METER DETECTION AND ALIGNMENT OF MIDI PERFORMANCE

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

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

ABSTRACT

Metrical alignment is an integral part of any complete automatic music transcription (AMT) system. In this paper, we present an HMM for both detecting the metrical structure of given live performance MIDI data, and aligning that structure with the underlying notes. The model takes as input only a list of the notes present in a performance, and labels bars, beats, and sub beats in time. We also present an incremental algorithm which can perform inference on the model efficiently using a modified Viterbi search. We propose a new metric designed for the task, and using it, we show that our model achieves state-of-the-art performance on a corpus of metronomically aligned MIDI data, as well as a second corpus of live performance MIDI data. The code for the model described in this paper is available at https://www.github.com/apmcleod/met-align.

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 (although the actual durations of a node at a given level will vary with the tempo). 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-value notes such that every node at a given depth has equal value. The metrical structure of a single $\frac{4}{4}$ bar, down to the quaver level, is shown in Figure 1. Each level of a metrical tree corresponds with a pulse level in the underlying music: bar, beat, and sub beat, from top to bottom. The nodes should align in time with corresponding pulses in the performed music. There are theoretically more divisions further down the tree all the way to the tatum level (the fastest pulse present in a piece of music, often the 32nd note), but as these three levels are enough to unambiguously identify the time signature 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. The time signature may change between bars (though this is not particularly common). However, such changes in structure are not currently handled by our model, and are left for future work. The proposed model 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 model 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 may take any value), and leave more uncommon and irregular meters for future work. Those interested in asymmetric meter detection should refer to [9].

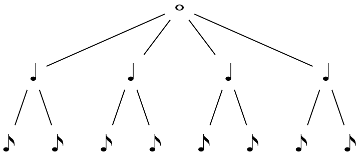

Figure 1. The metrical structure of a $\frac{4}{4}$ bar.

Existing work on full metrical alignment of live performance MIDI data is sparse. There is a good amount of existing work on meter detection (but not alignment) from metronomic data (e.g., [2, 14]), including some which labels the meter type (i.e., duple or compound) of a given piece of music, but does not align a full metrical structure with the notes of the piece (except for synthetic rhythms, as in [8]). There is existing work which performs full metrical alignment of MIDI data, but not from live performance [4]. In the acoustic domain, beat tracking and downbeat detection are relatively common areas of research, but stop short of a full meter detection and alignment (e.g. [1, 7]).

The related problems of rhythm quantisation and note value detection have also seen some attention, but neither are directly relevant to our task. For example, [17] quantises performed rhythms to a grid, but the set of possible onset locations for notes is known a priori (and changes based on the time signature of the underlying piece). [3] tracks beats and tempo, but does not go so far as to align a full metrical grid with bars and sub beats. [15] assigns a note value to each note, but does not explicitly align the notes with any underlying beat or meter.

[22] performs full metrical structure detection and alignment probabilistically from live performance data by jointly modelling 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 MIDI data to our knowledge. [19] proposes a Bayesian model for the meter detection and alignment of monophonic MIDI performance data which mod-

Attribution: Andrew McLeod, Mark Steedman. "Meter Detection and Alignment of MIDI Performance", 19th International Society for Music Information Retrieval Conference, Paris, France, 2018.

els 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. [20] 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 and alignment on a corpus of piano excerpts, and we compare against it in this work.

2. PROPOSED MODEL

Our proposed model tracks pulses at the tatum level of a musical performance based on two musicological principles: (1) the rate of these tatums should be relatively constant without large discontinuities; and (2) notes should lie relatively close to these tatums. The model is an HMM where the observed data is the notes of a given piece, grouped into sets.

