Rhythm Transcription of MIDI Performances Based on Hierarchical Bayesian Modelling of Repetition and Modification of Musical Note Patterns

Eita Nakamura, Katsutoshi Itoyama, Kazuyoshi Yoshii
Kyoto University, Kyoto 606-8501, Japan

***Abstract*—This paper presents a method of rhythm transcription (i.e., automatic recognition of note values in music performance signals) based on a Bayesian music language model that describes the repetitive structure of musical notes. Conventionally, music language models for music transcription are trained with a dataset of musical pieces. Because typical musical pieces have repetitions consisting of a limited number of note patterns, better models fitting individual pieces could be obtained by inducing compact grammars. The main challenges are inducing appropriate grammar for a score that is observed indirectly through a performance and capturing incomplete repetitions, which can be represented as repetitions with modifications. We propose a hierarchical Bayesian model in which the generation of a language model is described with a Dirichlet process and the production of musical notes is described with a hierarchical hidden Markov model (HMM) that incorporates the process of modifying note patterns. We derive an efficient algorithm based on Gibbs sampling for simultaneously inferring from a performance signal the score and the individual language model behind it. Evaluations showed that the proposed model outperformed previously studied HMM-based models.**

I. Introduction

Music transcription is a fundamental problem in music information processing, requiring the extraction of pitch and rhythm information from music audio signals. Many studies on acoustic modelling of musical sounds have been carried out for extracting pitch information [1,2]. Rhythm transcription (or quantisation), on the other hand, has been addressed in the aim of recognising score-written lengths (or note values) of musical notes [3–8]. Using prior knowledge on music scores is crucial for music/rhythm transcription, like speech recognition [9], and machine-learning techniques have been studied in efforts to construct music 'language' models. Hidden Markov models (HMMs) have been widely used to learn a 'generic' language model with a dataset of musical pieces [5–7, 10, 11].

Since musical pieces typically have repetitive structure consisting of a limited number of note patterns, more accurate language models could be obtained by inferring compact grammars fitting individual pieces. Finding repeated patterns in music is a topic of computational music structure analysis, and methods based on similarity matrices [12, 13], data compression [14], or Markov Oracle [15] have been studied. Once the repetitive structure of a piece is known, note-pattern models [6,8] could be used to learn the piece's grammar. Conversely, if used note patterns are given, we could infer the repetitive structure more accurately. To solve this chicken-and-egg problem, we need a framework for simultaneously inducing grammar and analysing music data.

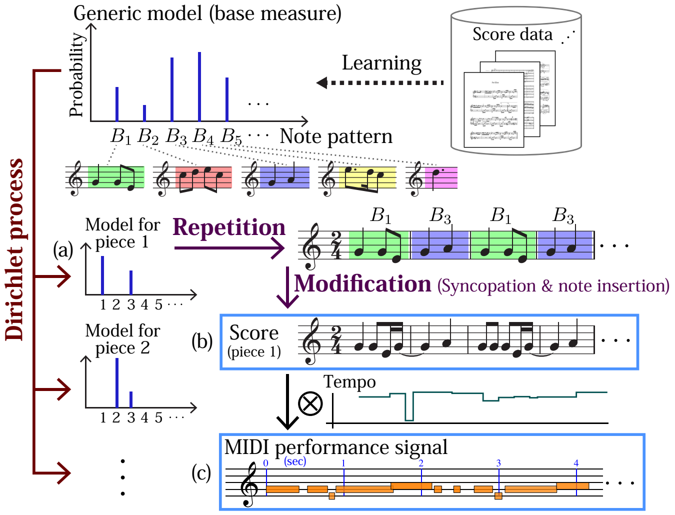

Fig. 1. Overview of the proposed model describing the generating process of music score with incomplete repetitions and the process of music performance.

This inter-relation between grammar induction and data analysis has been gathering attention in the field of natural language processing (NLP), and Bayesian methods have been extensively developed [16, 17]. A method for simultaneously learning word vocabulary and segmenting sentences into words [18] is analogous to our problem (characters vs. notes and words vs. note patterns). Recent studies have focused on the problem of modification to repeated patterns (words) [19, 20]. Modelling modification is necessary to identify incomplete repetitions, which are common in music [21].

