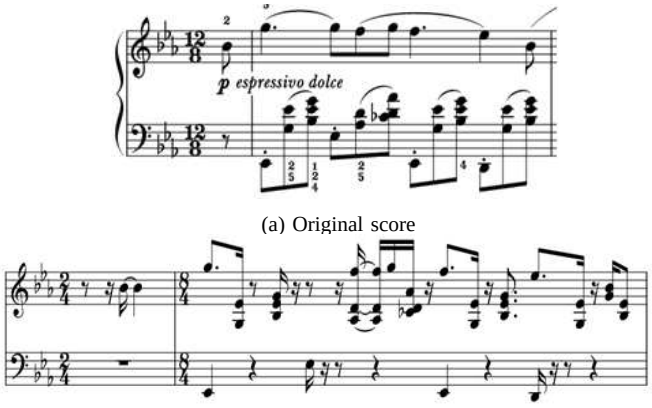
Probabilistic Model of Two-Dimensional Rhythm Tree Structure Representation for Automatic Transcription of Polyphonic MIDI Signals

Masato Tsuchiya*, Kazuki Ochiai*, Hirokazu Kameoka*, Shigeki Sagayama†
*Graduate School of Information Science and Technology, The University of Tokyo, Bunkyo-ku Hongo, Japan.
E-mail: {tsuchiya,ochiai,kameoka}@hil.t.u-tokyo.ac.jp
†National Institute of Informatics, Chiyoda-ku Hitotsubashi, Japan.
E-mail: sagayama@nii.ac.jp

***Abstract*—This paper proposes a Bayesian approach for automatic music transcription of polyphonic MIDI signals based on generative modeling of onset occurrences of musical notes. Automatic music transcription involves two subproblems that are interdependent of each other: rhythm recognition and tempo estimation. When we listen to music, we are able to recognize its rhythm and tempo (or beat location) fairly easily even though there is ambiguity in determining the individual note values and tempo. This may be made possible through our empirical knowledge about rhythm patterns and tempo variations that possibly occur in music. To automate the process of recognizing the rhythm and tempo of music, we propose modeling the generative process of a MIDI signal of polyphonic music by combining the sub-process by which a musically natural tempo curve is generated and the sub-process by which a set of note onset positions is generated based on a 2-dimensional rhythm tree structure representation of music, and develop a parameter inference algorithm for the proposed model. We show some of the transcription results obtained with the present method.**



(a) Original score

(b) Score transcribed with quantization

Fig. 1. Transcription results by using Finale2010

I. Introduction

Automatic music transcription is a process of converting musical signals into an original score, which involves multiple fundamental frequency estimation, onset detection, rhythm recognition (note value estimation) and tempo estimation. This technique can be used for a wide variety of applications including transcription systems for musical improvisations and score-based music retrieval systems. A number of transcription systems have already been developed. While there are a number of viable ways of transcribing monophonic music, polyphonic music transcription systems still pose a formidable challenge. This paper focuses on the problem of recognizing the rhythm and estimating the tempo of polyphonic music, in a situation where the pitch and onset/offset timing of each note (equivalent to a MIDI signal with absolute trigger times in units of seconds) are given.

To convert onset/offset information into a score, one simple approach would be to quantize the duration of each note. Though simple quantization methods are employed in many score notation softwares, they do not work well in the general case (see Fig. 1), since human performances usually involve fluctuations in tempo as well as in the onset/offset timings of notes. Thus, we must estimate tempo as well as rhythm (i.e., the note value of each note) to achieve accurate transcription. We shall henceforth call the problem of estimating rhythm and tempo the rhythm analysis problem. The inherent difficulty in the rhythm analysis problem lies in the chicken-and-egg interdependency between rhythm and tempo estimations. Namely, we need to know the rhythm of a piece of music to estimate the tempo and vice versa. Since the actual duration of each performed note is given by the product of the intended note value and the current instantaneous tempo [2], there can be an infinite interpretations on what the intended rhythm was and how the tempo varied if both information were missing.