2.1 State Space

Each state S in our model represents a single bar, containing (1) a list of the tatums from that bar and (2) a metrical hierarchy, describing which of those tatums are beats and sub beats. The list of tatums is represented by $S.t$, where $S.t_i$ is the ith tatum in the bar, and $S.t_{|S.t|}$ is the downbeat of the following bar. The tatums are in increasing time order, where $\mathcal{T}(S.t_i)$ represents the time of tatum $S.t_i$. A state's metrical hierarchy has some number of tatums per sub beat, sub beats per beat, and beats per bar, as well as an anacrusis length, measured by the number of tatums which fall before the first downbeat of a given piece. In our model, we restrict the number of tatums per sub beat to be 4, although in theory, any number could be used. We also restrict the anacrusis length to be some integer multiple of the number of tatums per sub beat, a simplifying assumption that ensures the first note of each piece will fall on a sub beat. The set of possible sub beat per beat and beat per bar pairs (i.e., time signatures) are taken all of those found in our training set ($\frac{2}{X}$, $\frac{3}{X}$, $\frac{4}{X}$, $\frac{6}{X}$, $\frac{9}{X}$, and $\frac{12}{X}$). A state's tempo, $\mathrm{T}(S)$, is defined as the average length of its beats.

Each possible initial state S_0 contains no tatums, and every possible metrical hierarchy is considered equally probable. To reduce our model's search space, we place a restriction on the range of allowed values for $T(S_1)$: $t_{min} \leq T(S_1) \leq t_{max}$. Nonetheless, because the possible tatum times for each state are unbounded, our model contains an infinite number of possible states. Thus, instead of predefined emission and transition probabilities, we define emission and transition functions, presented in the following sections.

2.2 Emission Function

After the initial state (which emits nothing), each state S_i emits a set of notes N_i, containing only notes n whose onset times lie between that state's first (inclusive) and last (exclusive) tatum. This set is allowed to be empty. Each emitted note has an onset time $\mathrm{On}(n)$, an offset time $\mathrm{Off}(n)$, and a pitch $\mathrm{Pitch}(n)$ (though it is unused).

The probability of a state S_i to emit the note set N_i is presented as $P(N_i|S_i)$ in Eqn (1). The first term, calculated entirely by the lexicalised probabilistic context-free grammar (LPCFG) presented in [13], is used to prefer generating rhythms which have a high probability according to the grammar. The LPCFG is a replacement grammar which first parses a given rhythm into a metrical tree structure. It then assigns strengths to nodes in the tree based on note duration in a process called lexicalisation. The probability of a tree is calculated by taking the product of the learned probabilities of each grammar transition, based on counting occurrences of a given transition from a training corpus of parsed rhythms. Each note is aligned to the nearest tatum by the LPCFG in order to calculate $P(rhythm)$, but this alignment is neither saved nor emitted. The LPCFG is designed to work directly on monophonic melodies only. Therefore, for polyphonic input, this $P(rhythm)$ term is in fact a product of one probability per voice, each of which is calculated by the LPCFG. For voice assignments, we use [12] as a preprocessing step.

$$P(N_i|S_i) = P(rhythm) \prod_{n \in N} P(n|S_i.t) \qquad (1)$$

The second term in Eqn (1) is used to prefer states whose tatums align closely with the emitted notes, and is calculated as in Eqn (2), where $\mathcal{N}_1(\mu, \sigma, x)$ conceptually represents a normal distribution with mean μ and standard deviation σ evaluated at x.[1] Thus, $P(n|S_i.t)$ is used to assign higher probabilities to those states which emit notes which are closely-aligned with their tatums. In this equation, $closest(S_i.t)$ represents the tatum from S_i whose time is closest to the onset time of the note n.

$$P(n|S_i.t) = \mathcal{N}_1\big(0, \sigma_n, \mathrm{On}(n) - \mathcal{T}(closest(S_i.t))\big) \qquad (2)$$

2.3 Transition Function

A state S_{i-1} may transition to a state S_i if and only if: (1) the two states' metrical hierarchies are identical (our model cannot handle pieces with time signature changes) and (2) the time of the last tatum in S_{i-1} is equal to the time of the first tatum in S_i. Note that the second condition is invalid in the case of a transition from S_0 to S_1 since S_0 contains no tatums; in this case, we instead restrict $S_1.t_1$ to lie exactly on the first observed note's onset time.