In this paper, we propose a hierarchical Bayesian model for simultaneously inferring from a performance signal the score and the individual language model behind it (Fig. 1). (a) First a language model based on a hierarchical HMM of note patterns is generated, then (b) musical notes in the score are generated by the language model, and finally (c) a performance signal is generated based on the score. Compact grammar models are represented as outputs of a Dirichlet process [22] with a small concentration parameter. Incomplete repetitions are represented as repetitions with modifications and described with a probabilistic model for modification of note patterns. The process of music performance is described with a tempo fluctuation model used in studies on score-performance matching [23, 24]. In this study we focus on the rhythmic aspect and

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

confine ourselves to monophonic music.

The main contribution of this study is its treatment of incomplete repetitions of note patterns. Bayesian grammar induction for music has been addressed in previous studies using probabilistic context-free grammar (PCFG) models using note patterns without modification [8] or note-wise productions without note patterns [25, 26]. The problem of simultaneously inducing grammar and segmenting text data with modification has not been addressed in NLP. We evaluated the model by comparing its accuracy of rhythm transcription with that of previously studied HMM-based models.

II. Related Work

Previously studied statistical music language models for rhythm/music transcription are reviewed in this section (Fig. 2). The output of a language model is a sequence of notes $x_{1:N} = x_1 \cdots x_N$ where N denotes the number of notes. (Similar notations will be used throughout this paper.) Since we are focusing on the rhythmic aspect, x_n represents the note value of the n-th note. The note value is defined as the score-written note length relative to a whole note (a quarter note has an $x = 1/4$, a dotted half note has an $x = 3/4$, etc.).

A. Note-Level Markov Model

Markov models of musical notes have been proposed in early studies [6]. In the first-order model the probability of $x_{1:N}$ is given as a product of transition probabilities:

$$P(x_{1:N}) = \prod_{n=1}^{N} P(x_n|x_{n-1}), \tag{1}$$

where, with an abuse of notation, $P(x_1|x_0) \equiv P(x_1)$ signifies the initial probability. We can extend it to a p-th-order model by replacing $P(x_n|x_{n-1})$ with $P(x_n|x_{n-1}, \ldots, x_{n-p+1})$.

A problem with this note-level model is that certain logical constraints on the sequence of note values cannot be incorporated in the model. For example, triplet notes must appear in triplets or in pairs with a double triplet note. This constraint cannot be described with a note-level Markov model of any order. Nor can metrical structure be incorporated in the model.

B. Note-Pattern Model

In most music, including classical and popular music, musical notes have metrical structure, and note-pattern models incorporating metrical structure have been proposed in previous studies [6, 8]. In the note-pattern Markov model [6], note patterns with a fixed time span (e.g., a bar) are considered as the state space. We notate a note pattern (a string of notes) with $B_k = z_{k,1} \cdots z_{k,L}$ $(k = 1, \ldots, K)$, where k indexes the set of K note patterns and $z_{k,\ell}$ $(\ell = 1, \ldots, L)$ denotes the ℓ-th note in note pattern k. The probability of the sequence of patterns $w_1 \cdots w_I$ $(w_i \in \{B_k\}_{k=1}^{K})$ is given as a product of transition probabilities $\pi_{kk'} = P(w_i = B_{k'}|w_{i-1} = B_k)$ and an initial probability $\pi_k^0 = P(w_1 = B_k)$.

The sequence of notes, denoted by $z_{1:M}$, is obtained by concatenating the generated note patterns $w_{1:I}$, and its probability is described with a hierarchical Markov model [27]: The upper level describes note patterns and the lower level describes notes. Each note z_m is indicated by the pair (k, ℓ), and the transition probability is given as

$$P\big(z_m{=}(k',\ell')|z_{m-1}{=}(k,\ell)\big) = \delta_{\ell L}\pi_{kk'}\delta_{\ell' 1} + \delta_{kk'}\delta_{\ell'(\ell+1)},$$

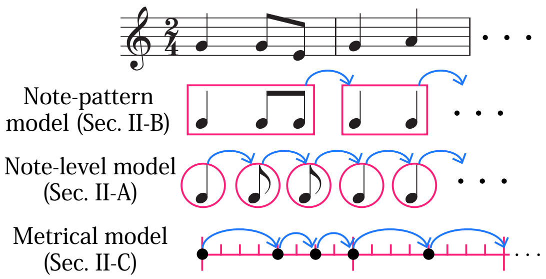
Note-pattern model (Sec. II-B)
Note-level model (Sec. II-A)
Metrical model (Sec. II-C)

Fig. 2. Three different representations of a music score in previously proposed models [5–8].

where δ denotes Kronecker's delta. A similar model based on PCFG has also been proposed [8].