When several estimation probelms have chicken-and-egg relationships, simultaneous estimation is generally preferable. Some attempts were made to simultaneously estimate tempo and rhythm of a MIDI performance by using a hidden Markov model to model a series of onset timings of polyphonic notes (e.g., [7]). With this approach, the onset timings of performed notes were projected onto a single time axis and were simply considered as a one-dimensional sequence. However, Most people would probably agree that music has a 2-dimensional hierarchical structure. In fact, polyphony usually consists of multiple independent voices and each voice has a regular tem-

poral structure (frequent motifs, phrases or melodic themes). This 2-dimensional structure characterizes an important regularity in music. Thus, simply projecting all the note onsets onto a single time axis leads to loss of a great deal of information. Motivated by this view, we successfully described the 2-dimensional hierarchical representation of onset occurences of musical notes in the form of a generative model [6], [5].

In addition to the 2-dimensional structure, music usually has a regular rhythmic structure. When listening to music, listeners do not expect to hear unnatural, irregular and rarely occurring rhythm patterns. Thus, the naturalness and regularity of rhythm patterns are important factors that allow humans to easily understand and recognize rhythm and tempo. The aim of this paper to incorporate a statistical vocabulary model of rhythm patterns into our previously developed model mentioned above.

II. Hierarchical Bayesian Generative Modeling of Note Information

A. Model Overview

Our basic strategy is to model a generative process of the onset timing of each performed note (MIDI data), and develop a parameter inference algorithm. To formulate the problem of simultaneously estimating rhythm and tempo, we propose modeling a generative process consisting of the following two sub-processes: (1) the sub-process by which the tempo curve (the track of local tempo) of a piece of music is generated, and (2) the sub-process by which a set of note onset positions (in terms of the relative time) is generated based on a 2-dimensional rhythm-vocabulary-based tree structure representation of music.

In the following, we present modeling sub-process 1 in II-B, sub-process 2 in II-C respectively, and present the way of incorporating a rhythm vocabulary model into our model in II-D. We believe that the most likely model parameters given the observation under this model would give a musically likely interpretation on a given MIDI performance (*i.e.*, a musical score). For parameter inference, we employ a Bayesian approach to infer the posterior distributions of all the model parameters. An approximate posterior inference algorithm is derived in Section III.

B. Sub-process for generating tempo curve

The tempo of a piece of music is not always constant and in most cases it varies gradually over time. If we use a [1]"tick" as a metrical time notion, an instantaneous (or local) tempo may be defined as the length of 1 tick in seconds. Now let us use μ_d to denote the real duration (in units of seconds) corresponding to the interval between d and $d+1$ ticks. Thus, μ_d corresponds to the local tempo and so the sequence $\mu_1, \ldots, \mu_D$ can be regarded as the overall tempo curve of a piece of music. One reasonable way to ensure a smooth overall change in tempo is to place a Markov-chain prior distribution over the sequence $\mu_1, \ldots, \mu_D$ that is likely to generate a sequence $\mu_1, \ldots, \mu_D$ such that $\mu_1 \simeq \mu_2, \mu_2 \simeq \mu_3, \ldots, \mu_{D-1} \simeq \mu_D$. Here, we assume the 1st order Gaussian-chain prior for convenience:

$$p(\boldsymbol{\mu}) = \prod_{d=2}^{D} \mathcal{N}\left(\mu_d; \mu_{d-1}, (\sigma^{\mu})^2\right) \tag{1}$$

where $N(x; \mu, \sigma) \propto \exp -\frac{(x-\mu)^2}{2\sigma^2}$. If we use ψ_d to denote the absolute beat time (in units of seconds) to d Tick, ψ_d can be written as $\psi_d = \psi_{d-1} + \mu_d$. which plays the role of mapping a relative time in units of ticks (integer) to an absolute time in units of seconds (continuous value).

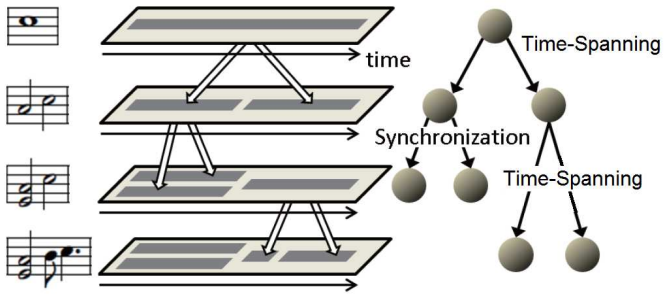


Fig. 2. Generative model of a 2-dimensional tree structure representation of musical notes.

