Information Sciences 572 (2021) 482–500

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Musical rhythm transcription based on Bayesian piece-specific score models capturing repetitions [☆]

Eita Nakamura [a,b,*], Kazuyoshi Yoshii [a,c]

[a] Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
[b] The Hakubi Center for Advanced Research, Kyoto University, Kyoto 606-8501, Japan
[c] AIP Center for Advanced Intelligence Project, RIKEN, Tokyo 103-0027, Japan

ARTICLE INFO

Article history:
Received 4 August 2020
Received in revised form 16 February 2021
Accepted 28 April 2021
Available online 4 May 2021

Keywords:
Music transcription
Musical language model
Bayesian modelling
Symbolic music processing
Musical rhythm

ABSTRACT

Most work on musical score models (a.k.a. musical language models) for music transcription has focused on describing the local sequential dependence of notes in musical scores and failed to capture their global repetitive structure, which can be a useful guide for transcribing music. Focusing on rhythm, we formulate several classes of Bayesian Markov models of musical scores that describe repetitions indirectly using the sparse transition probabilities of notes or note patterns. This enables us to construct piece-specific models for unseen scores with an unfixed repetitive structure and to derive tractable inference algorithms. Moreover, to describe approximate repetitions, we explicitly incorporate a process for modifying the repeated notes/note patterns. We apply these models as prior musical score models for rhythm transcription, where piece-specific score models are inferred from performed MIDI data by Bayesian learning, in contrast to the conventional supervised construction of score models. Evaluations using the vocal melodies of popular music showed that the Bayesian models improved the transcription accuracy for most of the tested model types, indicating the universal efficacy of the proposed approach. Moreover, we found an effective data representation for modelling rhythms that maximizes the transcription accuracy and computational efficiency.

1. Introduction

Music transcription is an actively studied but yet unsolved problem in music information processing [1,2]. One of the goals of music transcription is to convert a musical performance signal into a human-readable symbolic musical score. While recent studies have achieved highly accurate pitch detection methods [3,11,33], it is also necessary to transcribe rhythms in order to obtain symbolic music representations [6,7,10,15,17–20,23,31,32]. Since there are many logically possible representations of rhythms (including inappropriate ones for transcription) for a given performance [6], using a (musical) score model (a.k.a. musical language model) that describes the prior knowledge of musical scores is a key to solving this problem. Here, we study the problem of the rhythm transcription of monophonic music, which is the task of recognizing score-notated

[☆] This work is in part supported by JSPS KAKENHI Nos. 15K16054, 16H01744, 16J05486, and 19K20340; JST ACCEL No. JPMJAC1602; the Kyoto University Foundation; and the Kayamori Foundation. The work of EN was supported by the JSPS research fellowship (PD).

* Corresponding author at: Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan.
E-mail address: eita.nakamura@i.kyoto-u.ac.jp (E. Nakamura).

https://doi.org/10.1016/j.ins.2021.04.100
0020-0255/© 2021 Elsevier Inc. All rights reserved.

musical rhythms from human-performed MIDI data that contain onset time deviations. A rhythm transcription method can also be used as a module of audio-to-score music transcription systems, as studied in [17,20].

A common approach for music transcription is to integrate a score model and a performance/acoustic model to obtain a proper transcription that best fits an input performance signal, similar to the statistical speech recognition method [21]. This is a top-down approach for describing the generative process of musical performance signals using statistical models, in contrast to bottom-up approaches for extracting features from musical performance signals, such as using ratios of inter-onset intervals [13,25] and using the wavelet transform [30] to represent rhythmic features. A major advantage of the former approach is the potential to utilize statistical machine learning techniques.

In constructing a score model for music transcription, capturing both local and global features of musical scores is considered to be effective. Among the local features, the sequential dependences of musical notes can be used to induce output scores that obey the musical grammar. Among the global features, repetitive structures are commonly found in various styles of music [12,26] and can be a useful guide for transcription since they can complement the local information of musical scores. This is because by using a repetitive structure, one can in effect cancel out timing deviations and other “noises” in performances. Conventional score models for music transcription that are designed to capture the local sequential dependence of musical notes include Markov models [10,19,20,22–24,31], hidden Markov models (HMMs) [27], recurrent neural networks (RNNs) [28], and long-short term memory (LSTM) networks [34]. However, it is challenging to construct a score model incorporating a repetitive structure for transcription, particularly because the computational costs for imposing extensive constraints on output scores typically become prohibitively large.

Recently, Nakamura et al. [18] proposed a statistical score model incorporating a repetitive structure and derived a tractable algorithm for music transcription based on the model. A prominent feature of this method is that *piece-specific* score models for individual musical pieces are constructed and subsequently used for transcription, in contrast to the use of a *generic* score model for all target musical pieces in the conventional methods. The model is formulated as a Bayesian model whereby a piece-specific score model is first generated from a generic score model and a musical score is then generated from the piece-specific model (Fig. 1). This model is similar to the topic model of natural language based on the latent Dirichlet allocation (LDA) [5], where a word distribution of individual text is generated from a generic word distribution of a collection of text. Instead of a word distribution, a distribution of note patterns (i.e. subsequences of notes corresponding to bars) is considered as a score model for generating musical scores. The process for generating piece-specific models is described as a Dirichlet process [14] with a small concentration parameter, which induces sparse distributions of note patterns. As a consequence of the sparseness of the piece-specific score model, repetitions of note patterns are naturally induced in the generated score. It is important that the piece-specific score model is stochastically generated and thus the total model can describe unseen scores with an unfixed repetitive structure. Moreover, by combining a process of modifying the generated note patterns, the model can also describe approximate repetitions, which are commonly seen in music practice.

In [18], the efficacy of the score model was tested in the task of the rhythm transcription of monophonic music. It was found that both the Bayesian construction of score models and the note modification process improved the transcription accuracy, showing the potential of the approach. It was also found that the computational costs were too large for practical applications.

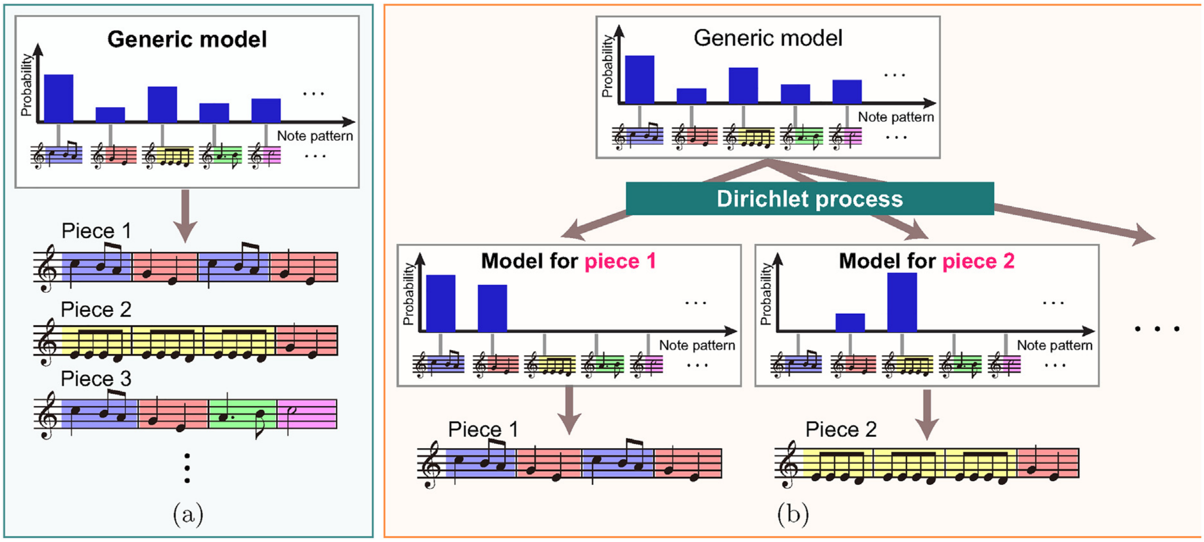

Fig. 1. Schematic illustrations of (a) a conventional score model and (b) the studied score model. In (a), all target scores are generated from a “generic” model. In (b), a piece-specific score model is generated for each individual score, where a repetitive structure is induced by a sparse distribution.

The purpose of this study is to find a score model capturing repetitions that can be practically used for rhythm transcription. We construct wider classes of Bayesian score models and conduct a systematic comparative evaluation of these models to find an optimal model in terms of transcription accuracy and computational costs. While repetitions were considered in units of note patterns in [18], it is theoretically possible to consider repetitions in units of notes. We construct Bayesian score models based on the note value Markov model (MM) [31] and the metrical MM [10,23], which are advantageous for computational efficiency compared to the note pattern MM considered in [18]. We apply the constructed score models to rhythm transcription and conduct evaluations to examine the effect of capturing repetitions and to reveal the best model for the task.

After introducing basic score models for rhythms (note value MM, metrical MM, and note pattern MM) and presenting statistical analyses on the repetitive structure in Section 2, we formulate our models in Section 3. Bayesian extensions of the three types of Markov models and the integration of note modification processes (note divisions and onset shifts) are formulated there. A rhythm transcription method based on the score models is presented in Section 4. Evaluations are carried out in Section 5, where models are compared in terms of the transcription error rate and computation time. We also examine the influence of the parameterization of the relevant hyperparameters. The data and source code used in this study are available at https://bayesianscoremodel.github.io.

2. Repetitive structure and sparseness of piece-specific score models

We here define the form of music data we address, introduce basic score models, and present statistical analyses that indicate the necessity of considering piece-specific score models in constructing musical score models capturing the repetitive structure of music.

2.1. Musical score data

The score data used in this study are extracted from a collection of vocal melodies of popular music. Specifically, we use 99 pieces from the RWC popular music dataset [9], 190 pieces by The Beatles, and 142 contemporary Japanese popular music (J-pop) pieces. To enable comparisons between different models, only pieces in 4/4 time are chosen, only rhythms (note onset positions) are used, the 16th-note length is used as the minimal beat resolution, and each piece is segmented into a half-note length. In this step, we disregard rests, note onsets in finer beat resolutions, and segments without any note onsets. In addition, for simplicity, if a pair of consecutive note onsets in two segments is more than a half-note length apart, a new onset is added at the beginning of the latter segment so that the maximum note length will be a half-note length. Hereafter the obtained segments are called *bars*, that is, pieces are represented in 2/4 time. The relative positions of note onsets in each bar in units of 16th notes are called the *metrical positions*, and the number of metrical positions N_b $(= 8)$ is called the *bar length*. We should remark that the time signature, bar length, and the manipulations made in preparing our data were chosen to balance the precision of the representation and the computational efficiency of numerical experiments; the extension for including longer durations and triplet notes is theoretically possible, as discussed in Section 5.5. Finally, we randomly split the data into two sets, one set for training the models and the other set for evaluating them, whose contents are shown in Table 1.

A *note pattern* (i.e. rhythmic pattern in a bar) can be represented as an N_b-bit binary vector $\sigma_1\sigma_2\cdots\sigma_{N_b}$, where $\sigma_b = 1$ indicates an onset at metrical position b and otherwise $\sigma_b = 0$. The onset time of a note measured from the beginning of a piece in units of 16th notes is called the *onset score time*. A score-notated note length (difference in the onset score time) is called a *note value*. As shown in Fig. 2, we can use one of three quantities (note value, metrical position, and note pattern) to represent musical rhythms. These three representations are equivalent; however, the metrical position of the first note onset is lost in the note value representation.