The transition probability $P(S_i|S_{i-1})$ is shown in Eqn (3), where the first term, defined in Eqn (4), models the probability of a tempo change and the second term, defined in Eqn (5), models the spacing of the tatums themselves.

$$P(S_i|S_{i-1}) = P(T(S_i)|T(S_{i-1}))P(S.t) \qquad (3)$$

[1] Normal distributions are used in multiple places throughout this model with potentially widely varying standard deviations, resulting in potentially wildly different results when evaluated at an identical number of standard deviations from the mean for different normal distributions. Since the distributions are used in contexts in which they cannot be properly normalised (due to their continuous domain), the precise probability value for $\mathcal{N}_1(\mu, \sigma, x)$ is calculated using a standard normal distribution with mean 0 and standard deviation 1 evaluated at $\frac{x-\mu}{\sigma}$.

$$P(T(S_i)|T(S_{i-1})) = \begin{cases} \mathcal{N}_1(\mu_{t_0}, \sigma_{t_0}, T(S_i)) & i = 1 \\ \mathcal{N}_1\big(0, \sigma_t, \frac{T(S_i)-T(S_{i-1})}{T(S_{i-1})}\big) & i \geq 2 \end{cases} \tag{4}$$

$$P(S.t) = E(b \in S.t) \prod_{b \in S.t} \big(E(sb \in b) \prod_{sb \in b} E(t \in sb)\big) \tag{5}$$

In Eqn (4), the tempo of the first bar (where $i = 1$) is assumed to be normally distributed around μ_{t_0} with standard deviation σ_{t_0}, while subsequent tempo changes are evaluated as the proportional change from the tempo of the previous bar, again normally distributed, this time with mean 0 and standard deviation σ_t. Percent change is used rather than absolute change because human perception of tempo changes have been shown to follow Weber's Law [21].

For the tatum timings in Eqn (5), the function $E(t)$, defined in Eqn (6), evaluates the probability of the evenness of any given list of times. $E(b \in S.t)$ calculates this for all of the beats b in the state, while the terms $E(sb \in b)$ and $E(t \in sb)$ perform the same calculation for the sub beats in each beat and the tatums in each sub beat respectively.

$$E(t) = \begin{cases} \mathcal{N}_1(\mu_e, \sigma_e, \frac{\sigma(t)}{\mu(t)})/E_{norm} & \frac{\sigma(t)}{\mu(t)} \geq \mu_e \\ \mathcal{N}_1(\mu_e, \sigma_e, \mu_e)/E_{norm} & \frac{\sigma(t)}{\mu(t)} < \mu_e \end{cases} \tag{6}$$

$$E_{norm} = \frac{1}{2} + \frac{\mu_e}{\sigma_e}\mathcal{N}_1(\mu_e, \sigma_e, \mu_e) \tag{7}$$

$E(t)$ is a piecewise function which takes as input a list of the lengths of a group of tatums, sub beats, or beats (rather than their times). Here, $\mu(t)$ represents the mean of those lengths and $\sigma(t)$ represents the standard deviation of those lengths. The function is calculated as a modified normal distribution with mean μ_e and standard deviation σ_e, based on the input list's standard deviation as a proportion of its mean. If this proportion is greater than or equal to μ_e, the result is calculated from a straightforward normal distribution. Otherwise, the result is exactly the value of a standard normal distribution evaluated at its mean.

This value is then normalised so as to ensure the new distribution's integral to again sum to 1 by dividing by the factor E_{norm}, defined in Eqn (7) as the sum of two terms. $\frac{1}{2}$ is the area of the standard normal distribution greater than the mean, and $\frac{\mu_e}{\sigma_e}\mathcal{N}_1(\mu_e, \sigma_e, \mu_e)$ is the area of the rectangle formed by extending the peak of the standard normal distribution to the left until the value corresponding to 0 from the non-standardised normal distribution, as values less than this correspond to a negative $\sigma(t)$, which is not possible.

2.4 Search Space Reduction

We use a modified Viterbi search to perform inference on our model, using a beam search where at each step we save only the B most probable hypothesis states (not including those still at S_0 with no tatums yet).

For the transition from S_0 to S_1, we introduce two heuristics: (1) the first tatum in S_1 must lie exactly on the onset of the first observed note and (2) the last tatum in S_1 must also lie exactly on a note onset, though which note specifically is not restricted by any means other than limiting the tempo of the first bar using t_{min} and t_{max}. According to these heuristics, for each S_0, the supervisor creates the observed note set for every possible S_1. Allowed times for the tatums in $S_1.t$ are also restricted based on each observed note set N_1. Essentially, all tatums are placed evenly unless there is a specific reason not to (i.e., unless a note onset lies close to a tatum).