C. Sub-process for generating score

An entire piece of music consists of many phrases. Each phrase can be decomposed into motifs, and a motif can also be split into frequently used rhythm patterns. In this sense, music has a hierarchical temporal structure. On the other hand, polyphony often consist of independent multiple parts, and each part can further be decomposed into multiple voices. Thus, we can assume that music consists of a time-spanning tree structure and a synchronizing structure of multiple notes at several levels of a hierarchy. We would like to describe this 2-dimensional tree structure representation of music in the form of a generative model.

Fig. 2 shows an example of the generative process of four musical notes in one bar of 4/4. In this example, a whole note is first split into two consecutive half notes. We call this process "time-spanning." Next, the former half note is copied in the same location, thus resulting in a chord of two half notes. We call this process "synchronization." A chord with an arbitrary number of notes can thus be generated by successively employing this type of binary production. Finally, the latter half note is split into a quaver and a dotted quarter note via the time-spanning process. This kind of generative process can be modeled by extending the idea of the probabilistic context-free grammar (PCFG) [10]. For simplicity, this paper focuses only on Chomsky normal form grammars, which consist of only two types of rules: emissions and binary productions. A PCFG is a pair consisting of a context-free grammar (a set of symbols and productions of the form $A \to BC$ or $A \to w$,

[1]Tick is a relative measure of time represented by the number of discrete divisions a quarter note has been split into. Then, if we consider 16 divisions per quarter note, for instance, the duration of 40 ticks corresponds to two-and-a-half beats.

Draw rule probability:
$\phi^T \sim \mathrm{Beta}(\phi^T; 1, \beta^T)$
[Probability of choosing either of two rule types]
For each parent node symbol G:
$\phi^B_G \sim \mathrm{Dirichlet}(\boldsymbol{\phi}^B_G; 1, \beta^B_G)$
[Probability of choosing production rules]
For each node in the parse tree:
$b_n \sim \mathrm{Bernoulli}(b_n; \phi^T)$
[Choose either EMISSION or BINARY-PRODUCTION]
If $b_n =$ EMISSION
$S_r \sim \delta_{S_r, S_n}, \quad G_r \sim \delta_{G_r, G_n}$
[Emit terminal symbol]
If $b_n =$ BINARY-PRODUCTION
$PR_n \sim \mathrm{Categorical}(PR_n; \phi^B)$
$G_{n1} \sim \delta_{G_{n1}, Left(PR_n)}, \quad G_{n2} \sim \delta_{G_{n2}, Right(PR_n)}$
[Choose production rules]
If PR_n is classified into SYNCHRONIZATION
$S_{n_1} \sim \delta_{S_{n_1}, S_n}, \quad S_{n_2} \sim \delta_{S_{n_2}, S_n}$
[Produce two synchronizing notes]
If PR_n is classified into TIME-SPANNING
$S_{n_1} \sim \delta_{S_{n_1}, S_n}, \quad S_{n_2} \sim \delta_{S_{n_2}, S_n + Length(G_{n_1})}$
[Split note n into two consecutive notes n_1 and n_2]

Fig. 3. The probabilistic formulation of generative model of a 2-dimensional tree structure representation. δ denotes Kronecker's delta. Thus, $x \sim \delta_{x,y}$ means $x = y$ with probability 1. $\mathrm{Bernoulli}(x; y)$ and $\mathrm{Beta}(y; \boldsymbol{z})$ are defined as $\mathrm{Bernoulli}(x; y) = y^x(1-y)^{1-x}$ and $\mathrm{Beta}(y; \boldsymbol{z}) \propto y^{z_1-1}(1-y)^{z_2-1}$, where $x \in \{0, 1\}$, $0 \leq y \leq 1$ and $\boldsymbol{z} = (z_1, z_2)$, respectively. $\mathrm{Discrete}(x; \boldsymbol{y})$ and $\mathrm{Dirichlet}(\boldsymbol{y}; \boldsymbol{z})$ are defined as $\mathrm{Discrete}(x; \boldsymbol{y}) = y_x$ and $\mathrm{Dirichlet}(\boldsymbol{y}; \boldsymbol{z}) \propto \prod_i y_i^{z_i - 1}$ where $\boldsymbol{y} = (y_1, \ldots, y_I)$ with $y_1 + \cdots + y_I = 1$ and $\boldsymbol{z} = (z_1, \ldots, z_I)$, respectively. Length(·) returns a note duration of each musical symbol. Left(·) and Right(·) respectively return a left and right symbol index derived from a parent node.

where A, B, and C are called "nonterminal symbols" and w is called a "terminal symbol") and production probabilities, and defines a probability distribution over the trees of symbols. The parameters of each symbol consist of (1) a distribution over rule types, (2) an emission distribution over terminal symbols, and (3) a binary production over pairs of symbols.