We represent a musical score as a sequence of onset score times $(\tau_n)_{n=0}^{N}$ where n represents the note onsets and N is the number of notes ($N + 1$ note onsets are necessary to define the lengths of N notes). The onset score times are described in units of 16th-note lengths. In the example of Fig. 2(b), $\tau_0 = 4, \tau_1 = 6, \tau_2 = 8, \tau_3 = 12$, etc. Their absolute values are unimportant but we assume that $\tau_n \pmod{N_b}$ represents the metrical position $b_n \in \{0, \ldots, N_b - 1\}$ of the nth note onset. The note

Table 1
Musical score data used in this study.

Dataset	Original data	Pieces	Bars	Notes
Train	Beatles	180	16,746	40,913
	RWC	89	11,449	31,998
	Contemporary J-pop	132	21,152	66,073
	Total	401	49,347	138,583
Test	Beatles	10	924	2,124
	RWC	10	1,452	3,950
	Contemporary J-pop	10	1,518	4,672
	Total	30	3,894	10,716

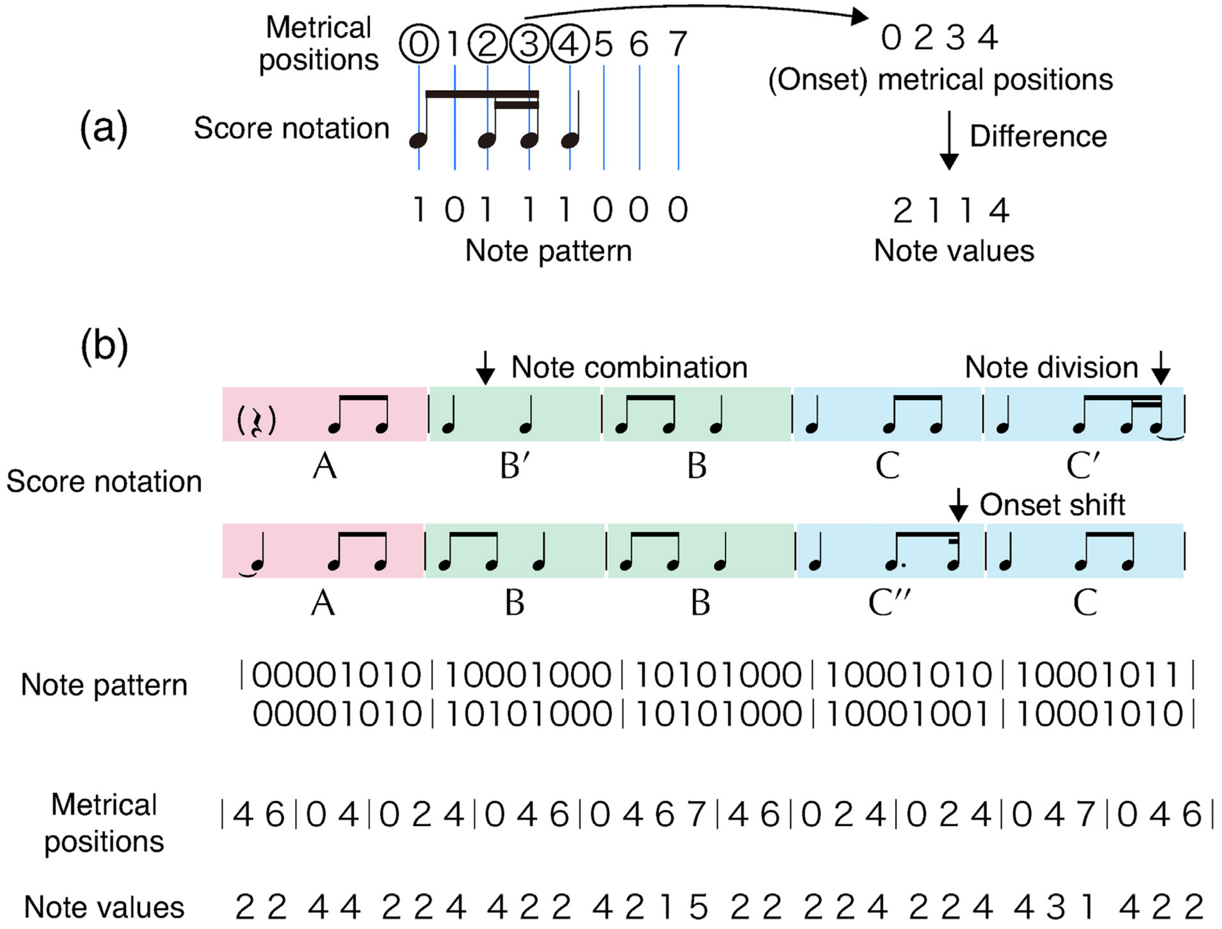

Fig. 2. Representations of musical rhythms. (a) A one-bar example. (b) A longer example. The score is taken from "Yasashii uta (tender song)" by a J-pop band Mr. Children.

value r_n of the nth note ($n = 1, \dots, N$) is defined as $r_n = \tau_n - \tau_{n-1}$. In the example of Fig. 2(b), $b_0 = 4, b_1 = 6, b_2 = 0, b_3 = 4$, etc. and $r_1 = 2, r_2 = 2, r_3 = 4$, etc. Since the maximum note value and N_b are 8, we can also write $r_n = b_n - b_{n-1}$ if $b_n > b_{n-1}$ and $r_n = b_n - b_{n-1} + N_b$ otherwise. Hereafter, we use the notation $\tau_{n_1:n_2} = (\tau_n)_{n=n_1}^{n_2}$ to indicate a sequence of variables.

2.2. Basic score models

A score model is a probabilistic generative model for sequences of onset score times that describe the probability $P(\tau_{0:N})$, or, equivalently, $P(b_{0:N})$ or $P(r_{1:N})$. Here we review three basic score models: the note value MM, the metrical MM, and the note pattern MM.

2.2.1. Note value Markov Models (NoteMM)

The basic first-order note value MM (NoteMM1) [31] is defined with the initial and transition probabilities of note values:

$$P(r_1 = r) = \pi_{\mathrm{ini}\,r}, \quad P(r_n = r \,|\, r_{n-1} = r') = \pi_{r'r}. \tag{1}$$

The zeroth-order note value MM (NoteMM0) is defined by using the unigram probabilities $P(r_n)$ instead of the transition probabilities $P(r_n|r_{n-1})$. The second-order note value MM (NoteMM2) and higher-order models can be constructed by considering transition probabilities of the form $P(r_n|\,r_{n-k}, \dots, r_{n-1})$. Since similar methods can be applied for lower-order and higher-order models [16], we describe only the first-order models in what follows.

2.2.2. Metrical Markov Models (MetMM)

The basic first-order metrical MM (MetMM1) [10,23] generates a sequence of metrical positions as

$$P(b_0 = b) = \chi_{\mathrm{ini}\,b}, \quad P(b_n = b \,|\, b_{n-1} = b') = \chi_{b'b}. \tag{2}$$

The onset score times $(\tau_n)_{n=0}^{N}$ and the note values $(r_n)_{n=1}^{N}$ are then given as

$$\tau_0 = b_0, \quad r_n = \tau_n - \tau_{n-1} = [b_{n-1}, b_n], \tag{3}$$

$$[b_{n-1}, b_n] := \begin{cases} b_n - b_{n-1}, & b_{n-1} < b_n; \\ b_n - b_{n-1} + N_b, & b_{n-1} \geqslant b_n. \end{cases} \tag{4}$$

2.2.3. Note pattern Markov models (PatMM)

The note pattern MMs can be described as a generalization of the metrical MMs. We denote a note pattern of metrical positions as

$$B_k = (B_{k1}, \ldots, B_{kI_k}), \tag{5}$$

where k is an index in the set of note patterns, I_k denotes the number of note onsets in note pattern B_k, and each B_{ki} ($i \in \{1, \ldots, I_k\}$) indicates a metrical position ($B_{ki} < B_{k(i+1)}$). For example, the first two note patterns in Fig. 2(b) are represented as $(4, 6)$ and $(0, 4)$. Note that B_{k1} needs not be 0: if $B_{k1} \neq 0$ it indicates that the first note of pattern B_k is a tied note. This is an extension of the note pattern model used in [18], which cannot describe tied notes.

The basic first-order note pattern MM (PatMM1) is described as a two-level hierarchical MM [8] with a state space indexed by a pair (k, i) of note-pattern index k and an internal index $i \in \{1, \ldots, I_k\}$ of notes in each pattern k. The upper-level model generates a sequence of note patterns as

$$\Gamma_{\mathrm{ini}\,k} = P(k_1 = k), \quad \Gamma_{k'k} = P(k_m = k \,|\, k_{m-1} = k'). \tag{6}$$

The lower-level model generates internal positions as

$$\gamma^k_{\mathrm{ini}\,i} = P(i_1 = i | k) = \delta_{i1}, \tag{7}$$

$$\gamma^k_{i'i} = P(i_{\ell+1} = i \,|\, i_\ell = i', k) = \delta_{(i'+1)i}, \tag{8}$$

$$\gamma^k_{i\,\mathrm{end}} = P(i_{I_k} = i | k) = \delta_{iI_k}, \tag{9}$$

where δ denotes Kronecker's delta. The initial and transition probabilities of the total model are given as

$$\psi_{\mathrm{ini}(ki)} = P(k_1 = k, i_1 = i) = \Gamma_{\mathrm{ini}\,k}\gamma^k_{\mathrm{ini}\,i} = \Gamma_{\mathrm{ini}\,k}\delta_{i1}, \tag{10}$$

$$\psi_{(k'i')(ki)} = P(k_n = k, i_n = i | k_{n-1} = k', i_{n-1} = i') = \delta_{kk'}\gamma^k_{i'i} + \gamma^{k'}_{i'\,\mathrm{end}}\Gamma_{k'k}\gamma^k_{\mathrm{ini}\,i} = \delta_{kk'}\delta_{i(i'+1)} + \delta_{i'I_{k'}}\Gamma_{k'k}\delta_{i1}, \tag{11}$$

in which case we have $b_n = B_{k_n i_n}$. The onset score times and note values are given as in Eqs. (3) and (4).

2.3. Repetitions and sparseness

To quantitatively see how a repetitive structure is reflected in the sparseness of piece-specific score models, a distribution (*generic* model) of note patterns obtained from all the pieces in our training dataset is shown with distributions (*piece-specific* models) obtained from individual pieces in Fig. 3. We use only the training dataset for the analysis in this subsection. The sparseness of the piece-specific distributions, which can be measured by the (Shannon) entropy, is evident. From the entro-

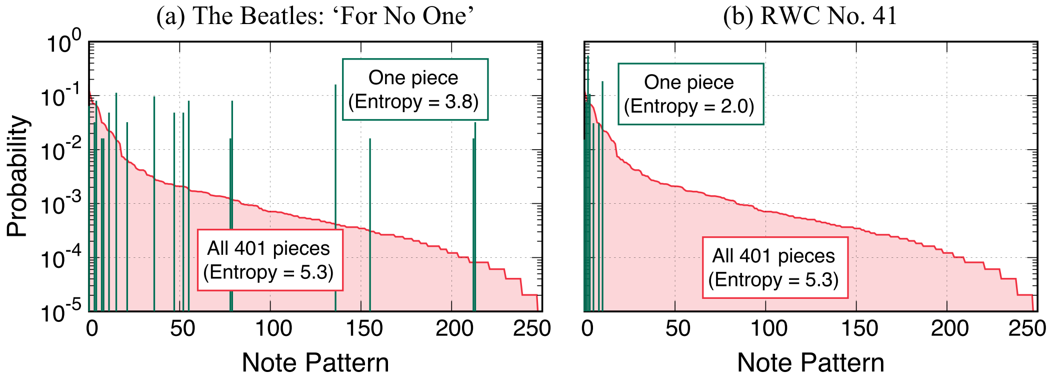

Fig. 3. A generic distribution of note patterns obtained from the training data (background, red) and piece-specific distributions obtained from individual pieces in the same data (foreground, green). The note patterns are ordered according to the probability.

pies of the piece-specific distributions of all the pieces in Fig. 4, we see that most pieces have a much lower entropy compared to the generic distribution with an entropy of 5.3.

