Bayesian Melody Harmonization Based on a Tree-Structured Generative Model of Chord Sequences and Melodies

Hiroaki Tsushima, Eita Nakamura, and Kazuyoshi Yoshii *Member, IEEE*

***Abstract*—This paper describes a melody harmonization method that generates a sequence of chords (symbols and onset positions) for a given melody (a sequence of musical notes). A typical approach to melody harmonization is to use a hidden Markov model (HMM) that represents chords and notes as latent and observed variables, respectively. This approach, however, does not consider the syntactic functions (e.g., tonic, dominant, and subdominant) and hierarchical structure of chords that play vital roles in traditional harmony theories. In this paper, we propose a unified hierarchical generative model consisting of a probabilistic context-free grammar (PCFG) model generating chord symbols associated with syntactic functions, a metrical Markov model generating chord onset positions, and a Markov model generating a melody conditioned by a chord sequence. To estimate a musically natural tree structure, the PCFG is trained in a semi-supervised manner by using chord sequences with tree structure annotations. Given a melody, a sequence of a variable number of chords can be estimated by using a Markov chain Monte Carlo method that partially and iteratively updates the symbols, onset positions, and tree structure of chords according to the posterior distribution of chord sequences. Experimental results show that the proposed method outperformed the HMM-based method and a conventional rule-based method in terms of predictive abilities.**

***Index Terms*—Melody harmonization, probabilistic context-free grammar, Markov model.**

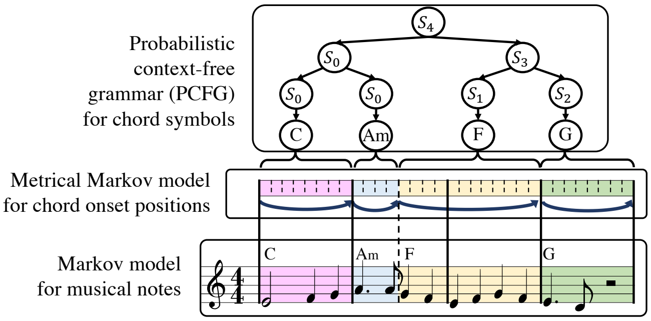

Probabilistic context-free grammar (PCFG) for chord symbols
S_4
S_0
S_3
S_0
S_0
S_1
S_2
C
Am
F
G
Metrical Markov model for chord onset positions
Markov model for musical notes
C
Am
F
G

Fig. 1. A tree-structured hierarchical generative model that stochastically generates a chord sequence and a melody in this order. The labels S_i represent syntactic categories behind chord symbols explained in Section III.

I. Introduction

CHORD arrangement is one of the most important tasks in the composition of popular music because chord sequences characterize musical moods and styles. To help musically untrained people compose original pieces, automatic melody harmonization, *i.e.*, automatic generation of a chord sequence for a given melody (note sequence), has been studied [1]–[14]. For this, it is necessary to computationally characterize acceptable chord sequences and the relationships between chords and melody notes. Statistical modeling is effective for data-driven induction of such musical grammar and can be applied to various musical styles [7], [10], [11].

A typical approach for melody harmonization based on statistical modeling is to integrate a probabilistic model representing a chord sequence with one representing the relationships between chords and a melody. If the former is a Markov or semi-Markov model and the latter is a generative model of a melody given a chord sequence, the integrated model is described as a hidden Markov model (HMM) [7] or hidden semi-Markov model (HSMM) [10]. In this model, the sequential dependency of chord symbols is described with transition probabilities, which can be learned from data.

This approach, however, does not consider that some chords play a similar syntactic role and that music has a hierarchical structure (phrase, section, etc.) [15]–[17]. In harmony theories, the syntactic roles of chord symbols are known as *harmonic functions* such as tonic (T), dominant (D), and subdominant (SD)[1]. The hierarchical structure of chords or harmonic functions is often represented as a tree [17]–[21]. For example, a chord sequence (C, Dm, G, Am, C, F, G, C) in C major key can be interpreted as a function sequence (T, SD, D, T, T, SD, D, T), which is further parsed as a binary tree ((T, ((SD, D), T)), (T, ((SD, D), T))), where subtrees such as (SD, D) and (T, ((SD, D), T)) appear repeatedly in different abstraction levels. It has been shown that statistical models such as an HMM and a probabilistic context-free grammar (PCFG) model that can represent the syntactic functions and hierarchical structure of chords outperform Markov models in predictive ability [22].

Manuscript received XXXX XX, 2019; revised XXXX XX, 2019; accepted XXXX XX, 2020. Date of publication XXXX XX, 2020; date of current version XXXX XX, 2020. This work was partially supported by JSPS KAKENHI Nos. 26700020, 16H01744, 16J05486, 16H02917, 16K00501, and 19K20340, JST ACCEL No. JPMJAC1602, and Kayamori Foundation. The associate editor coordinating the review of this manuscript and approving it for publication is Prof. XXX YYY (Corresponding author: Eita Nakamura).