To describe the generative process shown in Fig. 3, we must introduce an extension of PCFG. As we explain later, we explicitly incorporate a process of stochastically choosing either "time-spanning" or "synchronization" in the binary production process. Fig. 3 defines the proposed generative process of the set of the onset positions of some number R of musical notes. In our model, each node n of the parse tree corresponds to one musical note (with no pitch information) and a pair consisting of the onset position S_n and symbol G_n of that note is considered to be a nonterminal symbol.

We first draw a "switching" distribution (namely, a Bernoulli distribution) ϕ^{T} over the two rule types {EMISSION, BINARY-PRODUCTION} from a Beta distribution. Next, we generate a discrete distribution $\boldsymbol{\phi}^{\mathrm{B}}_G = (\phi^{\mathrm{B}}_{G,PR_1}, \ldots, \phi^{\mathrm{B}}_{G,PR_K})$ over the index of production rules PR_k when the symbol of parent note is G, where K denotes the total number of the defined production rules. The shapes of all the Beta distributions and the Dirichlet distribution in our model are governed by concentration hyperparameters: β^{T} and $\boldsymbol{\beta}^{\mathrm{B}}_1, \ldots, \boldsymbol{\beta}^{\mathrm{B}}_K$.

Given a grammar, we generate a parse tree in the following manner: start with a root node that has the designated root symbol, $S_{\mathrm{Root}} = 0$ and $G_{\mathrm{Root}} =$ Start symbol whose length is equal to the overall length of a piece of music in ticks. For each nonterminal node n, we first choose a rule type b_n using ϕ^{T}. If $b_n =$ EMISSION, we produce a terminal symbol S_r with the value of S_n, namely the onset position of note r. If $b_n =$ BINARY-PRODUCTION, we then choose a production rules PR_n using ϕ^{B}. If $PR_n =$ is classified into SYNCHRONIZATION type, we produce two nonterminal children n_1 and n_2 such that $S_{n_1} = S_{n_2} = S_n$, G_{n_1} and G_{n_2} are set to the left and right components generated by binary production process, respectively. This means that the notes of the child nodes have exactly the same discrete onset. If $PR_n =$ is classified into TIME-SPANNING type, we produce two nonterminal children n_1 and n_2 with $S_{n_1} = S_n$, $S_{n_2} = S_n + \mathrm{Length}(\mathrm{Left}(S_{n_1}))$. An onset position of right node is shifted by a length of a left node. Each symbols G_{n_1} and G_{n_2} of children are determined along with SYNCHRONIZATION. We apply the procedure recursively to any nonterminal children and finally obtain a sequence $S_1, \ldots, S_R$ corresponding to the onset positions of R musical notes.

The onset position τ_r of note r should thus be placed near the absolute time into which S_r is converted. Recall that ψ_d, which can be considered a function that takes a relative time d as an input and returns the corresponding absolute time as an output, is also assumed to have been generated (via the generative process described in II-B). Given S_r and ψ_d, we find it convenient to write the generative process of τ_r as

$$\tau_r \sim \mathcal{N}(\tau_r; \psi_{S_r}, (\sigma^\tau)^2). \tag{2}$$

D. Construction of Production Rules

Here we propose incorporating a vocabulary model of rhythm patterns into the generative process described in II-C. In our previous method [5], we used simple and exhaustive production rules: if a note duration of a parent node is l times as long as the 16th note, we can derive $l - 1$ production rules from each position to split. Though this exhaustive production rules are both simple and easy to implement, the search space becomes extremely large, thus increasing the possibility of obtaining undesired (incorrect) rhythm estimates.

When listening to a piece of music, even unskilled musicians and listeners are able to recognize its rhythm [2]. Humans seem to perceive rhythm not necessarily in a note-by-note manner, but rather as larger perceptual entities or units [1]. Namely, it is likely that the onset timings of a set of notes are categorized and recognized as a particular rhythm pattern. This is called categorical perception [13], which has been studied extensively to account for the mechanism of understanding speech and vision by humans. It is thus reasonable that the modern speech recognition systems employ a vocabulary model in order to recognize speech as a concatenation of words (rather than phonemes). Automatic music transcription bears

a lot of resemblance to speech recognition because it is also a process of converting an audio signal into original symbolic information. By analogy with speech recognition, transcription based on the exhaustive production rules is equivalent to recognizing speech in a phoneme-by-phoneme manner.