The sparseness of the piece-specific distributions cannot be explained merely by using the statistical fluctuation of finite data. To confirm this, we drew 401 sets of samples (note patterns) from the generic distribution, each of which corresponds to a piece in the data and has the same number of bars (the average number of bars was 123). The entropies calculated from the distributions obtained from the sampled data are also plotted in Fig. 4. The result in Fig. 4 shows that these distributions still have much higher entropies than the real data, indicating that the sparseness of the real piece-specific distributions originates from the repetitive structure.

As mentioned above, rhythmic patterns can also be described as sequences of metrical positions or note values (Fig. 2) and repetitions at the note pattern level induce repetitions at the note level and vice versa. This indicates that a repetitive structure is reflected in the sparseness of piece-specific note value MMs and metrical MMs, whose main parameters are the transition probabilities of note values and those of metrical positions, respectively. In Figs. 5(a) and (b), we plot the distributions of the entropies for these models: one for the real data and one for the sampled data from the generic model, respectively. We can again observe that the entropies of the actual piece-specific transition probabilities are much lower than the entropy of the generic model and that this cannot be explained merely by the finite sample effect.

These results support our assertion that a repetitive structure in music can be useful for transcription from the viewpoint of generative modelling. Lower entropy means a higher predictive ability in generative modelling. Thus, if we can infer the piece-specific score model appropriately, the transcription will be easier due to the model's high predictive ability.

2.4. Note modifications for approximate repetitions

In music, repetitions may appear with modifications, leading to approximate repetitions. The example in Fig. 2(b) illustrates the two most common types of modifications: note division/combination and onset shift [18,29]. In a note division/combination (or simply division/combination), a note onset is inserted/deleted while the other note onsets are unchanged. These modifications often appear in sung melodies where the number of syllables varies in repeated phrases or sections. In an onset shift (or simply shift), a note onset is shifted forward or backward in time. This is another typical type of rhythmic variation that includes syncopations.

Modelling these modifications for describing musical scores with approximate repetitions can be useful for transcription because by identifying modifications we can induce more repetitions in data, which can lead to a higher predictive ability. The concept of approximate repetitions is also important for humans to understand or transcribe music [12]. The example in Fig. 2(b) cannot be recognized as repeated phrases if modified note patterns are considered to be completely different ones.

3. Proposed score models

3.1. Overview of the studied models

Here, we explain the proposed score models. We construct score models based on three basic models: the note value MM, the metrical MM, and the note pattern MM. Each basic model is extended in a Bayesian manner to describe the repetitive structure and combined with note modification models to describe approximate repetitions. The incorporation of note mod-

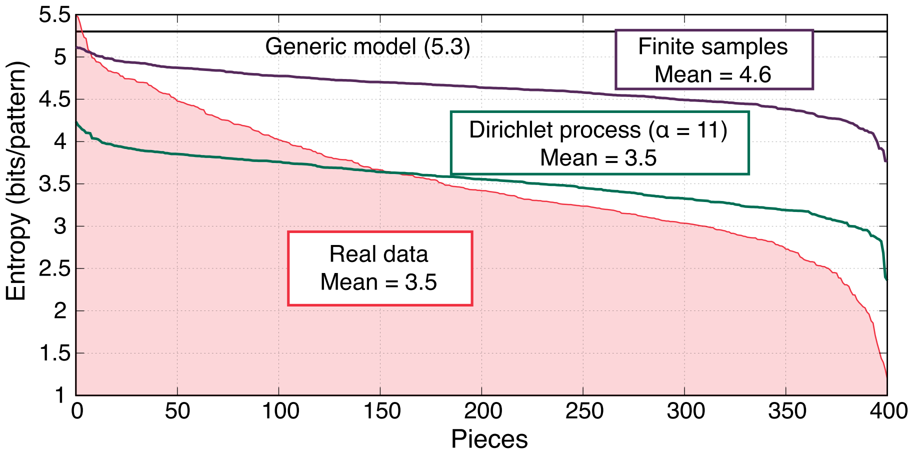

Fig. 4. Distribution of the entropies of piece-specific distributions of note patterns, that of sequences sampled from the generic note pattern distribution, and that of distributions generated from the Dirichlet process. The pieces are ordered according to their entropy.

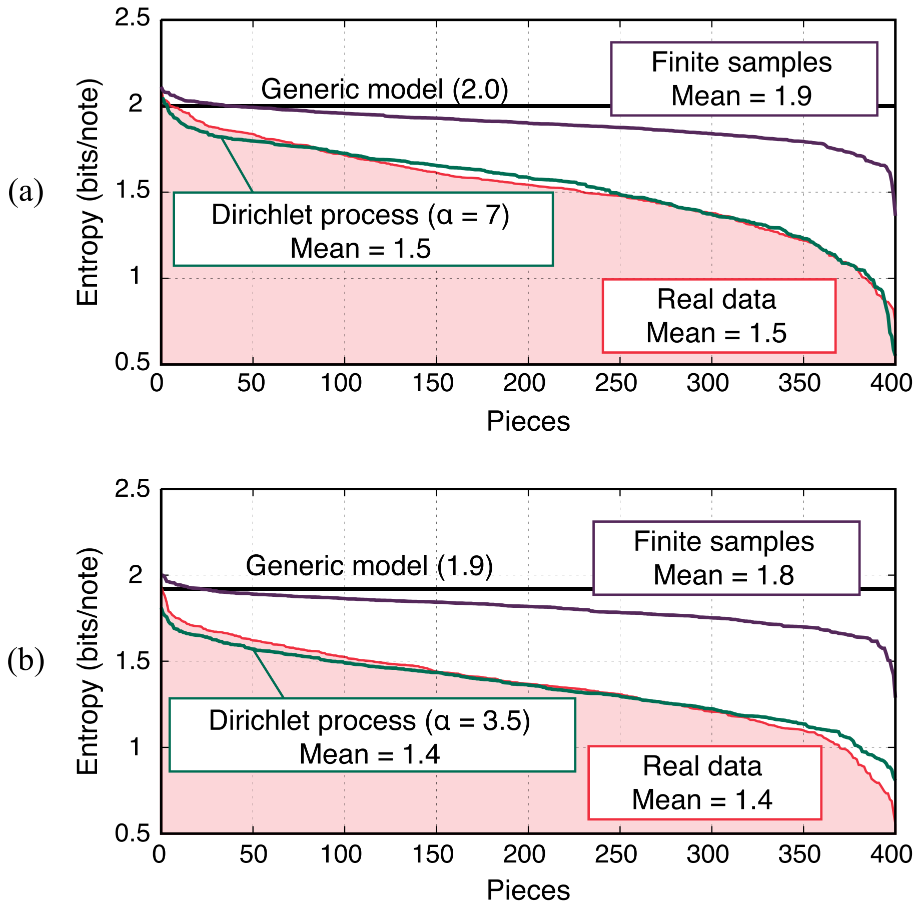

Fig. 5. Distributions of the entropies of the piece-specific transition probabilities of (a) note values and (b) metrical positions. The corresponding distributions obtained from sampled data from the generic transition probabilities and those obtained from the transition probabilities generated from the Dirichlet process are also shown. The pieces are ordered according to their entropy.

ifications is described in Section 3.2 and the Bayesian extensions are formulated in Section 3.3. The list of constructed models based on the note value MM and their acronyms are summarized in Table 2. Similarly we construct 9 models (MetMM0, MetMM0B, . . ., MetMM2B) based on the metrical MM and 7 models (PatMM0, PatMM0B, . . ., PatMM1SDB) based on the note pattern MM. We do not consider the second-order note pattern MM and its extensions due to their impractical computational cost.

Table 2
List of the constructed score models based on the note value Markov model (MM). In the acronyms, the number indicates the order of the Markov model, 'S' and 'D' indicate note shift and note division, and 'B' indicates Bayesian extension.

Acronym	MM order	Bayesian	Modification
NoteMM0	0th		–
NoteMM0B	0th	✓	–
NoteMM1	1st		–
NoteMM1B	1st	✓	–
NoteMM1SB	1st	✓	Shift only
NoteMM1DB	1st	✓	Division only
NoteMM1SDB	1st	✓	Shift & division
NoteMM2	2nd		–
NoteMM2B	2nd	✓	–

3.2. Incorporation of note modifications

3.2.1. Note modification operations

We consider the most common types of note modifications, onset shifts, note divisions, and note combinations (Section 2.4). As note divisions and combinations are the opposite of each other, we only consider note divisions in our models. In [18], only onset shifts for notes on bar onsets (i.e. syncopations) were considered. Here, we generalize the idea and consider onset shifts for any notes.

We denote the shift of the onset score time of the nth note as s_n and the note value and score time of the nth note before the application of onset shifts are notated as $\tilde{r}_n$ and $\tilde{\tau}_n$, respectively (Fig. 6 (right)). An onset shift is then described as

$$\tilde{\tau}_n \to \tau_n = \tilde{\tau}_n + s_n \quad \text{or} \quad \tilde{r}_n \to r_n = \tilde{r}_n + s_n - s_{n-1}. \tag{12}$$

The shift variable s_n describes the amount of the shift and takes values in the range $[-N_b + 1, N_b - 1]$ with probability

$$\xi_s = P(s_n = s). \tag{13}$$

We assume a constraint $-\tilde{r}_n < s_n \leqslant \tilde{r}_n$ to make the onset shift interpretable as a modification. Musically, we should also have $r_n = \tilde{r}_n + s_n - s_{n-1} > 0$ for all n.

In the most general case, a note division is described as an operation $\tilde{r} \to \boldsymbol{q}_h$ that maps a note value $\tilde{r}$ to a sequence of note values $\boldsymbol{q}_h = q_{h1} \dots q_{hG_h}$ (Fig. 6 (left)). Here, h labels the patterns of note divisions and G_h is the number of notes after a division. The division probability can be written as

$$\zeta_{\tilde{r}h} = P(\tilde{r} \to \boldsymbol{q}_h). \tag{14}$$

We must have $\zeta_{\tilde{r}h} = 0$ unless $\tilde{r} = q_{h1} + \cdots + q_{hG_h}$. Using an index $g \in \{1, \dots, G_h\}$, the pair of indices (h, g) can specify the note value after applying the note division as $r_n = q_{h_n g_n}$.

When we combine onset shifts and note divisions, we first apply note divisions and then apply an onset shift to each of the divided notes. The resulting notes are indexed by $(\tilde{r}, h, g, s)$, and their note values are given as

$$r_n = q_{h_n g_n} + s_n - s_{n-1}. \tag{15}$$

In this study, we only consider note divisions into two notes, that is, $G_h \leq 2$, to avoid impractical computational costs.