An additional advantage of the note-pattern model is that it can incorporate the logical constraints for triplet notes etc. On the other hand, a problem of this model is the treatment of syncopations. Since a syncopated note lies across a bar boundary, which is typically a boundary of note patterns, syncopated notes cannot be described with the above note-pattern model.

C. Metrical Model

Another type of model that incorporates the metrical structure is the metrical (grid) model [5, 7], in which musical notes are represented by their onset beat positions in a bar. Let s_n denote the beat position of the n-th note and let G denote the time span of a bar. The generation of musical notes is described with a Markov model on a grid of beat positions, and thus the probability of the sequence $s_{1:N+1}$ is given as a product of transition probabilities $P(s_n|s_{n-1})$ as in Eq. (1). The note value of the n-th note is given as

$$x_n = \begin{cases} s_{n+1} - s_n, & (s_{n+1} > s_n); \\ G + s_{n+1} - s_n, & (s_{n+1} \le s_n), \end{cases} \tag{2}$$

i.e., s_{n+1} is interpreted as a beat position in the next bar if it is smaller than or equal to s_n.

Although the original metrical model limits note values to the length of a bar, it is possible to extend the model to describe larger note values. This can be done by introducing a discrete variable $j_n = 0, 1, \ldots, J{-}1$ for each note n (for some positive integer J), which describes how may bar lines are passed between note onset $n{-}1$ and n. In other words, the note value of the n-th note is now given as $x_n = j_{n+1}G{+}s_{n+1}{-}s_n$. By taking the pair (s_n, j_n) as a state variable, we can extend the model to accommodate note values up to JG.

The metrical model is advantageous in the treatment of syncopations. In Eq. (2) the n-th note is syncopated if $s_{n+1} \neq 0$ and $s_{n+1} \le s_n$. On the other hand, its disadvantage is the difficulty of modifying or extending the model. For example, it is necessary to construct different models for different metres.

III. Proposed Model

The proposed model consists of two components; a language model and a performance model. Details of these models and an inference algorithm are explained in this section.

A. Language Model

To describe incomplete repetitions of note patterns, we extend the model in Sec. II-B in two directions; integration of modification of note patterns and Bayesian extension based on the Dirichlet process to describe compact grammar.

1) Modification of Note Patterns: The output of our language model, denoted by $x_{1:N}$, is obtained by modifying the note sequence generated by the note-pattern model in Sec. II-B, denoted by $z_{1:M}$ in the following (Fig. 3). We consider note insertions and syncopations, which are typical modifications of note patterns in music practice. The total note values of the corresponding notes are unchanged after these modifications.

Note insertions are represented by divided notes and thus described with a probability of the form $P(y_1 \cdots y_Q|z_m)$ where $y_1, \ldots, y_Q$ are notes produced from z_m by insertions, which satisfy $y_1 + \cdots + y_Q = z_m$. We use the symbol Q as the number of notes in an insertion pattern and $q(= 1, \ldots, Q)$ as an index of a note in that pattern. Let $C_h = y_1 \cdots y_Q$ $(h = 1, \ldots, H)$ denote an insertion pattern (including the unchanged case) and $\phi_{(k\ell)h}$ denote the probability $P\big(C_h|z_m = (k, \ell)\big)$. The sequence of notes after note insertions will be denoted by $y_{1:N}$, which is specified by an applied insertion pattern h_m for each z_m. This process of inserting notes can be integrated in the basic model as yet another lower-level Markov model.