H. Tsushima, E. Nakamura, and K. Yoshii are with Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan, email: {tsushima, enakamura, yoshii}@sap.ist.i.kyoto-u.ac.jp. E. Nakamura is also with the Hakubi Center for Advanced Research, Kyoto University, Kyoto 606-8501, Japan.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier XXXX

[1] In the simple cases consisting of triads, these functions are associated with the three main triads (C, G, and F in C major key) and their relative minor chords.

Another problem with the conventional Markov or semi-Markov models is that the metrical structure of chord onset positions cannot be modeled. Chord transitions are considered for a fixed time step (*e.g.*, 16th note) and in semi-Markov models chord durations are explicitly modeled. While chord onsets tend to appear more frequently at strong beats, such metrical structure cannot be described with these conventional Markov or semi-Markov models.

In this paper we propose a statistical melody harmonization method incorporating syntactic functions, tree structure, and the metrical structure of chords, aiming to generate musically meaningful chord sequences. We formulate a unified hierarchical generative model that consists of (1) a PCFG model representing the generative process of chord symbols, (2) a metrical Markov model [23], [24] representing that of chord onset positions, and (3) a Markov model representing that of a note sequence from a chord sequence (Fig. 1). The harmonic functions and hierarchical structures of chords are represented by the leaf and inside nodes of a latent tree behind chords. The metrical structure of chord rhythms is represented by the metrical Markov model. Given a melody, chord sequences with a variable number of symbols can be sampled from the posterior distribution by using a Markov chain Monte Carlo (MCMC) method.

For application of statistical models, it is crucial to train the parameters properly from data. In our previous work, we developed an unsupervised learning method for the PCFG model and experimentally showed the potential of this model for melody harmonization [12], [13]. The estimated latent trees behind chord sequences, however, often mismatch human's structural interpretations and the leaf nodes of the trees fail to represent the harmonic functions of chords. This is mainly because the PCFG learning tends to easily get stuck in bad local optima due to its strong sensitivity to initialization. It is thus hard to induce musically meaningful latent trees from non-annotated chord sequences like (C, Dm, G, Am, C, F, G, C) in an *unsupervised* manner.

To solve this problem, we propose two refined methods for learning the PCFG model. First, we develop an initialization method to make the model's grammar more similar to human's interpretation. We initialize the PCFG model by using an HMM that represents harmonic functions and chords as latent and observed variables, respectively (Section III-C3). Second, we develop a learning method using additional data to make the model predict latent trees similar to human's structural annotations. We train the PCFG model in a *semi-supervised* manner by using *weakly annotated* chord sequences like ((C, ((Dm, G), Am)), (C, ((F, G), C))) as explained in Section III-C2. We confirm that these methods contribute to estimating latent trees behind chords that represent hierarchical musical structure (Section V).

The major contribution of this study is to present a Bayesian inference method for the unified probabilistic model of chords and melodies, which is an extension of a previous study [12]. Another contribution is a comprehensive report of the experimental evaluation results. We also show the effectiveness of the initialization and semi-supervised learning methods by systematic comparisons. Sections III-C2 and III-C3 present new technical improvements and Section V presents new experimental results. The other sections describe the method proposed in [12] in full detail.

In Section II of this paper, we review related work on melody harmonization and music language modeling. Section III formulates the proposed model and Section IV presents the melody harmonization method based on the model. Section V reports the experimental evaluations. We conclude with a brief summary and mention of future work in Section VI.

II. Related Work

This section reviews related studies on melody harmonization and on music language models of chords and melodies.

A. Melody Harmonization

Melody harmonization systems can be roughly categorized into two types in terms of their objectives. In the first type of systems, the aim is to generate a sequence of chord symbols given a melody, and in the second type of systems, the aim is to generate multiple voices of musical notes.

As a study on the first type of systems, Chuan and Chew [6] proposed a method that selects musical notes from a given melody that are likely to form chords by using a support vector machine, constructs triads from the selected notes, and makes a chord sequence by using hand-crafted rules. Simon *et al.* [7] developed a commercial system called *MySong* based on an HMM representing sequential chord transitions, and Raczyński *et al.* [11] proposed a similar Markov model representing chords conditioned by melodies and time-varying keys. Tsushima *et al.* [12] proposed a harmonization method based on a PCFG model representing the hierarchical structure of chords and a Markov model representing the transitions of melody notes. De Prisco *et al.* [9] proposed a harmonization method for bass-line inputs based on a distinctive network that represents the relationships between bass notes, the previous chord, and the current chord.