3.2.2. Score models with note modifications

Note value MMs can be extended to incorporate note modifications as follows. The model incorporating onset shifts (NoteMM1S) is described by an HMM with a state space indexed by $z_0 = s_0$ and $z_n = (\tilde{r}_n, s_n)$ $(n = 1, \dots, N)$, where $\tilde{r}_n$ indicates the note value before modification and s_n indicates the amount of the shift. The initial and transition probabilities are given as

$$P(z_0 = s) = \xi_s, \quad P(z_1 = (\tilde{r}, s)) = \pi_{\mathrm{ini}\,\tilde{r}}\, \xi_s, \tag{16}$$

$$P(z_n = (\tilde{r}, s) \,|\, z_{n-1} = (\tilde{r}', s')) = \pi_{\tilde{r}'\tilde{r}}\, \xi_s, \tag{17}$$

and the final output is obtained from the output probability

$$P(r_n | z_{n-1}, z_n) = \mathbb{I}(r_n = \tilde{r}_n + s_n - s_{n-1}), \tag{18}$$

where $\mathbb{I}(\mathscr{C})$ is 1 when expression $\mathscr{C}$ is true and is 0 otherwise.

The note value MM incorporating note divisions (NoteMM1D) is described by an HMM with a state space indexed by $z = (\tilde{r}, h, g)$, where $\tilde{r}$ indicates the note value before note division and h and g indicate a note-division pattern and its internal note position, respectively (see Section 3.2.1). The initial and transition probabilities are given as

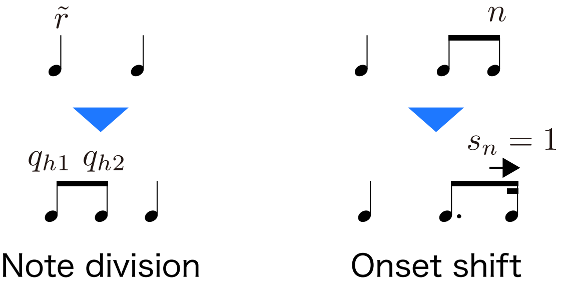

Fig. 6. The note division operation (left) and the onset shift operation (right).

$$P(z_1=(\tilde{r},h,g))=\pi_{\mathrm{ini}\,\tilde{r}}\zeta_{\tilde{r}h}\delta_{g1},\tag{19}$$

$$P(z_n=(\tilde{r},h,g)\,|\,z_{n-1}=(\tilde{r}',h',g'))=\delta_{\tilde{r}\tilde{r}'}\delta_{hh'}\delta_{g(g'+1)}+\delta_{g'G_{h'}}\pi_{\tilde{r}'\tilde{r}}\zeta_{\tilde{r}h}\delta_{g1},\tag{20}$$

and the final output is obtained from the output probability

$$P(r_n|z_n)=\mathbb{I}(r_n=q_{h_ng_n}).\tag{21}$$

The note value MM incorporating both onset shifts and note divisions (NoteMM1SD) is obtained by combining the above two models. It is described by an HMM with a state space indexed by $z_0=s_0$ and $z_n=(\tilde{r}_n,h_n,g_n,s_n)$ $(n=1,\ldots,N)$ and the initial and transition probabilities are given as

$$\begin{aligned}&P(z_0=s)=\xi_s,\quad P(z_1=(\tilde{r},h,g,s))=\pi_{\mathrm{ini}\,\tilde{r}}\zeta_{\tilde{r}h}\delta_{g1}\xi_s,\\&P(z_n=(\tilde{r},h,g,s)\,|\,z_{n-1}=(\tilde{r}',h',g',s'))=\left(\delta_{\tilde{r}\tilde{r}'}\delta_{hh'}\delta_{g(g'+1)}+\delta_{g'G_{h'}}\pi_{\tilde{r}'\tilde{r}}\zeta_{\tilde{r}h}\delta_{g1}\right)\xi_s.\end{aligned}\tag{22}$$

The final output is obtained from the output probability

$$P(r_n|z_{n-1},z_n)=\mathbb{I}(r_n=q_{h_ng_n}+s_n-s_{n-1}).\tag{23}$$

The model parameters of the (non-Bayesian) first-order note value models are summarized as follows. Hereafter we use r instead of $\tilde{r}$ to unify the notation. Let us write $\boldsymbol{\pi}_{\mathrm{ini}}=(\pi_{\mathrm{ini}\,r})_r,\boldsymbol{\pi}_r=(\pi_{rr'})_{r'},\boldsymbol{\zeta}_r=(\zeta_{rh})_h$, and $\boldsymbol{\xi}=(\xi_s)_s$.

- NoteMM1: $\boldsymbol{\pi}_{\mathrm{ini}},(\boldsymbol{\pi}_r)_r$.
- NoteMM1S: $\boldsymbol{\pi}_{\mathrm{ini}},(\boldsymbol{\pi}_r)_r,\boldsymbol{\xi}$.
- NoteMM1D: $\boldsymbol{\pi}_{\mathrm{ini}},(\boldsymbol{\pi}_r)_r,(\boldsymbol{\zeta}_r)_r$.
- NoteMM1SD: $\boldsymbol{\pi}_{\mathrm{ini}},(\boldsymbol{\pi}_r)_r,(\boldsymbol{\zeta}_r)_r,\boldsymbol{\xi}$.

We can similarly formulate four models (MetMM1, MetMM1S, MetMM1D, and MetMM1SD) based on the metrical MM with or without the note modification process and four models (PatMM1, PatMM1S, PatMM1D, and PatMM1SD) based on the note pattern MM. See the Supplemental Material for the explicit formulation.

3.3. Bayesian extensions based on Dirichlet processes

We extend the score models in a Bayesian manner to formulate the generative process of piece-specific models from a generic model. We assume that piece-specific models and generic models have the same architecture but different parameterizations. For NoteMM1SD, the generative process can be formulated by assigning prior distributions to the model parameters:

$$\boldsymbol{\pi}_{\mathrm{ini}}\sim\mathrm{DP}\;(\alpha_{\mathrm{ini}},\bar{\boldsymbol{\pi}}_{\mathrm{ini}}),\quad\boldsymbol{\pi}_r\sim\mathrm{DP}\;(\alpha_\pi,\bar{\boldsymbol{\pi}}_r),\quad\boldsymbol{\zeta}_r\sim\mathrm{DP}\;(\alpha_\zeta,\bar{\boldsymbol{\zeta}}_r),\quad\boldsymbol{\xi}\sim\mathrm{DP}\;(\alpha_\xi,\bar{\boldsymbol{\xi}}),\tag{24}$$

where DP denotes a Dirichlet process [14]; $\bar{\boldsymbol{\pi}}_{\mathrm{ini}},\bar{\boldsymbol{\pi}}_r$, etc. denote base distributions; and $\alpha_{\mathrm{ini}},\alpha_\pi$, etc. denote concentration parameters. Since the distributions are finite discrete distributions, the Dirichlet process can be described as Dirichlet distributions. For example,

$$\boldsymbol{\pi}_r\sim\mathrm{DP}(\alpha_\pi,\bar{\boldsymbol{\pi}}_r)=\mathrm{Dir}(\alpha_\pi\bar{\boldsymbol{\pi}}_r).\tag{25}$$

The expectation value of the distributions generated by a Dirichlet process is equal to the base distribution [14]. Based on this fact, we can interpret the distributions ($\boldsymbol{\pi}_r,\boldsymbol{\zeta}_r$, etc.) generated by the Dirichlet process as the piece-specific models and the base distributions ($\bar{\boldsymbol{\pi}}_r,\bar{\boldsymbol{\zeta}}_r$, etc.) as the generic model representing the average of all the piece-specific models. The distributions generated from a Dirichlet process become sparser as we use a smaller concentration parameter. The parameters of the note modification models $\boldsymbol{\zeta}_r$ and $\boldsymbol{\xi}$ are also considered for individual pieces to describe the situation in which the note modifications used in each piece have different statistical tendencies.

We can similarly construct Bayesian extensions of metrical MMs and note pattern MMs. For metrical MMs, we have

$$\boldsymbol{\chi}_{\mathrm{ini}}\sim\mathrm{DP}(\alpha_{\mathrm{ini}},\bar{\boldsymbol{\chi}}_{\mathrm{ini}}),\quad\boldsymbol{\chi}_b\sim\mathrm{DP}(\alpha_\chi,\bar{\boldsymbol{\chi}}_b),\tag{26}$$

$$\boldsymbol{\zeta}_r\sim\mathrm{DP}(\alpha_\zeta,\bar{\boldsymbol{\zeta}}_r),\quad\boldsymbol{\xi}\sim\mathrm{DP}(\alpha_\xi,\bar{\boldsymbol{\xi}}),\tag{27}$$

and for note pattern MMs, we have

$$\boldsymbol{\Gamma}_{\mathrm{ini}}\sim\mathrm{DP}(\alpha_{\mathrm{ini}},\bar{\boldsymbol{\Gamma}}_{\mathrm{ini}}),\quad\boldsymbol{\Gamma}_k\sim\mathrm{DP}(\alpha_\Gamma,\bar{\boldsymbol{\Gamma}}_k),\tag{28}$$

$$\boldsymbol{\zeta}_r\sim\mathrm{DP}(\alpha_\zeta,\bar{\boldsymbol{\zeta}}_r),\quad\boldsymbol{\xi}\sim\mathrm{DP}(\alpha_\xi,\bar{\boldsymbol{\xi}}).\tag{29}$$

where $\bar{\boldsymbol{\chi}}_{\mathrm{ini}},\bar{\boldsymbol{\chi}}_b,\bar{\boldsymbol{\Gamma}}_{\mathrm{ini}},\bar{\boldsymbol{\Gamma}}_k$, etc. denote the base distributions corresponding to the generic models and $\alpha_{\mathrm{ini}},\alpha_\chi,\alpha_{\mathrm{ini}},\alpha_\Gamma$, etc. denote the corresponding concentration parameters.

In Fig. 4, the entropies of the distributions of the note patterns (zeroth-order note pattern MMs) generated by a Dirichlet process are shown, where the base distribution is the generic model and the concentration parameter is adjusted to match the average entropy with the real data. Although the obtained distribution of entropies does not very well match the actual distribution, these distributions overlap much more than the actual distribution does with the entropy distribution of the sampled data from the generic model. Similarly, in Figs. 5(a) and (b), we plot the distributions of the entropies for the piece-specific note value MMs and metrical MMs obtained from the Dirichlet processes. We see that the Dirichlet processes can generate transition probabilities with a similar entropy to the entropies of the actual piece-specific transition probabilities. These results indicate that we can use small concentration parameters to account for the sparse piece-specific transition probabilities in the real data.

4. Rhythm transcription method

The aim of a rhythm transcription method is to estimate the score onset times $\tau_{0:N}$ (or note values $r_{1:N}$) from input MIDI data with note onset times $t_{0:N}$ (or durations $d_{1:N}$, where $d_n = t_n - t_{n-1}$) that are represented in units of seconds. We first construct a musical performance generative model that describes the probability $P(\tau_{0:N}, t_{0:N})$ or $P(r_{1:N}, d_{1:N})$ by combining a score model (one of the aforementioned models) and a performance model explained in Section 4.1. We then derive a rhythm transcription algorithm that can estimate the score by maximizing the probability $P(\tau_{0:N}|t_{0:N})$ or $P(r_{1:N}|d_{1:N})$, which is explained in Section 4.3. For the Bayesian score models, the model parameters for the piece-specific model are learned in the transcription step, as described in Section 4.2.