Syncopations can be regarded as simultaneous deviations of the last note of a note pattern and the first note of the next pattern (Fig. 3). Let $x_{1:N}$ represent a score with syncopations, obtained from $y_{1:N}$. Syncopations can be parameterised with the degree of syncopation s, so that notes are modified as

$$y_n \to x_n = y_n + s, \quad y_{n+1} \to x_{n+1} = y_{n+1} - s, \tag{3}$$

where x_{n+1} must be the first note of a note pattern[1]. The parameter s can take either positive or negative values. A positive (negative) s represents a suspension (anticipation). Syncopations can be integrated in the model by extending the state space of the basic model, w_i, to a pair (w_i, s_i). Using the notation $\theta_s = P(s)$, the transition probability is extended as $P(w_i = B_{k'}, s_i|w_{i-1} = B_k, s_{i-1}) = \pi_{kk'}\theta_{s_i}$, assuming independence of the probabilities of the component variables. Extension of the initial probability is similar.

In summary, a score $x_{1:N}$ generated by the language model is specified by stochastic variables $w_{1:I}$, $s_{1:I}$, and $h_{1:M}$. This means that each note x_n is specified with a set of indices (k, ℓ, h, q, s) and the language model can be described as a Markov model with the following transition probability:

$$\begin{aligned} &P\big(x_n = (k', \ell', h', q', s')|x_{n-1} = (k, \ell, h, q, s)\big) \\ &\quad = \delta_{qQ}\phi_{(k'\ell')h'}\delta_{q'1}\big[\delta_{\ell L}\pi_{kk'}\theta_{s'}\delta_{\ell'1} + \delta_{kk'}\delta_{ss'}\delta_{(\ell+1)\ell'}\big] \\ &\qquad + \delta_{hh'}\delta_{(q+1)q'}\delta_{kk'}\delta_{\ell\ell'}\delta_{ss'}. \end{aligned} \tag{4}$$

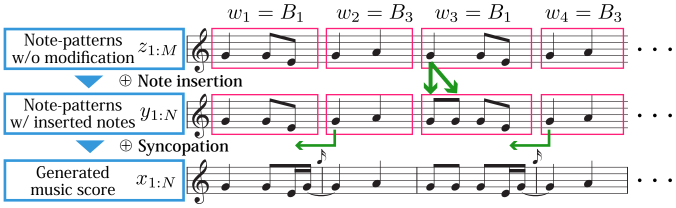
$w_1 = B_1$
$w_2 = B_3$
$w_3 = B_1$
$w_4 = B_3$
Note-patterns w/o modification $z_{1:M}$
⊕ Note insertion
Note-patterns w/ inserted notes $y_{1:N}$
⊕ Syncopation
Generated music score $x_{1:N}$

Fig. 3. Model for modification of note patterns.

2) Dirichlet Prior: The parameters of the language model, $\boldsymbol{\pi}_k = (\pi_{kk'})_{k'}$, $\boldsymbol{\pi}^0 = (\pi^0_k)_k$, $\boldsymbol{\phi}_{(k\ell)} = (\phi_{(k\ell)h})_h$, and $\boldsymbol{\theta} = (\theta_s)_s$, characterise the statistical properties of music. Because used note patterns and types of modifications vary among musical pieces, different values of the parameters are considered for individual pieces. In the Bayesian framework, these parameters are regarded as being generated from prior models.

The Dirichlet process can serve as a prior model that can control the sparseness of the generated distributions. In the case of finite distributions, the Dirichlet process for a discrete distribution $\boldsymbol{\pi}$ is described with a base distribution $\boldsymbol{\omega}$ and a concentration parameter α as follows:

$$\boldsymbol{\pi} \sim \mathrm{DP}(\alpha, \boldsymbol{\omega}) = \mathrm{Dir}(\alpha\boldsymbol{\omega}), \tag{5}$$

where Dir($\cdot$) denotes the Dirichlet distribution. Distributions obtained in this way satisfy $\mathbb{E}[\boldsymbol{\pi}] = \boldsymbol{\omega}$, and for small α most components of $\boldsymbol{\pi}$ tend to be zero. We put such Dirichlet priors for $\boldsymbol{\pi}_k$ and $\boldsymbol{\pi}^0$:

$$\boldsymbol{\pi}_k \sim \mathrm{Dir}(\alpha\boldsymbol{\omega}_k), \quad \boldsymbol{\pi}^0 \sim \mathrm{Dir}(\alpha\boldsymbol{\omega}^0). \tag{6}$$