It is likely that humans recognize rhythm not necessarily by perceiving each interonset interval as a particular note value but rather by perceiving a set of interonset intervals as a particular rhythm pattern. We would like to mimic this perception process in a computaionally reasonable way. We call a dictionary of rhythm patterns "rhythm vocabulary" (analogous to vocabulary in natural language) similar to the one proposed in [7]. By selecting frequently occurring rhythm patterns from musical scores, we can define them as non-terminal symbols. Defining production rules using the defined symbols would allow for a transcription with a unit of rhythm patterns. The production rules with rhythm vocabularies can be derived from actual pieces of music. For example, if an original score is Fig. 4(a) and rhythm patterns are defined at the granularity smaller than half note, the production rules can be defined as Fig. 4(b)(c)(d) shows. We omitted rest notations from the set of symbols for simplicity, and rest notations should be dealt with in the future. Of course, we cannot predict how we should define the production rules and rhythm vocabulary prior to analysis. Therefore, similarly to speech recognition systems, it is important that we must change the set of production rules and rhythm vocabulary according to genres, composers, and so forth.

III. Approximate posterior inference

A. Bayesian inference approach

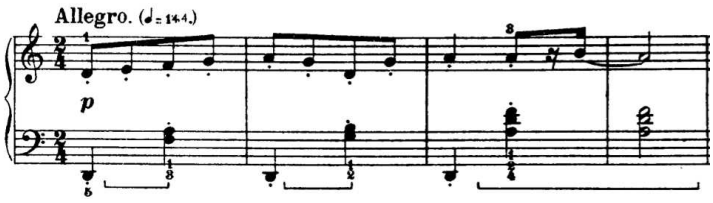
So far we have presented our proposal generative model. In this section, we describe an inference algorithm approximating a posterior distribution for our generative model via Markov Chain Monte Carlo (MCMC) method. The probabilistic variables of interest in our model are

$$
\begin{aligned}
&\psi = \{\psi_d\}_d && : \text{absolute time corresponding to } d\text{-th ticks},\\
&\mu = \{\mu_d\}_d && : \text{local tempo between } d\text{-th and } d+1\text{-th ticks},\\
&S = \{S_r\}_r && : \text{onset position of note } r \text{ (in ticks)},\\
&G = \{G_r\}_r && : \text{symbol of note } r \text{ , and}\\
&\phi^{\mathrm{B}},\ \phi^{\mathrm{T}} && : \text{rule probabilities},
\end{aligned}
$$

which we denote as Θ. The subscript r denotes an order of discretized onsets generation differentiated from the subscript i denoting an order of observed onsets i, and they differ from each other in many cases. Our goal is to compute the posterior $p(\Theta|\tau)$ where $\tau = \{\tau_i\}$ is a set consisting of observed onsets. Unfortunately, it is difficult to obtain the exact posterior $p(\Theta|\tau)$, because computing $p(\tau)$ involve many intractable integrals. By using the conditional distributions defined in II-B and II-C we can write the joint distribution $p(\tau, \Theta)$ as

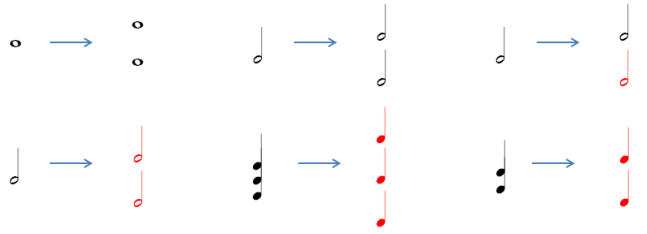
$$
\begin{aligned}
&p(\tau, \psi, \mu, S, G, \boldsymbol{\phi}^B, \phi^T)\\
&= p(\tau|\psi, S)p(\psi|\mu)p(\mu)p(S, G|\boldsymbol{\phi}^B, \phi^T)p(\boldsymbol{\phi}^B)p(\phi^T). \quad (3)
\end{aligned}
$$

To obtain the distribution of $p(\tau)$, we need marginalize out a lot of variables from the joint distribution $p(\tau, \Theta)$.