4.1. Performance model

A performance (timing) model describes the generative process of durations $d_{1:N}$ from note values $r_{1:N}$, which gives the probability $P(d_{1:N}|r_{1:N})$. We consider a simplified model with a constant (inverse) tempo v. Given a note value r_n for the nth note, the corresponding duration d_n is generated as

$$d_n \sim \mathrm{Gauss}(v r_n, \sigma_t^2), \tag{30}$$

where $\mathrm{Gauss}(\mu, \Sigma)$ denotes a Gaussian distribution with mean μ and variance Σ, and σ_t represents the amount of onset time deviations. The onset times $t_{0:N}$ can be determined by the durations $d_{1:N}$ up to an initial time t_0, which is insignificant.

In a real music transcription situation, the global tempo is unknown and the tempo changes over time, especially in solo performances and classical musical performances. The performance model can be generalized to describe these situations by introducing an additional process to generate a sequence of local tempos v_n [19]. To focus on the effect of musical score models, we only consider a constant and known tempo in this study.

4.2. Parameter learning

For non-Bayesian score models, the model parameters are pretrained or preset and are fixed during the transcription step. The initial and transition probabilities can be directly learned from musical score data. The probabilities $\bar{\zeta}_r$ and $\bar{\xi}$ of the modification models cannot be directly learned from musical score data, and thus they are manually preset to reflect the belief that modifications are rare. Specifically, we assign probabilities $\bar{\xi}_0$ and $\bar{\zeta}_0$, which are slightly less than unity, for cases without modifications, and other entries in $\bar{\zeta}_r$ and $\bar{\xi}$ are assumed to have uniform probabilities.

For Bayesian models, the hyperparameters of the prior distributions except for the concentration parameters are pretrained or preset to the parameterization of the corresponding non-Bayesian models. The concentration parameters can in principle be optimized according to the transcription accuracy. In our evaluation in Section 5.2, they are preset to a fixed value, and how the values of the concentration parameters for the unigram or transition probabilities influence the transcription accuracy is studied in Section 5.4.

The other parameters of the Bayesian models, namely the parameters and internal variables of a piece-specific score model, are treated as variables and estimated in the transcription step according to the input MIDI data. Let $X = d_{1:N}$ (or $t_{0:N}$) denote the input MIDI data, Z denote the set of internal variables of a score model including note values, Θ denote the set of parameters of the piece-specific score model, and Ξ denote the set of hyperparameters. For NoteMM1S, for example, $Z = (s_0, \tilde{r}_1, s_1, \ldots, \tilde{r}_N, s_N)$, $\Theta = (\boldsymbol{\pi}_{\mathrm{ini}}, \boldsymbol{\pi}_r, \xi)$, and $\Xi = (\alpha_{\mathrm{ini}}, \bar{\boldsymbol{\pi}}_{\mathrm{ini}}, \alpha_\pi, \bar{\boldsymbol{\pi}}_r, \alpha_\xi, \bar{\xi})$. The generative process can be summarized as

$$P(X, Z, \Theta\,|\,\Xi) = P(X|Z)P(Z|\Theta)P(\Theta|\Xi). \tag{31}$$

Here, the first factor on the right-hand side represents the performance model (Section 4.1), the second factor represents the (non-Bayesian part of the) score model (Sections 2.2 and 3.2), and the last factor represents the prior distributions (Section 3.3). Parameters Θ of the piece-specific score model reflecting both the piece-specific distribution of the rhythmic patterns inferred from the observation X and the prior distribution, which induces the sparseness of Θ, can be obtained by maximizing Eq. (31). Since an analytical solution to this inference problem is impossible, as is typically the case for Bayesian models with latent variables, we use the Gibbs sampling method to estimate Θ [4]. Using the Gibbs sampling method, we can draw samples from the posterior probability $P(Z, \Theta\,|X, \Xi)$. After some iterations, we obtain the optimal parameters Θ that maximizes the likelihood $P(X|\Theta) = \sum_Z P(X|Z)P(Z|\Theta)$ among the sampled parameters. For a simple case (the first-order metrical MM), an explicit algorithm using Gibbs sampling is presented in Appendix A. Although the algorithmic details of the Gibbs sampling method are important for the implementation of the models, they are derived by means of standard techniques and are not essential for understanding the main results of this study. Thus, for the other models, we present those details with the explicit formulations of the integrated models in the Supplemental Material.

4.3. Algorithms for rhythm transcription

Our generative models for rhythm transcription constructed as combinations of a score model and a performance model are HMMs whose latent states are the variables of the score model and whose observed variables are onset times or durations of input (human-performed) MIDI data. Thus, the standard Viterbi algorithm [21] can be used to estimate the variables of the score model, including the note values or metrical positions, from the input MIDI data. For Bayesian models, the model parameters are first estimated from the input signal using the method explained in Section 4.2, and the latent variables are then estimated by the Viterbi algorithm. As an example, an explicit algorithm for rhythm transcription based on the first-order Bayesian metrical MM (MetMM1B) is presented in Appendix.A.

For some of the present models, especially PatMM1DB and PatMM1SDB, the latent state space is huge, and the Viterbi algorithm requires an impractical amount of computation time. To address this issue, we apply a beam search and only retain the top-W most likely states at each Viterbi update (W is a preset number called the beam width). Using N_Z to represent the number of latent states, the computational costs are reduced to $O(N_Z W)$ from $O(N_Z^2)$ via this technique, while the chance of obtaining the globally optimal solution decreases for smaller Ws. Since the inferential precision increases as we increase W, its value should be set to a practical upper limit for tractable computation. For Bayesian extensions of these models, we also apply a similar method in the forward algorithm used for estimating the posterior probability.

5. Evaluation

To evaluate the performance of the studied models, we conducted two evaluation experiments. In the first experiment (Section 5.2), in order to compare the effects of different model architectures, the predictive ability of the non-Bayesian score models is evaluated. In the second experiment (Section 5.3), the transcription accuracies of the models are measured to examine the effects of the Bayesian extensions and modification models. We also examine the influence of the hyperparameters of the studied models in Section 5.4. Using the measurement of the computation time of the models, in Section 5.5, we discuss which models are practically important.

5.1. Setup

The initial and transition probabilities of all the models were trained with the same training dataset (Section 2.1). The probabilities were estimated by the maximum likelihood method with an additive smoothing constant of 0.1. We confirmed that the results were not sensitive to the smoothing constant, except for the transition probabilities of PatMM1 for which there were more parameters than the number of training samples. For this model we applied a linear interpolation with the unigram probabilities (0.8 unigram probability + 0.2 transition probability) that was roughly optimized w.r.t. the transcription accuracy.

We used for the evaluations the test dataset consisting of melodies taken from 30 pieces of popular music (Section 2.1). For the transcription experiments, we collected performed MIDI data played by amateur musicians and recorded them with a digital piano. Four musicians participated and were exclusively assigned some pieces in the test dataset. The first 40 bars of each piece were used, and the musicians were asked to play the scores in a tempo of 105 bpm (beats per minute). For systematic examinations, we also used synthetic MIDI data generated by the performance timing model in Section 4.1 with $\sigma_t = 0.04$ sec and a tempo of 144 bpm (the value of σ_t was determined to simulate human performance and the tempo was determined so that the transcription difficulty roughly matches that for real MIDI data). All pieces in the test dataset were used for the synthetic data.

5.2. Evaluation of the score models

We evaluated the non-Bayesian score models listed in Table 3 in terms of the cross entropy, which is a standard measure for quantifying the performance of language models. Lower entropy means a higher predictive ability. For all classes of models (note value MM, metrical MM, and note pattern MM) the cross entropy decreased as we increased the order. Comparing the note value MMs and metrical MMs, the former outperformed when the order was zero, but the latter outperformed for

Table 3
Cross entropies (CEs) (bits/symbol) of the non-Bayesian score models.

Model	CE	Model	CE	Model	CE
NoteMM0	2.29	NoteMM1	2.07	NoteMM2	1.93
MetMM0	2.63	MetMM1	2.01	MetMM2	1.87
PatMM0	1.97	PatMM1	1.89		

the higher-order cases. The zeroth-order note pattern MM has lower cross entropy than the first-order note value MM and metrical MM. This demonstrates the strength of the note pattern models, which can capture long-range sequential dependence at the note level. The reason the first-order note pattern MM was worse than the second-order metrical MM is likely because the data sparseness was more severe for the former model.

5.3. Evaluation of the transcription accuracy

We used the synthetic data in addition to the real data of human performances to clearly examine the effects of the score models. The following setups were used for the rhythm transcription algorithms based on the studied models. For the evaluations using the synthetic data, we fixed the parameter σ_t to its actual value (0.04 sec) used for the synthesis; and for the real data, we set $\sigma_t = 0.035$ sec, which was roughly optimized in the preliminary experiments. All the concentration parameters were set to 10. To estimate the model parameters of the Bayesian models, we iterated the Gibbs sampling 100 times, except for the models with too large of computational costs. For these models (MetMM1SDB, PatMM1DB, and PatMM1SDB), we iterated 20 times. The influence of varying the values for these parameters is investigated in Section 5.4. The probability parameters for note modifications were set as $\bar{\xi}_0 = \bar{\zeta}_0 = 0.9$, reflecting the assumption that note modifications are rare. For PatMM1DB and PatMM1SDB, we set the beam width as $W = 200$. This value was a practical upper limit to run all experiments in a reasonable amount of time. We use the error rate (i.e. the ratio of the number of notes with incorrectly estimated note values and the total number of notes) as an evaluation measure. Since note values can be obtained from metrical positions, this evaluation measure can be applied universally for the transcribed results of all the models. Since the transcription results depend on random numbers used for Gibbs sampling for the Bayesian models, we run the methods 10 times with different random number seeds and obtained the average error rates.