The hyperparameters $\boldsymbol{\omega}_k$ and $\boldsymbol{\omega}^0$, when learned from a database, are interpreted as a generic model (Fig. 1), or they can be set to uniform distributions in an unsupervised learning setting. When concentration parameter α is small, a compact grammar is induced; i.e., a small number of note patterns will be used for each piece. We also put Dirichlet priors for $\boldsymbol{\phi}_{(k\ell)}$ and $\boldsymbol{\theta}$:

$$\boldsymbol{\phi}_{(k\ell)} \sim \mathrm{Dir}(\boldsymbol{\xi}), \quad \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\lambda}). \tag{7}$$

B. Performance Model

In the setup for rhythm transcription, the score is observed indirectly through a performance. The performance signal is specified with a sequence of onset times $t_{1:N+1}$, or equivalently, a sequence of inter-onset intervals $d_{1:N}$ where $d_n = t_{n+1} - t_n$ $(n = 1, \ldots, N)$. A performance model gives the probability $P(d_{1:N}|x_{1:N})$ of a performance given a score.

The performance model we use is based on a linear dynamical system proposed for score-performance matching [23,24]. The model describes two sources of temporal fluctuations in music performance. One is the fluctuation in onset time due to human motor noise and the other is the variation in tempos. Tempo is considered as a latent variable v_n, which represents the ratio d_n/x_n up to the noise of onset time, and its variation is described with a Markov process. Assuming that the sources for tempo variation and the noise in onset time are both Gaussian, the performance model is given as

$$v_n|v_{n-1} \sim N(v_{n-1}, \sigma_v^2), \quad d_n|v_n, x_n \sim N(v_n x_n, \sigma_t^2), \tag{8}$$

where σ_v (σ_t) is the standard deviation for tempo variation (motor noise). The complete-data probability for the performance model is given as

$$P(d_{1:N}, v_{1:N}|x_{1:N}) = \prod_{n=1}^{N} P(d_n|v_n, x_n)P(v_n|v_{n-1}), \tag{9}$$

[1] The variable s introduced here has no relations with s_n used in Sec. II-C.

where $P(v_1|v_0) \equiv P(v_1)$ signifies the initial probability for tempo. In practice, values of the tempo variable is discretised in a range typically used for music practice to enable inference. By combining Eqs. (4) and (9), the proposed model is a hierarchical HMM [27] with the latent variable $Z_n \equiv (x_n, v_n)$.

C. Inference

Our goal is to simultaneously infer the latent variables $Z = Z_{1:N}$ and the parameters of the language model, $\Theta = (\boldsymbol{\pi}_k, \boldsymbol{\pi}^0, \boldsymbol{\phi}_z, \boldsymbol{\theta})$, given the observed performance signal $D = d_{1:N}$ and the hyperparameters $\Lambda = (\boldsymbol{\omega}_k, \boldsymbol{\omega}^0, \alpha, \boldsymbol{\xi}, \boldsymbol{\lambda})$. In the model-learning step, the parameters Θ are estimated by maximising the posterior $P(\Theta|D,\Lambda)$. In the transcription step, the latent variables are estimated by maximising the posterior $P(Z|D,\Theta) \propto P(Z,D|\Theta)$, which can be done by the standard Viterbi algorithm. Since direct maximisation of $P(\Theta|D,\Lambda)$ is difficult, we use a Gibbs sampling method that yields asymptotically exact inference. In this method, samples are drawn from the joint distribution $P(Z,\Theta|D,\Lambda)$, from which samples from the distribution $P(\Theta|D,\Lambda)$ can be obtained instantly.

The Gibbs sampling method is based on alternating samplings of the parameters from the probabilities $P(\Theta|Z,D,\Lambda)$ and the latent variables from the probabilities $P(Z|\Theta,D,\Lambda)$. In the former sampling, model parameters are sampled from posterior Dirichlet distributions. For example, the transition probability $\boldsymbol{\pi}_k$ is sampled as

$$\boldsymbol{\pi}_k|Z,\Lambda \sim \mathrm{Dir}\big(\alpha\boldsymbol{\omega}_k + \boldsymbol{f}_k(Z)\big), \tag{10}$$

where $f_{kk'}(Z)$ is the number of times that transition $B_k \to B_{k'}$ appears in $x_{1:N}$. Other parameters can be sampled similarly.

