of symbolic music data. We have shown that this grammar improves over multiple baseline methods when run on hand-aligned symbolic music input, and that it can be combined with a beat tracking model to achieve good meter detection results on automatically-aligned symbolic music data. The fact that lexicalization adds definite value over a standard PCFG shows that there are complex rhythmic dependencies in music which such lexicalization is able to capture.

Our model is somewhat sensitive to a lack of training data, though it does learn metrical stress patterns quickly, and we will also look at more aggressive cross-level back-off techniques to make the grammar more robust to such a lack of data. For example, it may be possible to model the transitions at the sub-beat level of a $\frac{9}{X}$ meter type using the beat level transitions of a $\frac{3}{X}$ meter type. Furthermore, we will also look to apply our model to more uncommon or irregular meter types such as $\frac{5}{X}$ or $\frac{7}{X}$, perhaps as the concatenation of the more common meter types.

The proposed LPCFG shows promise in meter detection even using only rhythmic data, and future work will incorporate melodic and harmonic information into the grammar. For example, harmonic changes are most likely to occur at the beginnings of bars, and low notes occur more frequently on strong beats, suggesting that incorporating pitch and harmony into the lexical heads may improve performance. Without such additions, our grammar is helpless to hypothesize a meter for an isochronous melody.

Another avenue of future work is to adapt the grammar for use on live performance data by performing inference on the grammar jointly with a beat-tracker. This is more natural than performing beat-tracking as a preprocessing step, as beat and meter are closely related. We will also consider the grammar's application to acoustic data; we have run preliminary experiments using off-the-shelf onset detection models, but found that a more complex approach is needed. Specifically, some sort of voicing information for the onsets would improve performance dramatically, since it would give a more accurate measurement of the lengths of the notes corresponding to each onset.

[1] We do not include evaluation on the Essen corpus with hand-aligned tatum because the pieces are very short and quite simple rhythmically.

6. REFERENCES

[1] T. Fouloulis, A. Pikrakis, and E. Cambouropoulos, "Traditional Asymmetric Rhythms: A Refined Model of Meter Induction Based on Asymmetric Meter Templates," in *Proceedings of the Third International Workshop on Folk Music Analysis*, 2013, pp. 28–32.

[2] H. C. Longuet-Higgins and M. Steedman, "On Interpreting Bach," *Machine Intelligence*, vol. 6, pp. 221–241, 1971.

[3] M. Steedman, "The Perception of Musical Rhythm and Metre," *Perception*, vol. 6, no. 5, pp. 555–569, jan 1977.

[4] H. C. Longuet-Higgins and C. S. Lee, "The perception of musical rhythms," *Perception*, vol. 11, no. 2, pp. 115–128, 1982.

[5] N. Spiro, "Combining Grammar-Based and Memory-Based Models of Perception of Time Signature and Phase," in *Music and Artificial Intelligence*. Springer Berlin Heidelberg, 2002, pp. 183–194.

[6] H. Takeda, T. Nishimoto, and S. Sagayama, "Rhythm and Tempo Recognition of Music Performance from a Probabilistic Approach." in *ISMIR*, 2004.

[7] ——, "Rhythm and Tempo Analysis Toward Automatic Music Transcription," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2007, pp. 1317–1320.

[8] E. Nakamura, K. Yoshii, and S. Sagayama, "Rhythm Transcription of Polyphonic MIDI Performances Based on a Merged-Output HMM for Multiple Voices," in *Proceedings of the 13th Sound and Music Computing Conference*, 2016.

[9] J. C. Brown, "Determination of the meter of musical scores by autocorrelation," *The Journal of the Acoustical Society of America*, vol. 94, no. 4, p. 1953, 1993.

[10] B. Meudic, "Automatic Meter Extraction from MIDI Files," in *Proc. Journées d'informatique musicale*, 2002.

[11] D. Eck and N. Casagrande, "Finding Meter in Music Using An Autocorrelation Phase Matrix and Shannon Entropy," in *ISMIR*, 2005, pp. 504–509.