The results are shown in Fig. 7. We first discuss the results for the synthetic data. For non-Bayesian models, higher-order models outperformed the corresponding lower-order models. The rank of the transcription accuracy approximately matches the rank of the cross entropy in Table 3. The Bayesian models without modification models had significantly lower error rates than the corresponding non-Bayesian models. The effect of the Bayesian extension was often larger than that of increasing the order of the models. For example, MetMM0B outperformed MetMM1 and MetMM1B outperformed MetMM2. Regarding the effects of the modification models, incorporating note divisions decreased the error rate for all model types, incorporating onset shifts did so only for the metrical MM, and incorporating both operations was no more effective than incorporating

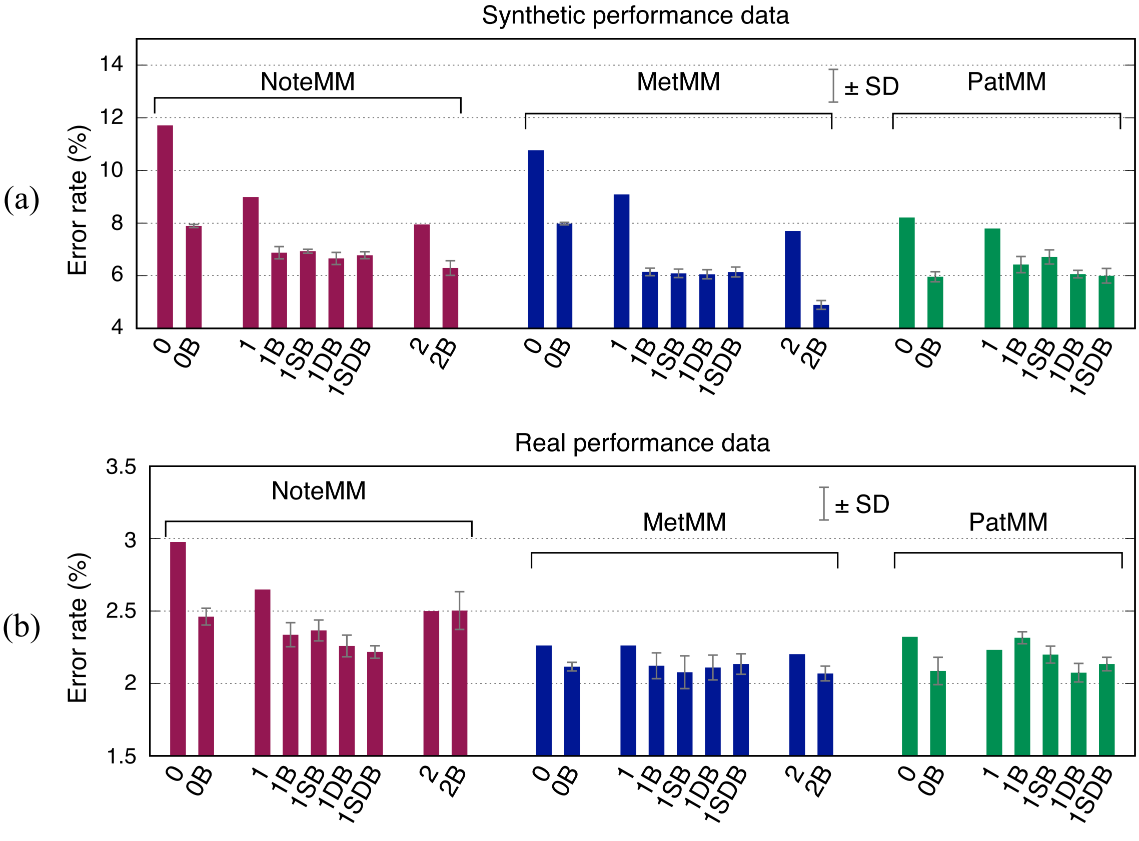

Fig. 7. Transcription error rates for (a) the synthetic MIDI data and for (b) the real performed MIDI data. See Table 2 for the model labels.

only note divisions for all model types. These results show that the incorporation of modification models can often improve the transcription results but not significantly.

Comparing the different model types, the metrical MMs consistently outperformed the corresponding note value MMs in most cases. The accuracies of PatMM0 and PatMM1 were close to that of MetMM2, and the effect of the Bayesian extension was larger for the metrical MMs. The best model in terms of accuracy was MetMM2B.

For the real data, most Bayesian models again significantly outperformed the corresponding non-Bayesian models. Incorporating the modification models was effective for some cases, but there were also cases where the error rates increased. We should notice that the error rates seem to saturate around 2.0% and many models remain within the range of statistical fluctuation. Particularly, the differences in the error rates between MetMM1 and MetMM2 and between MetMM1B and MetMM2B are much smaller than the cases with the synthetic data. This fact also makes it difficult to clearly identify the best models. MetMM2B again had the lowest error rate, but many other models had similar error rates. The note value MMs were consistently worse than the corresponding metrical MMs.

Our results differ from the results of [18] in one respect. Paper [18] reported a significant decrease in the error rate by incorporating the modification models into the note pattern MM. This can be explained by the fact that the state space of the note pattern MM in [18] did not span all possible note patterns and the search space was extended by incorporating note modifications. This bias towards decreasing the performance of the basic note pattern MM can be clearly seen in the evaluation results ([18], Table 1), and it is removed in our results by constructing note pattern MMs with a search space equivalent to other types of MMs.

We manually analysed the estimation errors made by MetMM2B, which had the lowest error rates, for the real data. There were approximately 70 errors (corresponding to an error rate of 2.08%) in the transcription results and 43 of them were clear performance errors. Note that a performance error of one note onset usually induces two estimation errors of note values. The remaining estimation errors were caused by less clear but significant timing deviations. There was a performance where complex rhythms with frequent tied notes were not accurately played, and the model made 18 estimation errors. From these error analyses, we found that the limiting transcription accuracy of approximately 2% is reasonable for this test dataset.

The transcription examples in Fig. 8 demonstrate the effect of the Bayesian learning of piece-specific score models. This piece has repetitions of a tied dotted rhythm (see the first bars of the segments in Fig. 8), which are played with inaccurate timings. Since this rhythm is not common, the MetMM2 method using a generic score model made errors when the timing deviations were too large. In contrast, the MetMM2B method was able to recognize the correct rhythm in most cases by inducing the piece-specific score model capturing these repetitions and in effect cancelling out the timing deviations.

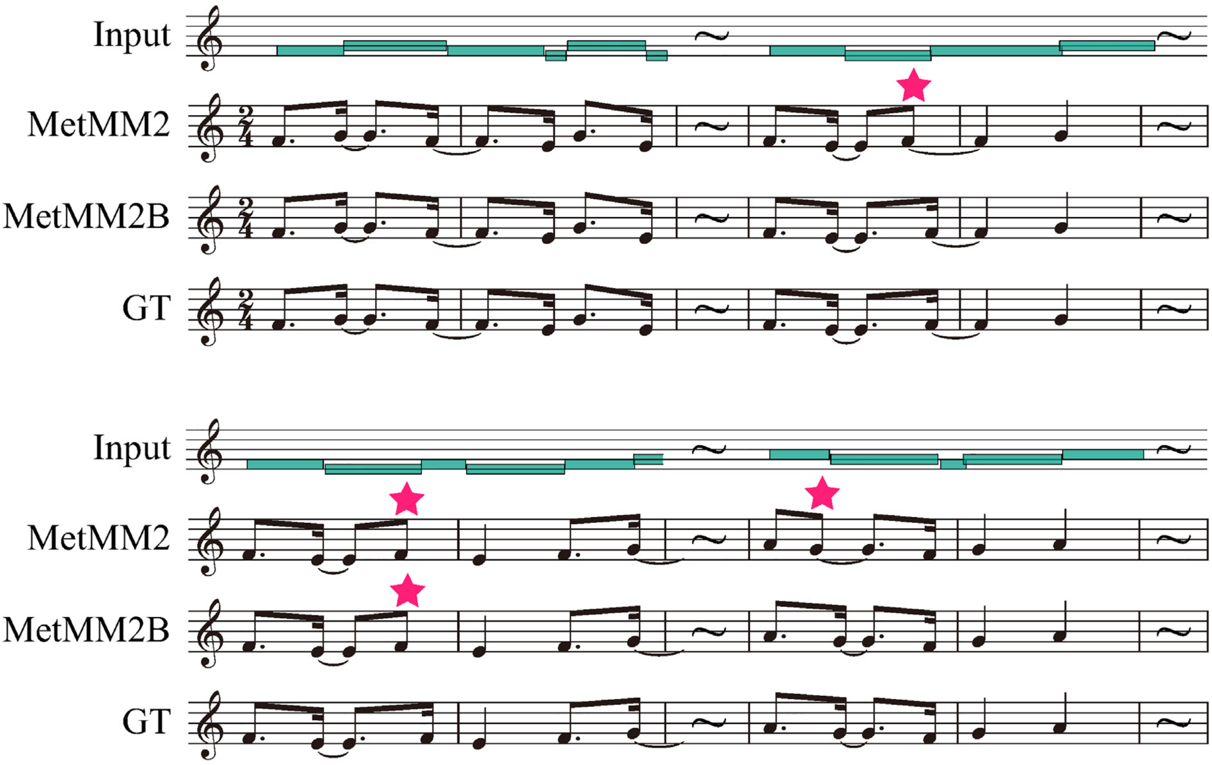

Fig. 8. Examples of transcription results (RWC No. 89). Only bars with repeated rhythms, which are of our interest, are shown. GT refers to the ground truth score. Transcription errors are indicated by stars.

5.4. Influence of the hyperparameters

We studied the influence of the values of the hyperparameters on the transcription accuracy. Here, we focus on the Bayesian metrical MMs that do not incorporate the modification process to investigate the generic tendencies because the non-Bayesian models have poorer performance than the Bayesian models, and the Bayesian note pattern MMs and other Bayesian models with modification models have practically prohibitive computational costs for precise measurements.

The effects of varying the values of the three most relevant parameters, including the concentration parameter for the unigram/transition probabilities, the number of iterations for Gibbs sampling, and the parameter σ_t of the performance model, were investigated by measuring the transcription error rates as in Section 5.3. When varying one of these parameter values, the other parameters were fixed to the values used in Section 5.3. For the plots in Fig. 9, we run each algorithm with 10 different random number seeds and obtained the average error rate. For the concentration parameter and the number of

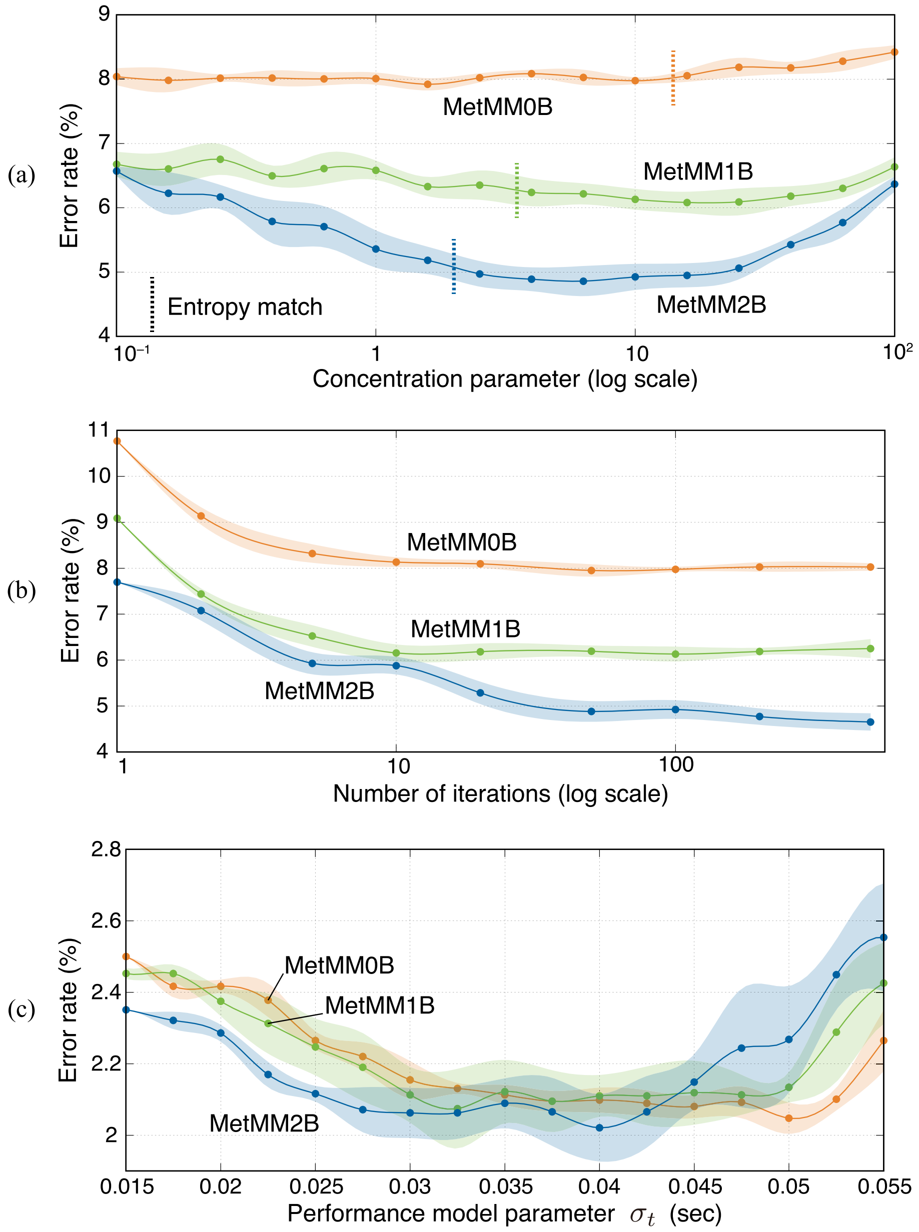

Fig. 9. Transcription error rates for varying (a) the concentration parameter of the unigram/transition probabilities, (b) the number of iterations for the Gibbs sampling, and (c) the performance model parameter σ_t. The solid curves indicate the mean values and the shades indicate the ranges of ±1 standard deviation.

iterations, we used the synthetic MIDI data to obtain results with small statistical fluctuations; and for performance model parameter σ_t, we used the real performed MIDI data.