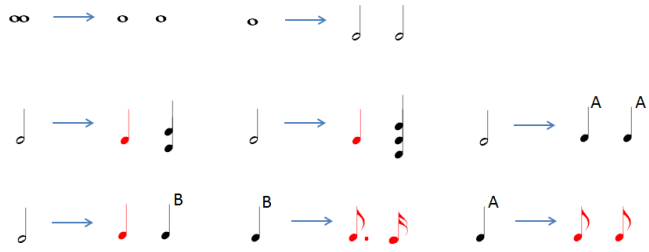


(a) An example score

(b) Muscial symbols consisting of terminal symbols (red) and non-terminal symbols (black). The longitudinally concecutive two or three notes denote chord symbol.



(c) Synchronization type production rules



(d) Time-spanning type production rules

Fig. 4. Definition of the symbols and production rules by which an example score can be generated.

However, the posterior $p(\Theta|\tau)$ can be approximated by using Gibbs sampling algorithm. In Gibbs sampling procedure, the value of one of the probabilistic variables is replaced with a new value drawn from the distribution conditioned on all the other remaining variables. For our formulation, suppose at each step t we have a set of sampled variables over the posterior distribution $p(\Theta^{(t)}|\tau)$. Then, each variable is alternately sampled from the following conditioned distribuion:

$$\boldsymbol{\psi}^{(t+1)}, \boldsymbol{\mu}^{(t+1)} \sim p(\boldsymbol{\psi}, \boldsymbol{\mu}|\tau, S^{(t)}, G^{(t)}, \boldsymbol{\phi}^{B(t)}, \phi^{T(t)}) \quad (4)$$

$$\boldsymbol{\phi}^{B(t+1)} \sim p(\boldsymbol{\phi}^B|\tau, \boldsymbol{\psi}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, S^{(t)}, G^{(t)}, \phi^{T(t)}) \quad (5)$$

$$\boldsymbol{\phi}^{T(t+1)} \sim p(\boldsymbol{\phi}^T|\tau, \boldsymbol{\psi}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, S^{(t)}, G^{(t)}, \boldsymbol{\phi}^{B(t+1)}) \quad (6)$$

$$S^{(t+1)}, G^{(t+1)} \sim p(S, G|\tau, \boldsymbol{\psi}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\phi}^{B(t+1)}, \boldsymbol{\phi}^{T(t+1)}) \quad (7)$$

On condition that the Markov chain derived from a model have ergodicity properties, by cycling through all the variables the posterior distribution $p(\Theta^{(t)}|\tau)$ will theoretically converge to a global optimum. (5) and (6) are performed only when we want to learn the rule probabilities. These update formulas of

the probabilistic variable are all given in analytical form, but they are omitted here owing to a shortage of writing spaces.

Because the basic Gibbs sampling algorithm is based on the assumption that each variable is independent from each other, successive sampled values indeed have strong dependencies. This way, sampling one variable at a time need a quite strong assumption, so by using other sampling algorithms (e.g. Blocking Gibbs sampling algorithm) this assumption may be partially relaxed so as not to affect the parameter inference result. Note that the order of sampling variable seems to be non-trivial and the effective way to determine this order must be investigated in the future.

B. Iterative estimation

As formulated in II-B and II-C, $\boldsymbol{\psi}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}$ can be sampled from a conditioned Gaussian distribution. $\boldsymbol{\phi}^{B(t+1)}$ and $\boldsymbol{\phi}^{T(t+1)}$ also can be sampled from a Dirichlet distribution and Beta distribution respectively. Instead of sampling $S^{(t+1)}, G^{(t+1)}$, we use the inside-outside (IO) algorithm to infer the distribution of them for simplicity. The posterior distribution converge to a local minimum by iteratively sampling with (4)–(6) and the estimation of the distribution over S, G by the IO algorithm.

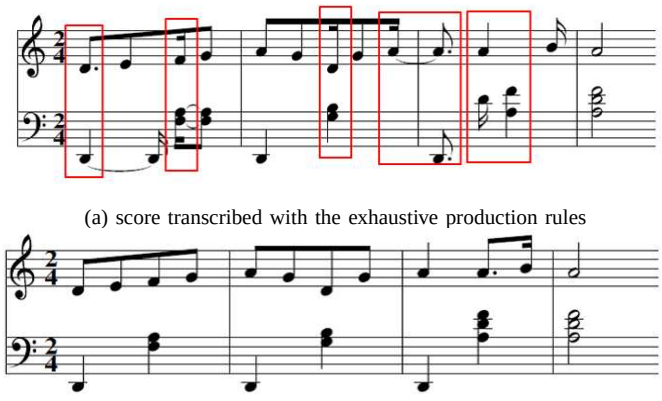
This iterative estimation algorithm is expected to give appropriate convergence of $\boldsymbol{\Theta}$ when an initial point is chosen to be close to the global minimum. An ideal initial point, however, varies depending on input MIDI signals. In order to avoid being trapped into local minimum as much as possible, we employed an annealing method used in [12].