Specifically, a given value for $S_1.t$ is legal if it can ever be generated by the following procedure. First, the appropriate number of beats (according to a given state's metrical hierarchy) are placed between the first and last tatum times, as if each tatum was evenly spaced. Next, each placed beat—excluding the last beat as well as the first—may be shifted to the location of any note whose onset time is within half of one sub beat length of the original beat location. Each beat (again excluding the first and last as appropriate) may then be nudged up to half of a tatum length around its location with a magnetism of M_b, as shown in Eqn (8). Here, t is the original time of the beat, M is the magnetism (M_b in this case) which is used to control how far the beat is nudged, and N is the set of notes which lie within the given window. This equation can always return the original time, though it is also allowed to nudge the given time towards either the onset time of the closest note ($closest(N)$) or the average onset time of all notes within the window ($avg(N)$), if N is large enough. Sub beats are placed similarly: initially evenly between any of the existing beats, then nudged up to one tatum length around its location with magnetism M_{sb}. Notice that the sub beats are not shifted. Finally, tatums are placed evenly between the sub beats (and neither shifted nor nudged).

$$nudge(t, M, N) = \begin{cases} t & always \\ t + M(closest(N) - t) & |N| > 0 \\ t + M(avg(N) - t) & |N| > 1 \end{cases} \tag{8}$$

Allowed times for the tatums in $S_i.t$ for $i > 1$ are restricted to those which can be generated by the same procedure, with the exception that the final beat in $S_i.t$ may now be shifted and nudged. Initial beat locations are calculated such that $T(S_{i-1}) = T(S_i)$.

Intuitively, this process of shifting and nudging allows a hypothesis' tempo to smoothly increase or decrease based on the observed notes. Beats are allowed to change the tempo more drastically than sub beats because they are more salient, and more likely to align with note onsets.

Even with the above restrictions, the search space is still large. As mentioned we use a beam search, where at each step we save only the top B most probable hypothesis states (not including those still at S_0 with no tatums yet). Before we remove those hypotheses which fall outside the beam, we remove hypotheses which are deemed to be too close to another more probable hypothesis based on a threshold Δ_{min}. Specifically, a hypothesis which has identical metrical hierarchy to a more probable hypothesis, and whose tempo and most recent tatum time both lie within Δ_{min} of that other hypothesis' tempo and most recent tatum time is removed.

2.5 Supervisor

It is important to note that due to the way in which the observed note sets are grouped by bar, the individual note sets for different paths through the HMM state space for a given piece will not be identical, although the union of all note sets on any given path equals exactly the set of notes present in the piece. To handle this complication, we introduce a *supervisor* during the HMM decoding process which takes each note individually in onset order, grouping them into note sets and passing the sets to the appropriate hypothesis state at each step. Specifically, for a given hypothesis state, the supervisor determines the longest and shortest possible lengths for the following bar (based on allowed shifts and nudges as described in the previous section). Then, it creates every possible note set given those bounds, and allows the hypothesis state to transition and branch on each of those note sets.

2.6 Optimisations

Here we describe two changes used to make our model more robust in regards to the idiosyncrasies of live performance such as staccato and ornamentation.

For handling staccato notes which are much shorter than their note values would suggest in the score, we extend each note's offset until either the onset of the following note in the bar within its voice (if one exists), or to the end of its bar. This allows the LPCFG, which is trained on metronomic MIDI where staccato is not present, to better recognise the rhythms present in live performance.

For handling ornamentation such as trills, we use a threshold $trill_{max}$. Any note whose onset time is within $trill_{max}$ of the onset time of the previous note within its voice is removed (though the removed notes are still used when deciding whether to remove the subsequent note). The overall effect of this process is that trills or any very fast ornamentation (which again would not be present in the LPCFG's training data) are reduced to a single short note with its onset at the start of the trill or ornamentation. If this optimisation is used in conjunction with the extend notes optimisation, the remaining notes are extended only after trills and ornamentation are removed, and the result is that a fast ornamentation is replaced by a single long note.