Regarding the concentration parameter of the unigram/transition probabilities (Fig. 9(a)), the optimal values for all the models were in the range $[1, 100]$ but the values differ among the models. The theoretically optimal values at which the expectation value of the entropies of the unigram/transition probabilities generated by the Dirichlet processes match the corresponding mean entropy of the piece-specific unigram/transition probabilities of the training data are also shown there. We see that the theoretically optimal values do not necessarily match the empirically optimal values that maximize the transcription accuracy. We can also confirm that the differences in the error rates for these values and the value 10 used in Section 5.3 are within the range of one standard deviation of the statistical fluctuation.

Regarding the number of iterations of the Gibbs sampling (Fig. 9(b)), the error rate monotonically decreased as the number of iterations increased until a plateau was reached for each model. For MetMM0B and MetMM1B, the plateau was already reached with approximately 10 iterations; and for MetMM2B, the error rate seemed to continue decreasing still at the 500th iteration. Since the computation time for the parameter estimation is approximately proportional to the number of iterations, the trade off between the computation time and estimation precision is relevant, especially for higher-order models.

Regarding the performance model parameter σ_t (Fig. 9(c)), we see that different models had different optimal values in the range $[0.03, 0.05]$ sec, but with a nonmonotonic behaviour and a large statistical fluctuation. From these results, we understand that it is difficult to reliably find the precise optimal value of σ_t for the real data and that the tentative value 0.035 sec used in Section 5.3 is at least not a bad choice.

5.5. Discussion of the computation time and extendability

We now discuss the computation time, which is another important aspect of transcription methods. The computation times for the studied models are listed in Table 4. These values were measured on a computer with a 3.6 GHz Intel Core i9 CPU and 64 GB of RAM. Roughly speaking, the computation time increases linearly as the number of notes in the input data increases and it also increases linearly as the number of iterations for Gibbs sampling for the Bayesian models increases.

From the results we see that incorporating modification models significantly increases the computation time and note pattern MMs require much more time than note value MMs and metrical MMs. Considering both the accuracy and computation time, the first-order and second-order metrical MMs with Bayesian extensions are the most useful models in practical applications. Incorporating modification models into these models would increase the accuracy but also significantly increase the computation time. The order of a model, the use of the Bayesian extension and the number of iterations for Gibbs sampling, and the use of modification models should be determined according to the given resource of the computation time.

In practical applications, we need to address longer note values and finer beat resolutions than those considered in this study. For example, if we consider the full bar length in 4/4 time and a beat resolution corresponding to one-third of a 16th note (to accommodate triplets), $N_b = 48$. The computation times required for the first-order note value MMs and metrical MMs are roughly proportional to N_b^2, and those for the second-order models are roughly proportional to N_b^3. However, the number of possible note patterns increases as 2^{N_b}, which makes the computation times of the note pattern MMs intractable, at least without refined searching techniques. This means that the metrical MMs also have better extendability.

6. Conclusion

We have studied the statistical description of the repetitive structure of musical notes using Bayesian score models and its application to rhythm transcription. The main results are summarized as follows.

Table 4
Computation times (sec) for transcribing the real data. For the Bayesian models, the number of iterations for Gibbs sampling was 100; and for PatMM1DB and PatMM1SDB, *W* was 1000. The acronyms for the models are explained in Table 2.

Model	Time	Model	Time	Model	Time
NoteMM0	< 0.1	NoteMM0B	0.13	NoteMM1	< 0.1
NoteMM1B	0.65	NoteMM1SB	111	NoteMM1DB	23
NoteMM1SDB	1.4×10^3	NoteMM2	< 0.1	NoteMM2B	5.0
MetMM0	< 0.1	MetMM0B	0.7	MetMM1	< 0.1
MetMM1B	0.7	MetMM1SB	111	MetMM1DB	413
MetMM1SDB	2.0×10^4	MetMM2	< 0.1	MetMM2B	4.8
PatMM0	1.3	PatMM0B	558	PatMM1	1.6
PatMM1B	656	PatMM1SB	2.4×10^4		
PatMM1DB	3.3×10^4	PatMM1SDB	9.6×10^4		

- The repetitive structure of musical rhythms is reflected in the sparseness of piece-specific score models, and the Dirichlet process describing the generative process of piece-specific score models from a generic score model can explain the distribution of the entropies of piece-specific models, particularly for the note value MM and metrical MM (Fig. 5).
- We have confirmed that the Bayesian score models capturing repetitions were indeed effective for significantly improving the transcription accuracy.
- Considering the transcription accuracy and the computational efficiency, the second-order Bayesian metrical MM was found to be the best among the tested models.
- Incorporating the note modification process can help capture approximate repetitions and therefore improve the accuracy but the effect was small and the computational cost significantly increased.

In general, these results provide yet another piece of evidence for the importance of musical language modelling for music transcription and demonstrate the effectiveness of utilizing the repetitive structure. Since repetitions are found in various musical elements including rhythms, pitch configurations, and chord progressions, the present approach can also be useful for polyphonic music transcription, chord recognition, and other automatic music recognition tasks.

As future work, it is important to extend the model and incorporate pitches and multiple voices for applications to more general types of music transcription. The present approach of inferring piece-specific models can be even more effective in such cases that the amount of noise is larger than the case of rhythm transcription studied here. Such a situation is common when MIDI sequences are obtained from audio signals [17,20]. Another interesting possibility is to combine the present Bayesian approach with deep generative models that can learn more complex structures than Markov models can learn.

CRediT authorship contribution statement

Eita Nakamura: Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Software, Writing - original draft. **Kazuyoshi Yoshii:** Data curation, Funding acquisition, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Rhythm transcription algorithm for the first-order Bayesian metrical Markov model (MetMM1B)

In this appendix, an explicit algorithm for rhythm transcription based on the first-order Bayesian metrical Markov model (MetMM1B) is presented with some details of computation procedure. We use the symbol $\mathbb{R}_+$ to represent the set of nonnegative real numbers and write $\boldsymbol{a} \in \mathbb{R}_+^{\boldsymbol{D}}$ to express that $\boldsymbol{a}$ is a D-dimensional vector with nonnegative elements (probability vectors are further assumed to be normalized).

A.1. Generative model

The generative process of the model is summarized as follows. The metrical Markov model (Section 2.2.2) generates the metrical positions $b_{0:N} = (b_0, \dots, b_N)$ as

$$P(b_0 = b) = \chi_{\mathrm{ini}\,b}, \quad P(b_n = b \,|\, b_{n-1} = b') = \chi_{b'b}. \tag{A.1}$$

Here, each metrical position b_n takes a value in $\{0, 1, \dots, N_b - 1\}$ (N_b is the bar length), and $\boldsymbol{\chi}_{\mathrm{ini}} = (\chi_{\mathrm{ini}\,b})_b \in \mathbb{R}_+^{N_b}$ and $\boldsymbol{\chi}_b = (\chi_{bb'})_{b'} \in \mathbb{R}_+^{N_b}$ are probability vectors. The performance model (Section 4.1) generates note durations $d_{1:N}$ ($d_n > 0$) of a performance MIDI signal as

$$P(d_n \,|\, b_{n-1} = b', b_n = b) = \phi_{b'bd_n} = \mathrm{Gauss}(d_n; [b', b]\,v, \sigma_t^2), \tag{A.2}$$

where

$$\mathrm{Gauss}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{A.3}$$

denotes a Gaussian distribution, $v > 0$ denotes the (inverse) tempo, $\sigma_t > 0$ denotes the amount of onset time deviations, and $[b', b]$ represents the note value as in Eq. (4). In the experimental setup of this study, parameters v and σ_t are constants specified prior to an application of the algorithm. The complete-data probability of the whole data sequence ($d_{1:N}$ and $b_{0:N}$) is given as

$$P(d_{1:N}, b_{0:N}) = \chi_{\mathrm{ini}\,b_0} \prod_{n=1}^{N} \chi_{b_{n-1}b_n} \phi_{b_{n-1}b_n d_n}. \tag{A.4}$$

The piece-specific model parameters $\boldsymbol{\chi}_{\mathrm{ini}} = (\chi_{\mathrm{ini}\,b})_b$ and $\boldsymbol{\chi}_b = (\chi_{bb'})_{b'}$ are generated from the generic model parameters $\bar{\boldsymbol{\chi}}_{\mathrm{ini}} \in \mathbb{R}_+^{N_b}$ and $\bar{\boldsymbol{\chi}}_b \in \mathbb{R}_+^{N_b}$ by a Dirichlet process:

$$P(\boldsymbol{\chi}_{\mathrm{ini}}) = \mathrm{Dir}(\boldsymbol{\chi}_{\mathrm{ini}}; \alpha_{\mathrm{ini}} \bar{\boldsymbol{\chi}}_{\mathrm{ini}}), \tag{A.5}$$

$$P(\boldsymbol{\chi}_b) = \mathrm{Dir}(\boldsymbol{\chi}_b; \alpha_\chi \bar{\boldsymbol{\chi}}_b), \tag{A.6}$$

where

$$\mathrm{Dir}(\boldsymbol{\chi}; \boldsymbol{\alpha}) = \Gamma(\alpha_1 + \cdots + \alpha_D) \prod_{i=1}^{D} \frac{\chi_i^{\alpha_i - 1}}{\Gamma(\alpha_i)} \tag{A.7}$$

denotes a Dirichlet distribution ($\Gamma(x)$ denotes the gamma function), and $\alpha_{\mathrm{ini}} > 0$ and $\alpha_\chi > 0$ are concentration parameters. These parameters are treated as constants to be specified prior to an application of the algorithm.

In the application of this Bayesian model for rhythm transcription, the generic model parameters $\bar{\boldsymbol{\chi}}_{\mathrm{ini}}$ and $\bar{\boldsymbol{\chi}}_b$ are pre-trained using musical score data (Section 4.2). The piece-specific model parameters $\boldsymbol{\chi}_{\mathrm{ini}}$ and $\boldsymbol{\chi}_b$ are treated as variables and estimated during a run of the algorithm, as described below.