IV. Experiment

We conducted two experiments to verify the two hypothesises described in II-D by comparing transcription accuracy rate. Five recorded MIDI signals were chosen from the CrestMuse Performance Expression Data-base (CrestMusePEDB)[9], and several parts of them were extracted such that they do not include grace notes and the notes shorter than sixteenth notes for simplicity. Because the order of the observed onsets differ from the order of notes on the score, we gave the correct permutation for each musical piece by hand. Rhythm vocabularies and production rules were listed manually such that they can generate all the experimental scores. Rhythm vocabularies were defined at the granularity shorter than a half note. The length and the total number of notes contained by a piece of music were also given. Iterative estimation was run for 80 times.

A. Evaluation of incorporating musical knowledge (rhythm vocabularies)

The objective of the first experiment is to confirm the improvement of accuracy by the effect of rhythm vocabularies. We compared the proposal method with the previous one [5], which used the exhaustively defined production rules described in II-D. In this experiment, only if the onsets position is correct, the note is classified into correct ones so as to implement fair evaluation. The results are showned in Table. I. While transcription with exhaustive production rules provided the poor results for some pieces, rhythm vocabularies overall improved the accuracy rate, especially in an up-tempo piece of music. These observations may appear incorporation of musical knowledge limited the search space and make it possible to parse at the more larger granularity level as speech is usually recognized word by word. It is worth noting that, even if all the patterns of rhyrhm vocabularies were defined, it won't be equivalent to the previous method because our method have the multiple production rules which have the same split positon. Therefore, we would suggest that even when the number of production rules has increased, the set of the rhythm vocabularies based on musical knowledge be effective for transcription task. Fig. 5 shows an example of the score obtained with the proposal and previous methods applied to Bartok's piece of music.



(a) score transcribed with the exhaustive production rules

(b) score transcribed with the rhythm vocabularies

Fig. 5. Transcription results obtained with the proposal and previous methods applied to Bartok Roumanian Folk Dance No.2. The red rectangles indicate rhythm estimation errors

B. Evaluation of unsupervised learning of the probability distribution over production rules ϕ^B, ϕ^T

The objective of the second experiment is to verify the hypothesis that most of muscial pieces have repetition of a few rhythm patterns within themselves and transcribing with the same rhythm patterns as much as possible will improve accuracy rate. Then, we evaluated both the transcription result with no learning of $\boldsymbol{\phi}^B, \phi^T$ and the result with unsupervised learning. At the latter, we obtained 10 results estimated by Gibbs sampling algorithm and calculated the worst, best, and average accuracy rate. In this evaluation, the notes whose position and symbol are equal to the corresponding note of the original score will be classified into correct notes, which is the more severe condition than that of the previous evaluation. The results are shown in Table. II. As can be seen from this result, unsupervised learning incurred an adverse effect for No.1-3 experimental data, while average accuracy rates were improved for No.4 and 5. This may be because the experimental data of No.4 and 5 include the more repetition of the same rhythm patterns within them and the posterior seemed to converge at

TABLE I
TRANSCRIPTION ACCURACY RATE OF THE EXHAUSTIVE PRODUCTION RULES AND THOSE WITH RHYTHM VOCABULARIES

No.	Experimental data	bar	notes	bpm	note level	rhythm level
	W. A. Mozart	1-4	47	Allegretto	89.3%	97.8%
1	Piano Sonata No.15 K545	9-12	69	98-109	62.3%	98.4%
	B. Bartok			Moderato		
2	Roumanian Folk Dance No.1 Sz56	19-26	64	86-97	62.5%	89.0%
	B. Bartok			Allegro		
3	Roumanian Folk Dance No.2 Sz56	1-12	106	109-132	51.4%	86.6%
	E. Grieg			Andante		
4	Lyric Pieces No.1 Arietta	1-8	103	73-77	54.3%	85.4%
	J. S. Bach			Allegro		
5	The Well-Tempered Clavier prelude No.2 BWV871	1-4	95	108-112	92.7%	95.8%

TABLE II
TRANSCRIPTION ACCURACY RATE OF NO LEARNING AND UNSUPERVISED LEARNING OF $\boldsymbol{\phi}^B, \phi^T$.