3. EVALUATION

3.1 Corpora

For evaluation, we use two corpora: one containing metronomic MIDI files of the 48 fugues from Bach's Well-Tempered Clavier (WTC)[2] and Bach's 15 Inventions,[3] and another of 13 live performance MIDI files of Bach's fugues and preludes from the WTC, from CrestMusePEDB[4] [10]. For training, we also use the miscellaneous corpus, released and used by [20] for training, divided into a live performance portion (containing 22 pieces by various composers recorded from a MIDI keyboard) and a metronomic portion (containing 45 non-performed pieces by various composers). For voice assignments in all corpora, we run [12] as a preprocessing step.

3.2 Training

To train most of the parameters for the beat tracking model, we measure statistics from the live performance portion of the miscellaneous corpus. This results in values of $\mu_{t_0} = 1.0885\ s$, $\sigma_{t_0} = 709.918\ ms$, $\sigma_t = 0.0743$, $\mu_e = 0.0181$, $\sigma_e = 0.0336$, $\sigma_n = 6.655\ ms$, $t_{min} = 0.4\ s$, and $t_{max} = 3\ s$.

The remaining parameters are set in an ad hoc fashion through testing on the miscellaneous corpus, and we have found our model's performance not to be very sensitive to the precise values used. Specifically, we use $M_b = 1.0$, $M_{sb} = 0.5$, and $trill_{max} = 0.1\ s$. Δ_{min} and B are simply optimisations used to improve the speed of our model, and we use values of $1\ ms$ and 200 respectively, though in practice, lower values of Δ_{min} or higher values of B only improve our model's performance.

For our standard evaluation, we train the LPCFG's probabilities from the metronomic portion of the miscellaneous corpus, since this allows for a direct comparison with the model of [20]. However, it is noted in [13] that the grammar is sensitive to a lack of training data, particularly a lack of training data in the style of the evaluation corpus, which happens when training on the miscellaneous corpus for evaluation on Bach compositions. To investigate this further, we also run experiments when training the LPCFG's probabilities on a superset containing the metronomic portion of the miscellaneous corpus as well as the entire metronomic corpus of Bach compositions. Note that when evaluating this version of our model, we leave out the piece currently being evaluated from the grammar's training set so as to avoid overfitting. In all experiments, we train the LPCFG with data that has undergone the same optimisations as the data to be evaluated (in terms of extending notes and removing trills and ornamentation).

3.3 Metric

Quantitative evaluation of previous work on meter alignment, particularly with MIDI data, is uncommon, and a few possible metrics are discussed in [18]. [20] reports five values which take into account tempo, phase, and the branching factor at each level of the metrical tree. Work on acoustic meter detection (e.g. [11]) often reports F-measures of beats and downbeats, treated as points in time.

To evaluate our model's performance, we would rather use a metric similar to that from [13] which is a single value, takes into account the tree structure's groupings (rather than just its beat locations), and has some idea of the partial correctness of a metrical alignment. However, as it is designed for use mainly on beat-aligned data where a metrical hypothesis cannot move in and out of phase throughout a piece, a few adjustments must be made to adapt it for use on live performance data. We call our newly designed evaluation metric the metrical F-measure.

[2] The fugues were acquired from `www.musedata.org`.

[3] The inventions were acquired from `www.imslp.org`.

[4] We do not include the 13th prelude from WTC Book I due to an error in the file.

Method	Metronomic	Live Performance
Temperley [20]	67.65	47.62
This Work	78.71	39.63
+T	75.36	39.40
+X	79.89	45.27
+X +T	77.67	47.81
+Bach	80.48	38.21
+Bach +T	80.08	42.35
+Bach +X	**80.50**	55.43
+Bach +X +T	80.48	**56.51**

Table 1. The average metrical F-measure of our method compared against those of [20] on our two corpora. +T indicates use of the remove trills and ornamentation optimisation, +X indicates use of the extend notes optimisation, and +Bach indicates using the additional Bach training data for the LPCFG.

It takes into account every grouping at three levels of the metrical hierarchy throughout an entire piece: the sub beat level, the beat level, and the bar level.