A.2. Estimation of piece-specific model parameters

For rhythm transcription, we estimate metrical positions $b_{0:N}$ from a given performance MIDI signal with note durations $d_{1:N}$. In the Bayesian setting, the piece-specific model parameters are estimated first, as described in this subsection, and then the metrical positions are estimated to obtain the final rhythm transcription, as described in the next subsection. As formulated in Section 4.3, the former step is conducted by using the Gibbs sampling method [4] with the following steps of latent variable sampling and parameter sampling.

In the latent variable sampling step, model parameters $\boldsymbol{\chi}_{\mathrm{ini}}$ and $\boldsymbol{\chi}_b$ are fixed and $b_{0:N}$ are sampled by the forward filtering-backward sampling method. The forward variables $F_n(b_n) = P(d_{1:n}, b_n)$ are computed by the forward algorithm:

$$F_0(b_0) := \chi_{\mathrm{ini}\,b_0} \quad \text{(initialization)}, \tag{A.8}$$

$$F_n(b_n) = \sum_{b_{n-1}} \chi_{b_{n-1}b_n} \phi_{b_{n-1}b_n d_n} F_{n-1}(b_{n-1}). \tag{A.9}$$

The variables $b_{0:N}$ are then sampled by the backward sampling method as

$$P(b_N | d_{1:N}) = \frac{F_N(b_N)}{\sum_b F_N(b)}, \tag{A.10}$$

$$P(b_n | b_{n+1:N}, d_{1:N}) = \frac{\chi_{b_n b_{n+1}} \phi_{b_n b_{n+1} d_{n+1}} F_n(b_n)}{\sum_b \chi_{b b_{n+1}} \phi_{b b_{n+1} d_{n+1}} F_n(b)}. \tag{A.11}$$

In the parameter sampling step, metrical positions $b_{0:N}$ are fixed and model parameters $\boldsymbol{\chi}_{\mathrm{ini}}$ and $\boldsymbol{\chi}_b$ are sampled by using the probabilities

$$P(\boldsymbol{\chi}_{\mathrm{ini}}) = \mathrm{Dir}(\boldsymbol{\chi}_{\mathrm{ini}}; \alpha_{\mathrm{ini}} \bar{\boldsymbol{\chi}}_{\mathrm{ini}} + \mathbf{c}_{\mathrm{ini}}), \tag{A.12}$$

$$P(\boldsymbol{\chi}_{b'}) = \mathrm{Dir}(\boldsymbol{\chi}_{b'}; \alpha_\chi \bar{\boldsymbol{\chi}}_{b'} + \mathbf{c}^\chi_{b'}), \tag{A.13}$$

where the variables $\mathbf{c}_{\mathrm{ini}} = (c_{\mathrm{ini}\,b})_b$ and $\mathbf{c}^\chi_{b'} = (c^\chi_{b'b})_b$ count how many times metrical positions appear in $b_{0:N}$ and are defined as

$$c_{\mathrm{ini}\,b} = \delta_{b_0 b}, \quad c^\chi_{b'b} = \sum_{n=1}^{N} \delta_{b_{n-1}b'} \delta_{b_n b}. \tag{A.14}$$

The standard method for sampling a probability vector $\boldsymbol{\chi}$ from a Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\chi}; \boldsymbol{\alpha})$ can be used: after sampling a number y_i from the gamma distribution $\mathrm{Gam}(\alpha_i, 1)$ independently for each i, the vector $\boldsymbol{\chi}$ is obtained by normalization ($\chi_i = y_i / \sum_j y_j$).

For initialization, the model parameters $\boldsymbol{\chi}_{\mathrm{ini}}$ and $\boldsymbol{\chi}_b$ are set to $\bar{\boldsymbol{\chi}}_{\mathrm{ini}}$ and $\bar{\boldsymbol{\chi}}_b$, respectively. We then iterate the latent variable sampling step and the parameter sampling step iteratively for a predetermined number of times. At each iteration the evidence probability $P(d_{1:N}) = \sum_b F_N(b)$, calculated by Eq. (A.9) with the temporary values of $\boldsymbol{\chi}_{\mathrm{ini}}$ and $\boldsymbol{\chi}_b$, is memorized. After running all iterations, the parameterization that yields the maximum evidence probability is finally chosen and used for the subsequent step for estimating metrical positions.

A.3. Estimation of metrical positions

After piece-specific model parameters $\boldsymbol{\chi}_{\rm ini}$ and $\boldsymbol{\chi}_b$ are estimated, the metrical positions $\hat{b}_{0:N}$ for the final rhythm transcription result are estimated by maximizing the probability $P(b_{0:N}|d_{1:N})$ (Section 4.2). By the Bayes formula, $P(b_{0:N}|d_{1:N}) \propto P(d_{1:N}, b_{0:N})$ and this is equivalent to maximizing the probability in Eq. (A.4) w.r.t. $b_{0:N}$. To solve this problem, we can apply the Viterbi algorithm [21] as follows. The Viterbi variables $V_n(b_n)$ and backtracking indices $U_n(b_n)$ are computed as

$$V_0(b_0) := \chi_{{\rm ini}\, b_0} \quad \text{(initialization)}, \tag{A.15}$$

$$V_n(b_n) = \max_{b_{n-1}} [\chi_{b_{n-1} b_n} \phi_{b_{n-1} b_n d_n} V_{n-1}(b_{n-1})], \tag{A.16}$$

$$U_n(b_n) = \operatorname*{argmax}_{b_{n-1}} \left[\chi_{b_{n-1} b_n} \phi_{b_{n-1} b_n d_n} V_{n-1}(b_{n-1})\right]. \tag{A.17}$$

The most probable sequence $\hat{b}_{0:N}$ is then obtained by backtracking:

$$\hat{b}_N = \operatorname*{argmax}_{b_N} V_N(b_N), \tag{A.18}$$

$$\hat{b}_n = U_{n+1}(\hat{b}_{n+1}), \tag{A.19}$$

which completes our rhythm transcription algorithm.

In the actual implementation most computations involving probabilities of data sequences explained in this appendix are performed in the logarithmic domain to avoid overflow. See the source code in the Supplemental Material for implementation details. Rhythm transcription algorithms for other models studied in this paper can be derived similarly. The details are also given in the Supplemental Material.

Appendix B. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.ins.2021.04.100.

References

[1] E. Benetos, S. Dixon, Z. Duan, S. Ewert, Automatic music transcription: An overview, IEEE Signal Process. Mag. 36 (1) (2019) 20–30.
[2] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, A. Klapuri, Automatic music transcription: Challenges and future directions, J. Intell. Inf. Syst. 41 (3) (2013) 407–434.
[3] E. Benetos, T. Weyde, An efficient temporally-constrained probabilistic model for multiple-instrument music transcription, Proc. ISMIR (2015) 701–707.
[4] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[5] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[6] A.T. Cemgil, P. Desain, B. Kappen, Rhythm quantization for transcription, Comp. Mus. J. 24 (2) (2000) 60–76.
[7] P. Desain, H. Honing, The quantization of musical time: A connectionist approach, Comp. Mus. J. 13 (3) (1989) 56–66.
[8] S. Fine, Y. Singer, N. Tishby, The hierarchical hidden Markov model: Analysis and applications, Mach. Learn. 32 (1) (1998) 41–62.
[9] M. Goto, H. Hashiguchi, T. Nishimura, R. Oka, RWC music database: Popular, classical and jazz music databases, Proc. ISMIR (2002) 287–288.
[10] M. Hamanaka, M. Goto, H. Asoh, N. Otsu, A learning-based quantization: Unsupervised estimation of the model parameters, Proc. ICMC (2003) 369–372.
[11] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, D. Eck, Onsets and frames: Dual-objective piano transcription, Proc. ISMIR (2018) 50–57.
[12] D. Huron, Sweet Anticipation: Music and the Psychology of Expectation, The MIT Press, 2006.
[13] N. Jacoby, J.H. McDermott, Integer ratio priors on musical rhythm revealed cross-culturally by iterated reproduction, Curr. Biol. 27 (3) (2017) 359–370.
[14] M.I. Jordan, Dirichlet processes, Chinese restaurant processes and all that, in: Tutorial Presentation at the NIPS Conference, 2005.
[15] H. Longuet-Higgins, Mental Processes: Studies in Cognitive Science, MIT Press, 1987.
[16] J.-F. Mari, J.-P. Haton, A. Kriouile, Automatic word recognition based on second-order hidden Markov models, IEEE Trans. Speech Audio Process. 5 (1) (1997) 22–25.
[17] E. Nakamura, E. Benetos, K. Yoshii, S. Dixon, Towards complete polyphonic music transcription: Integrating multi-pitch detection and rhythm quantization, Proc. ICASSP (2018) 101–105.
[18] E. Nakamura, K. Itoyama, K. Yoshii, Rhythm transcription of MIDI performances based on hierarchical Bayesian modelling of repetition and modification of musical note pattern, Proc. EUSIPCO (2016) 1946–1950.
[19] E. Nakamura, K. Yoshii, S. Sagayama, Rhythm transcription of polyphonic piano music based on merged-output HMM for multiple voices, IEEE/ACM TASLP 25 (4) (2017) 794–806.
[20] R. Nishikimi, E. Nakamura, K. Itoyama, K. Yoshii, Musical note estimation for F0 trajectories of singing voices based on a Bayesian semi-beat-synchronous HMM, Proc. ISMIR (2016) 461–467.
[21] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–286.
[22] S. Raczynski, E. Vincent, S. Sagayama, Dynamic Bayesian networks for symbolic polyphonic pitch modeling, IEEE/ACM TASLP 21 (9) (2013) 1830–1840.
[23] C. Raphael, A hybrid graphical model for rhythmic parsing, Artif. Intell. 137 (2002) 217–238.
[24] C. Raphael, A graphical model for recognizing sung melodies, Proc. ISMIR (2005) 658–663.
[25] A. Ravignani, B. Thompson, T. Grossi, T. Delgado, S. Kirby, Evolving building blocks of rhythm: how human cognition creates music via cultural transmission, Ann. N. Y. Acad. Sci. 1423 (2018) 176–187.
[26] P.E. Savage, S. Brown, E. Sakai, T.E. Currie, Statistical universals reveal the structures and functions of human music, PNAS 112 (29) (2015) 8987–8992.

[27] R. Schramm, A. McLeod, M. Steedman, E. Benetos, Multi-pitch detection and voice assignment for a Cappella recordings of multiple singers, Proc. ISMIR (2017) 552–559.
[28] S. Sigtia, E. Benetos, S. Dixon, An end-to-end neural network for polyphonic piano music transcription, IEEE/ACM TASLP 24 (5) (2016) 927–939.
[29] G. Siorosa, M.E.P. Davies, C. Guedes, A generative model for the characterization of musical rhythms, J. New Music Res. 47 (2) (2018) 114–128.
[30] L.M. Smith, H. Honing, Time-frequency representation of musical rhythm by continuous wavelets, J. Math. Music 2 (2) (2008) 81–97.
[31] H. Takeda, T. Otsuki, N. Saito, M. Nakai, H. Shimodaira, S. Sagayama, Hidden Markov model for automatic transcription of MIDI signals, Proc. MMSP (2002) 428–431.
[32] D. Temperley, D. Sleator, Modeling meter and harmony: A preference-rule approach, Comp. Mus. J. 23 (1) (1999) 10–27.
[33] Y.-T. Wu, B. Chen, L. Su, Polyphonic music transcription with semantic segmentation, Proc. ICASSP (2019) 166–170.
[34] A. Ycart, E. Benetos, A study on LSTM networks for polyphonic music sequence modelling, Proc. ISMIR (2017) 421–427.