No.	no learning	unsupervised learning		
		worst	average	best
	91.5%	48.9%	74.5%	100.0%
1	96.9%	23.2%	41.0%	79.2%
2	76.6%	45.3%	76.1%	89.1%
3	76.2%	23.2%	41.0%	79.2%
4	78.6%	73.8%	86.7%	96.1%
5	91.7%	83.0%	97.9%	100.0%

an early stage of estimating iterations.

V. CONCLUSIONS

We have proposed a generative model incorporating musical knowledge for automatic polyphonic transcription of MIDI signals. Automatic music transcription involves two subproblems that are interdependent of each other: rhythm recognition and tempo estimation. To circumvent the chicken-and-egg problem, we modeled the generative process of MIDI signals by formulating the sub-process by which a musically natural tempo curve is generated and the sub-process by which a set of note discrete onset positions is generated based on a 2-dimensional rhythm tree structure representation of music. The score-generation sub-process which reflect musical knowledge is expected to improve transcription accuracy. The experiments presented that our proposal method outperformed the previous method thanks to rhythm vocabularies, showing some of the transcription results. In future work, we are going to develop a model to deal with theory of music to extract the more various musical information.

ACKNOWLEDGMENT

This research was funded in part by Ministry of Education, Culture, Sports, Science and Technology (MEXT) / Japan Science and Technology Agency (JST) contract 23240021.

REFERENCES

[1] A. T. Cemgil, P. Desain, and B. Kappen, "Rhythm quantization for transcription," *Computer Music Journal*, vol. 24, no. 2, pp. 60–76, 2000.

[2] P. Desain, and H. Honing, "The quantization of musical time: A connectionist approach," *Computer Music Journal*, vol 13, no. 3, pp. 56–66, 1989.

[3] P. Desain, R. Aarts, A. T. Cemgil, B. Kappen, H. van Thienen, and P. Trilsbeek, "Robust Time-Quantization for Music, from Performance to Score," In *Proc. Audio Engineering Society Convention*, 106, 1999.

[4] C. Raphael, "A hybrid graphical model for rhythmic parsing," *Artificial Intelligence*, vol. 137, no. 1, pp. 217–238, 2002.

[5] H. Kameoka, K. Ochiai, M. Nakano, M. Tsuchiya, and S. Sagayama, "Context-free 2D tree structure model of musical notes for Bayesian modeling of polyphonic spectrograms," In *Proc. of ISMIR2012*, 2012.

[6] M. Nakano, Y. Ohishi, H. Kameoka, R. Mukai, and K. Kashino. "Bayesian nonparametric music parser," In *Proc. ICASSP2012*, pp. 461–464, 2012.

[7] H. Takeda, T. Nishimoto, and S. Sagayama, "Rhythm and tempo analysis toward automatic music transcription," In *Proc. ICASSP2007*, vol. 4, pp. IV-1317–1320, 2007.

[8] M. Tanji and I. Hitoshi, "Metrical Structure Analysis using Extended PCFG from performance MIDI Data," *IPSJ special interest group on music and computer*, 14 (2009): 1-6. (in Japanese)

[9] M. Hashida, T. Matsui, and H. Katayose, "A new music database describing deviation information of performance expressions," In *Proc. ISMIR2008*, pp. 489–494. 2008.

[10] P. Liang, S. Petrov, M. Jordan, and D. Klein, "The infinite PCFG using hierarchical Dirichlet processes," In *Proc. EMNLP-CoNLL*, pp. 688–697. 2007.

[11] M. Hoffman, D. Blei, and P. Cook, "Bayesian nonparametric matrix factorization for recorded music," In *Proc. ICML*, pp. 439–446. 2010.

[12] H. Kameoka, T. Nishimoto, and S. Sagayama, "A multipitch analyzer based on harmonic temporal structured clustering," *IEEE Trans. ASLP*, vol. 15, no. 3, pp. 982–994, 2007.

[13] S. R. Harnad (Ed.) *Categorical perception: The groundwork of cognition*: Cambridge University Press, 1990.