For each hypothesised grouping at these metrical levels, we check if it matches a ground truth grouping. A hypothesised grouping is said to match a ground truth grouping if its beginning and ending times are each within 70 ms of the beginning and ending times of that particular ground truth grouping, regardless of the metrical level of either grouping. [5] Each matched pair of groupings within a piece count as a true positive, while any unmatched hypothesis groupings count as false positives, and any unmatched ground truth groupings count as false negatives. The metrical F-measure of a piece is then calculated as the harmonic mean of precision and recall as usual, and our reported results average these metrical F-measures across all songs in each corpus.

3.4 Results

We compare our model against that of Temperley [20], which is trained entirely on the miscellaneous corpus. For direct comparison, the standard version of our model is trained on the same corpus, but we present an evaluation of a few different versions of it based on different optimisations or training data. Results can be found in Table 1, where +T indicates use of the remove trills and ornamentation optimisation, +X indicates use of the extend notes optimisation, and +Bach indicates that the LPCFG training was augmented with the additional Bach compositions. We do not also augment Temperley's model with additional training data because there is no straightforward way to do so, and the model does not seem to be one which would be as sensitive to a lack of training data as our model.

The results show that on metronomic data, our model without optimisations clearly outperforms Temperley's when using identical training data. The optimisations offer no significant improvement (which is unsurprising as they were designed specifically to help with live performance), but augmented training data leads to a small but consistent increase in performance across all optimisation configurations. On live performance, our model without optimisations underperforms Temperley's, both with and without augmented training data. However, the optimisations lead to increased performance: our model using both optimisations matches Temperley's performance with identical training data, and exceeds it by almost 9 points with augmented training data. The effect of each optimisation is discussed in detail in Section 3.4.1.

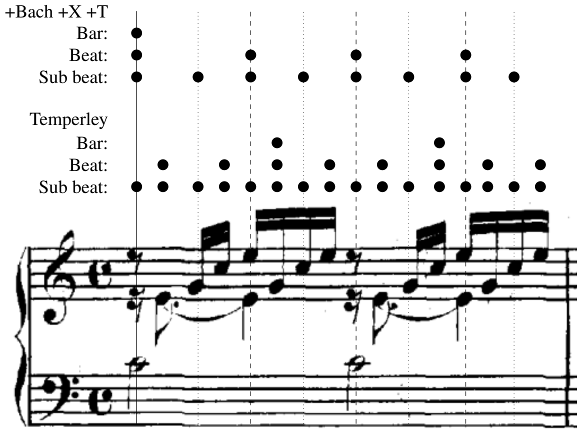

Figure 2. The first bar of the 1st prelude from WTC Book I (BWV 846). Above the music, the results from Temperley's model (bottom) are shown as well as the results from our +Bach +X +T model (top).

The distribution of metrical F-measures for Temperley's model, run on live performance data, appears to be binomial: of the 13 pieces, three score below 20, while six score above 55, indicating that while the model performs well in general, it sometimes guesses a meter which is nearly entirely incorrect. With both optimisations, on the other hand, our model's scores are normally distributed around 65, with 8 pieces scoring between 55 and 75. Additionally, no pieces score below 20, indicating that it is more likely to make some partially correct guess, even if it is not entirely correct. The 1st prelude from WTC Book I illustrates this difference in performance, and its first bar is shown in Figure 2 along with the results of Temperley's model and our +Bach +X +T model. The piece is in $\frac{4}{4}$ time, and Temperley's model achieves a score of only 15.74, guessing a $\frac{3}{8}$ time whose beats are even out of phase with the ground truth sub beats throughout much of the piece. On the other hand, our model scores 93.27, guessing a $\frac{4}{4}$ time which begins perfectly aligned, although it does shift slightly out of phase later in the piece.

One example of a piece for which Temperley's model outperforms ours is the 2nd prelude of WTC Book II, the first bar of which is shown in Figure 3 along with the results of Temperley's model and our +Bach +X +T model. For this piece, Temperley's model achieves a score of 78.99 while ours only manages a score of 61.83. This piece is in $\frac{4}{4}$ time and contains relatively non-syncopated

[5] This 70 ms window is taken directly from a popular beat tracking metric [6].