We can use the forward filtering-backward sampling method to draw samples from $P(Z|\Theta,D,\Lambda)$. After computing the forward variables $\alpha_n(Z_n) = P(Z_n, d_{1:n}|\Theta)$ by the forward algorithm, the latent variables are sampled iteratively as

$$P(Z_n|Z_{n+1:N},D,\Theta) \propto \pi_{Z_nZ_{n+1}}P(d_{n+1}|Z_n,Z_{n+1})\alpha_n(Z_n)$$

with an initial draw of Z_N from $P(Z_N|d_{1:N},\Theta) \propto \alpha_N(Z_N)$. Note that the latent variables $x_{1:N}$ and $v_{1:N}$ are highly correlated in $P(Z|D,\Lambda)$ and should be sampled jointly.

Because the number of states of the language models can be $\mathcal{O}(10^4)$ and the product state space with the tempo variable has even more states, the computational cost of the forward algorithm can be impractical. This can be understood from the quadratic time complexity of the forward algorithm: With N_s states the time complexity of the forward algorithm is $\mathcal{O}(NN_\mathrm{s}^2)$. To solve this problem, we can use particle filtering for the approximate calculation of the forward variables. With N_p particles the time complexity is reduced to $\mathcal{O}(NN_\mathrm{s}N_\mathrm{p})$.

IV. Evaluation

A. Setup

We evaluated the proposed model by comparing its accuracy of rhythm transcription with that of previously studied models based on HMMs. A database of MIDI performances of 30 Japanese popular songs by various artists was prepared by the authors (the durations of the pieces ranged from about 15 sec to 50 sec). All normal, dotted, and triplet note values ranging from the whole note to the 16th note were used as candidate note values. For the proposed model, all note patterns of half-note length consisting of these candidate note values and the following note patterns of whole-note lengths were used: (1), (3/4, 1/4), and (1/4, 3/4). All possible pairs of those note values were used as note insertion patterns. The degree of syncopation was also taken from the candidate note values and their negative values and zero.

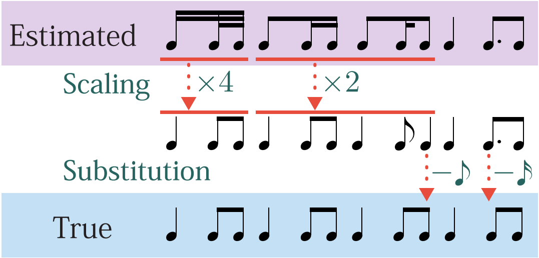

Fig. 4. Example of scaling operations and substitutions to recover the correct transcription from an estimated result.

We tested the model in three different learning conditions, supervised, semi-supervised, and unsupervised, and also in two cases of with or without modifications to note patterns (total of six cases). The supervised learning condition without modifications is equivalent to the original note pattern model in [6], for which $(\boldsymbol{\pi}_k)_k$ and $\boldsymbol{\pi}^0$ were trained with the melodies of 100 songs in the RWC popular music database [28]. For semi-supervised learning $(\boldsymbol{\omega}_k)_k$, $\boldsymbol{\omega}^0$, and $\boldsymbol{\lambda}$ were trained with the same dataset. For unsupervised learning, all $\boldsymbol{\omega}_k$ and $\boldsymbol{\omega}^0$ were set to uniform distributions and $\lambda_s = 10$ if $s = 0$ and 0.05 otherwise. Other hyperparameters were set as $\alpha = 700$, $\xi_h = 0.1$ for the non-insertion case and 0.01 otherwise, $\sigma_t = 0.02$ s, and $\sigma_v = 0.06$ seconds per quarter note [24].

For comparison, we implemented the note-level HMM using note bigrams [6] and the metrical HMM [5]. These models were also trained with the RWC database, and we set $J = 2$ for the metrical model. The performance model was the same as that for the proposed model.