[12] A. Volk, "The Study of Syncopation Using Inner Metric Analysis: Linking Theoretical and Experimental Analysis of Metre in Music," *Journal of New Music Research*, vol. 37, no. 4, pp. 259–273, dec 2008.

[13] W. B. de Haas and A. Volk, "Meter Detection in Symbolic Music Using Inner Metric Analysis," in *ISMIR*, 2016, pp. 441–447.

[14] A. Volk and W. B. de Haas, "A corpus-based study on ragtime syncopation," *ISMIR*, 2013.

[15] N. Whiteley, A. T. Cemgil, and S. Godsill, "Bayesian Modelling of Temporal Structure in Musical Audio," in *ISMIR*, 2006.

[16] D. Temperley, *Music and Probability*. The MIT Press, 2007.

[17] ——, "A Unified Probabilistic Model for Polyphonic Music Analysis," *Journal of New Music Research*, vol. 38, no. 1, pp. 3–18, mar 2009.

[18] P. Toiviainen and T. Eerola, "Autocorrelation in meter induction: The role of accent structure," *The Journal of the Acoustical Society of America*, vol. 119, no. 2, p. 1164, 2006.

[19] D. Jurafsky and J. H. Martin, "Speech and language processing," *International Edition*, 2000.

[20] I. J. Good, "The population frequencies of species and the estimation of population parameters," *Biometrika*, pp. 237–264, 1953.

[21] C. Lee, "The perception of metrical structure: Experimental evidence and a new model," in *Representing Musical Structure*, P. Howell, R. West, and I. Cross, Eds. Academic Press, May 1991, ch. 3, pp. 59–128.

[22] H. Schaffrath and D. Huron, "The essen folksong collection in the humdrum kern format," *Menlo Park, CA: Center for Computer Assisted Research in the Humanities*, 1995.

A. RULE OF CONGRUENCE

A metrical structure hypothesis begins as unmatched, and is considered to be fully matched if and only if both its beat and sub-beat levels have been matched. Thus, a metrical hypothesis can be in one of four states: fully matched, sub-beat matched, beat matched, or unmatched.

If a hypothesis is unmatched, a note which is shorter than a sub-beat and does not divide a sub-beat evenly is counted as a mismatch. A note which is exactly a sub-beat in length is either counted as a mismatch (if it is not in phase with the sub-beat), or the hypothesis is moved into the sub-beat matched state. A note which is between a sub-beat and a beat in length is counted as a mismatch. A note which is exactly a beat in length is either counted as a mismatch (if it is not in phase with the beat), or the hypothesis is moved into the beat matched state. A note which is longer than a beat, and which both is not some whole multiple of a beat in length and does not divide a bar evenly, is counted as a mismatch.

If a hypothesis is sub-beat matched, it now interprets each incoming note based on that sub-beat length. That is, any note which is longer than a single sub-beat is divided into up to three separate notes (for meter matching purposes only): (1) The part of the note which lies before the first sub-beat boundary which it overlaps (if the note begins exactly on a sub-beat, no division occurs); (2) The part of the note which lies after the final sub-beat boundary which it overlaps (if the note ends exactly on a sub-beat, no division occurs); and (3) the rest of the note. After this processing, a note which is longer than a sub-beat and shorter than a beat is counted as a mismatch. (Due to note division, this only occurs if the note is two sub-beats in length and each beat has three sub-beats.) A note which is exactly a beat in length moves the hypothesis into the fully matched state. A note which is longer than a beat and is not some whole multiple of beats is counted as a mismatch.

If a hypothesis is beat matched, it now interprets each incoming note based on that beat length exactly as is described for sub-beat length in the previous paragraph. After this processing, a note which is shorter than a sub-beat and does not divide a sub-beat evenly is counted as a mismatch. A note which is exactly a sub-beat in length is either counted as a mismatch (if it is not in phase with the sub-beat), or the hypothesis is moved into the fully matched state. A note which is longer than a sub-beat and shorter than a beat, and which does not align with the beginning or end of a beat, is counted as a mismatch.

Once a metrical hypothesis is fully matched, incoming notes are no longer checked for matching, and the hypothesis will never be removed.