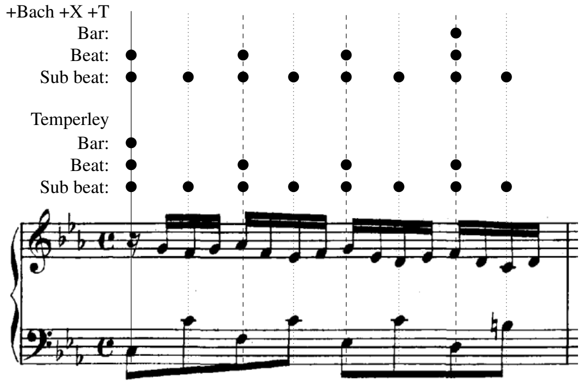

Figure 3. The first bar of the 2nd prelude from WTC Book II (BWV 871), showing an example the nearly isochronous bars which give our model problems. Above the music, the results from Temperley's model (bottom) are shown as well as the results from our +Bach +X +T model (top).

rhythms, with many bars containing either only sixteenth notes or only eighth notes in a given voice, as can be seen in the figure. While Temperley's model captures this meter correctly (with some phase errors), our model guesses a $\frac{4}{4}$ time which is early by a single beat. Our model has some difficulty finding the correct phase of isochronous melodies since it uses no pitch or harmonic information (which are the most salient indicators of metrical phase in such isochronous pieces). Temperley's model, on the other hand, also includes chord detection, allowing it to better handle such melodies.

3.4.1 Optimisations

Another aspect of our model to investigate is the effect of the different optimisations on its performance. As can be seen from Table 1, they (+X and +T) have little effect on metronomic data (which is not surprising given that they are designed specifically for live performance). However, on live performance data, they improve performance significantly. Both with and without augmented training data, the remove trills optimisation has a small effect by itself (essentially none without the data and very small with it), but extending notes leads to a significant improvement. The combination of both optimisations improves performance even further, leading to peak performance both with and without augmented training data.

One specific example where the remove trills optimisation leads to improvement with augmented training data is on the 7th fugue from WTC Book I, where our +Bach and +Bach +T models achieve scores of 31.58 and 60.20 respectively. There is a repeated trill throughout this piece, leading the +Bach model to lengthen its beat length such that the trill is interpreted as 16th notes. With the remove trills optimisation, however, our model is able to find the correct metrical structure. Essentially, the remove trills optimisation frees our model from the constraint of trying to align its tatums with each note in a trill or ornamentation.

An example of a piece for which the extend notes optimisation makes an improvement is the 17th prelude from WTC Book I. In this piece, in $\frac{3}{4}$ time, the lowest voice has a very common repeated rhythm of an eighth note followed by two sixteenth notes followed by four more eighth notes, where the eighth notes are all played staccato. With the optimisation, our model correctly recognises the beat and sub beat levels, although it incorrectly guesses $\frac{2}{4}$ time rather than the correct $\frac{3}{4}$ time, scoring 53.59. Without the optimisation, on the other hand, these eighth notes are not as salient, and the model instead guesses a $\frac{2}{2}$ meter which moves in and out of phase throughout the piece, achieving a score of only 16.47. Throughout the corpus, the extend notes optimisation helps find strong notes whenever they are played staccato.

The combination of both optimisations improves overall performance even further, enabling the model to handle both staccato passages and ornamentation. The improvements from both optimisations are seen in the fully optimised model, alongside other slight improvements throughout the corpus such as fixing the placement of a single misaligned beat here or there. For example, in the previously discussed 17th prelude from WTC Book I, the fully optimised model achieves a metrical F-measure of 60.35 while no other model eclipses a score of 54, even though the basic metrical alignment (a $\frac{2}{4}$ meter) does not change between the it and the +Bach +X model.

4. CONCLUSION

In this paper, we have described a model for metrical structure detection and alignment from live performance MIDI.

Our model is in the form of an HMM which performs metrical structure detection and alignment given only a list of note pitches and onset and offset times, and we have shown that the model achieves state-of-the-art performance when evaluated on a corpus of metronomic data, as well as a second corpus of live performance data. The HMM incorporates a rhythmic grammar as one component, working with the grammar to align an input piece with a metrical structure. This joint model is probabilistic and incremental, and requires no information a priori except for note onset and offset times. We have also proposed a new metric for the task, which takes into account vertical misalignments (for example, those which align the beat level of a piece with bars) and partial correctness.