We used as an evaluation measure the rhythm correction ratio, i.e., the ratio of the smallest number of edit operations needed to correct the estimated result to the number of notes in the data. In addition to note-wise correction (substitution), the scaling operation applied for a subsequence of note values was included (Fig. 4). This is because there is arbitrariness in choosing the unit of note values: For example, a quarter note played in a tempo of 60 BPM has the same duration as a half note played in a tempo of 120 BPM. Although the details must be omitted for the lack of space, the smallest number of necessary edit operations N_e can be calculated by a dynamic programming similar to that used in computation of the Levenshtein distance. The rhythm correction ratio $\mathscr{R}$ is then given as $\mathscr{R} = N_\mathrm{e}/N$.

B. Results

Results are shown in TABLE I. For the proposed model, incorporation of the modification model of note patterns im-

TABLE I
AVERAGE RHYTHM CORRECTION RATE $\mathscr{R}$ WITH STANDARD ERROR. LOWER IS BETTER.

Model	Learning	Modification	$\mathscr{R}$ [%]
1 Proposed	Unsupervised	✓	12.8 ± 1.3
2	Unsupervised		16.5 ± 1.8
3	Semi-supervised	✓	$\mathbf{6.6 \pm 1.0}$
4	Semi-supervised		17.7 ± 2.2
5	Supervised	✓	7.8 ± 1.2
6	Supervised		14.7 ± 1.8
7 Note-level HMM [6]	Supervised		7.9 ± 1.4
8 Metrical HMM [5]	Supervised		7.3 ± 1.3

proved the average rhythm correction rate in all learning conditions. By comparing the results for supervised learning and semi-supervised learning with the modification model, we see that the grammar induction for individual pieces indeed works effectively. In the best case, i.e., the semi-supervised learning condition with the modification model, the result outperformed the previously studied models. With the modification model, the unsupervised case was 5 points lower than the supervised case. An example result (Fig. 5) shows that the proposed model succeeded to capture the incomplete repetitive structure with syncopations and note insertions.

There is still room for improving the performance of the proposed model. One direction is removing the arbitrariness in choosing the fixed length of note patterns by introducing variable-length note-pattern model, which can be done with the use of hierarchical Dirichlet process. Using pitch information and modelling hierarchical repetitions would also be effective.

V. CONCLUSION

We have developed a framework for simultaneously inferring the score and the individual language model behind it from a performance signal. The proposed model has succeeded to learn compact grammar and segment a piece into representative note patterns. We plan to extend the model to an infinite vocabulary model, so that the representative note patterns of variable length can be automatically inferred. Another direction is an extension for polyphonic music, for which introducing voice structure is a key issue. The presented formulation of simultaneous grammar induction and data analysis would be effective for realistic music transcription from audio signals and also for NLP.

ACKNOWLEDGEMENTS

The authors thank Daichi Mochihashi and Ryo Nishikimi for useful discussions. This work was partly supported by JST OngaCREST Project, JSPS KAKENHI 24220006, 26700020, 26280089, 15K16054, 16H01744 and 16J05486, and Kayamori Foundation. EN is supported by the JSPS research fellowship program (PD).

REFERENCES

[1] A. Klapuri and M. Davy (eds.), *Signal Processing Methods for Music Transcription*, New York: Springer, 2006.

[2] E. Benetos *et al.*, "Automatic Music Transcription: Challenges and Future Directions," *J. Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

[3] A. T. Cemgil *et al.*, "Rhythm Quantization for Transcription," *Comp. Music J.*, vol. 24, no. 2, pp. 60–76, 2000.

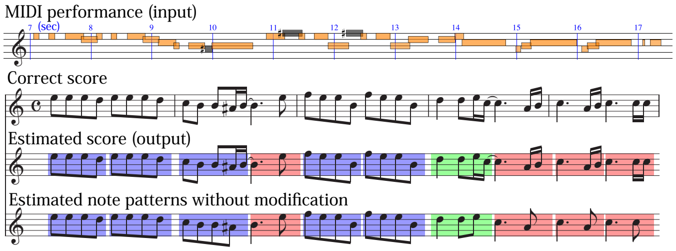

Fig. 5. Example result of rhythm transcription by the proposed model (semi-supervised with modifications modelled). The third low shows the estimated result and the fourth low shows the result after removing modifications.