In future work, we would like to extend the evaluation of our model with more data. In particular, our corpus of 13 pieces of live performance MIDI would benefit from an expansion, and allow us to perform a more in-depth analysis of the results.

Metrical structure detection and alignment is clearly an important task for any complete transcription system, and we have shown that our joint model is able to perform the task well, even using only rhythmic data. Incorporating additional information such as pitch or harmony should only lead to better performance. Specifically, it has been shown that harmonic changes are most likely to occur at the beginnings of bars [16], and low notes may be a salient feature of strong beats in addition to note duration [5].

5. ACKNOWLEDGEMENTS

This work was partially funded by EU ERC H2020 Advanced Fellowship GA 742137 SEMANTAX and a Google faculty award.

6. REFERENCES

[1] Sebastian Böck, Florian Krebs, and Gerhard Widmer. Joint beat and downbeat tracking with recurrent neural networks. In *ISMIR*, pages 255–261, 2016.

[2] Judith C. Brown. Determination of the meter of musical scores by autocorrelation. *The Journal of the Acoustical Society of America*, 94(4):1953, 1993.

[3] A. T. Cemgil and B. Kappen. Monte carlo methods for tempo tracking and rhythm quantization. *Journal of Artificial Intelligence Research*, 18:45–81, January 2003.

[4] W. Bas De Haas and Anja Volk. Meter detection in symbolic music using inner metric analysis. In *ISMIR*, pages 441–447, 2016.

[5] Simon Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30(1):39–58, March 2001.

[6] Simon Dixon. Evaluation of the audio beat tracking system beatroot. *Journal of New Music Research*, 36(1):39–50, March 2007.

[7] 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*, 25(1), 2017.

[8] Douglas Eck and Norman Casagrande. Finding meter in music using an autocorrelation phase matrix and shannon entropy. In *ISMIR*, pages 504–509, 2005.

[9] Thanos Fouloulis, Aggelos Pikrakis, and Emilios 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*, pages 28–32, 2013.

[10] Mitsuyo Hashida, Toshie Matsui, and Haruhiro Katayose. A new music database describing deviation information of performance expressions. *ISMIR*, pages 489–494, 2008.

[11] Florian Krebs, Andre Holzapfel, A. Taylan Cemgil, and Gerhard Widmer. Inferring metrical structure in music using particle filters. 23(5):817–827, 2015.

[12] Andrew McLeod and Mark Steedman. HMM-based voice separation of MIDI performance. *Journal of New Music Research*, 45(1):17–26, January 2016.

[13] Andrew McLeod and Mark Steedman. Meter detection in symbolic music using a lexicalized PCFG. In *Proceedings of the 14th Sound and Music Computing Conference*, pages 373–379, 2017.

[14] Benoit Meudic. Automatic meter extraction from MIDI files. In *Journées d'informatique musicale*, 2002.

[15] Eita Nakamura, Kazuyoshi Yoshii, and Shigeki Sagayama. Rhythm Transcription of Polyphonic Piano Music Based on Merged-Output HMM for Multiple Voices. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(4):794–806, April 2017.

[16] Hélène Papadopoulos and Geoffroy Peeters. Joint estimation of chords and downbeats from an audio signal. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1):138–152, January 2011.

[17] Christopher Raphael. Automated rhythm transcription. In *ISMIR*, 2001.

[18] David Temperley. An evaluation system for metrical models. *Computer Music Journal*, 28(3):28–44, September 2004.

[19] David Temperley. *Music and Probability*. The MIT Press, 2007.

[20] David Temperley. A unified probabilistic model for polyphonic music analysis. *Journal of New Music Research*, 38(1):3–18, March 2009.

[21] Kim Thomas. Just noticeable difference and tempo change. *Journal of Scientific Psychology*, pages 14–20, 2007.

[22] Nick Whiteley, A. Taylan Cemgil, and Simon Godsill. Bayesian modelling of temporal structure in musical audio. In *ISMIR*, 2006.