[4] D. Temperley, *The Cognition of Basic Musical Structures*, The MIT Press, 2001.

[5] C. Raphael, "Automated Rhythm Transcription," *Proc. ISMIR*, pp. 99–107, 2001.

[6] H. Takeda *et al.*, "Hidden Markov Model for Automatic Transcription of MIDI Signals," *Proc. MMSP*, pp. 428–431, 2002.

[7] M. Hamanaka *et al.*, "A Learning-Based Quantization: Unsupervised Estimation of the Model Parameters," *Proc. ICMC*, pp. 369–372, 2003.

[8] M. Tsuchiya *et al.*, "Probabilistic Model of Two-Dimensional Rhythm Tree Structure Representation for Automatic Transcription of Polyphonic MIDI Signals," *Proc. APSIPA*, paper id 14002890, pp. 1–6, 2013.

[9] S. Levinson *et al.*, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition," *The Bell Sys. Tech. J.*, vol. 62, no. 4, pp. 1035–1074, 1983.

[10] C. Ames, "The Markov Process as a Compositional Model: A Survey and Tutorial,' *Leonardo*, vol. 22, no. 2, pp. 175–187, 1989.

[11] F. Pachet *et al.*, "Finite-Length Markov Processes with Constraints," *Proc. IJCAI*, pp. 635–642, 2011.

[12] M. Cooper *et al.*, "Automatic Music Summarization via Similarity Analysis," *Proc. ISMIR*, pp. 81–85, 2002.

[13] J. Paulus *et al.*, "State of the Art Report: Audio-Based Music Structure Analysis," *Proc. ISMIR*, pp. 625–636, 2010.

[14] D. Meredith *et al.*, "Algorithms for Discovering Repeated Patterns in Multidimensional Representations of Polyphonic Music," *J. New Music Res.*, vol. 31, no. 4, pp. 321–345, 2002.

[15] C. Wang *et al.*, "Music Pattern Discovery with Variable Markov Oracle: A Unified Approach to Symbolic and Audio Representations," *Proc. ISMIR*, pp. 176–182, 2015.

[16] M. Johnson *et al.*, "Bayesian Inference for PCFGs via Markov Chain Monte Carlo," *Proc. HLT-NAACL*, pp. 139–146, 2007.

[17] H. Shindo *et al.*, "Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing," *Proc. ACL*, pp. 440–448, 2012.

[18] D. Mochihashi *et al.*, "Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling," *Proc. ACL-IJCNPL*, pp. 100–108, 2009.

[19] T. Nakamura *et al.*, "Multimodal Concept and Word Learning Using Phoneme Sequences with Errors," *Proc. IROS*, pp. 157–162, 2013.

[20] Y. Yang *et al.*, "A Log-Linear Model for Unsupervised Text Normalization," *Proc. EMNLP*, pp. 61–72, 2013.

[21] L. Stein, *Structure & Style: The Study and Analysis of Musical Forms*, Summy-Birchard Inc., 1979.

[22] M. Jordan, "Dirichlet Processes, Chinese Restaurant Processes and All That," Tutorial presentation at the NIPS Conference, 2005.

[23] C. Raphael, "Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models," *IEEE TPAMI*, vol. 21, no. 4, pp. 360–370, 1999.

[24] E. Nakamura *et al.*, "A Stochastic Temporal Model of Polyphonic MIDI Performance with Ornaments," *J. New Music Res.*, vol. 44, no. 4, pp. 287–304, 2015.

[25] M. Nakano *et al.*, "Bayesian Nonparametric Music Parser," *Proc. ICASSP*, pp. 461–464, 2012.

[26] E. Nakamura *et al.*, "Tree-Structured Probabilistic Model of Monophonic Written Music Based on the Generative Theory of Tonal Music," *Proc. ICASSP*, pp. 276–280, 2016.

[27] S. Fine *et al.*, "The Hierarchical Hidden Markov Model: Analysis and Applications," *Machine Learning*, vol. 32, no. 1, pp. 41–62, 1998.

[28] M. Goto *et al.*, "RWC Music Database: Popular, Classical, and Jazz Music Databases," *Proc. ISMIR*, pp. 287–288, 2002.