Recently, a deep neural network (DNN)-based method for melody harmonization was proposed by Lim *et al.* and was shown to outperform an HMM-based method in both objective and subjective evaluation metrics [25]. An advantage of our approach over such DNN-based approaches is that the proposed model can learn and represent the grammatical structure of music in a similar way as humans do, which is considered essential for enhancing the directability of interactive composition systems [13]. In contrast, it is generally difficult to analyze or interpret the internal structure of DNNs.

As a study on the second type of systems, Ebcioğlu [1] proposed a rule-based method for generating four-part chorales in Bach's style. Several methods using variants of genetic algorithms based on music theories have also been proposed [2], [3], [8]. Allan and Williams [4] proposed an HMM-based method that represents chords as latent variables and notes as observed outputs. An HSMM has been used for explicitly representing the durations of chords [10]. Paiement *et al.* [5] proposed a hierarchical tree-structured model that describes chord movements from the viewpoint of hierarchical time scales by dividing the notations of chords. To generate four-part chorales, a deep recurrent neural network has also been used for capturing the long-term dependency characteristics of melodies and harmonies [14].

B. Statistical Modeling of Melodies and Chords

Statistical models of melodies play an important role in the tasks of automatic harmonization and melody generation. The standard approach for modeling sequences of musical notes (melodies) is to use Markov models [26]. Pachet *et al.* [27] proposed an efficient approach for generating a melody based on Markov models with constraints. To describe the generative process of melodies conditioned on chords, Simon *et al.* [7] proposed a generative model based on HMMs and bag-of-words that considers chord symbols as latent variables and notes as observed words.

In the generative theory of tonal music (GTTM) [17], a note sequence is assumed to form a tree that describes the degrees of the relative importance of individual notes. This theory consists of many hand-crafted rules used for recursively reducing a note sequence to a single note. Computational implementation of the GTTM and its application to music analysis have been studied [28], [29]. A probabilistic formulation of the GTTM based on a PCFG was recently proposed for learning the production rules of the PCFG from melodies [30].

Statistical models of chord sequences have been actively investigated for automatic chord estimation [31]–[33], musical grammar analysis [18], [19], and automatic music arrangement [21], [34]. The most popular approach is to use a Markov (or n-gram) model [31], [34]. To avoid the sparseness problem with a large value of n, various smoothing methods have been proposed [35]. Yoshii *et al.* [32] proposed a vocabulary-free infinity-gram model that represents the dependency of a chord on a variable-length history of chords.

To make generated chord sequences musically natural and interpretable, it is considered effective to describe the musical properties of chords (*e.g.*, syntactic categories, cadential properties, and hierarchical structure) as in harmony theories. Tsushima *et al.* [22] attempted unsupervised learning of HMMs for spontaneously discovering the syntactic categories of chords. As a result, the learned syntactic categories corresponded to the harmonic functions and the trained HMM outperformed Markov models in predictive ability.

To capture the syntactic structure behind chord sequences, tree-structured models have been studied. Paiement *et al.* [33] formulated several hidden layers of state transitions for representing the tree structure behind chords. Some studies attempted to explicitly describe the generative grammar to represent the hierarchical structure of chords [18]–[21]. Steedman [18] and Rohrmeier [19] proposed CFG-based production rules for chord sequences. A probabilistic extension was later studied for music arrangement [21]. In these studies, a list of non-terminal symbols and that of production rules were manually given based on music theories or musical intuition.

III. Probabilistic Modeling

This section explains the probabilistic model of chords and melody notes. After introducing some mathematical notations in Section III-A, the model is formulated in Section III-B. Methods for training the model parameters are described in Section III-C. For simplicity of notation, we consider that data consist of only one pair of a chord sequence and a melody. Extension for the case with multiple sequences is straightforward.

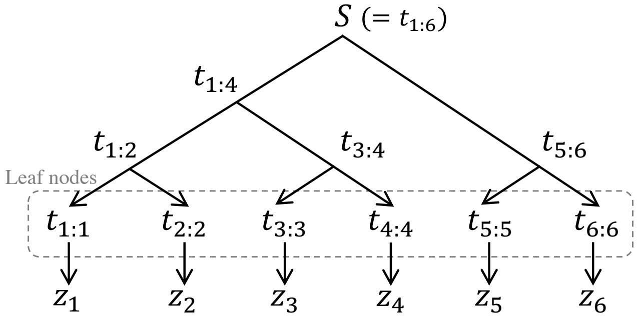


Fig. 2. Mathematical notation for tree structure $\mathbf{t}$ behind chord symbols $\mathbf{z}$ based on a PCFG model. In this example, the number of chords N is 6 and S represents the start symbol.

TABLE I
Mathematical notation in this paper

Symbol	Meaning	Section
N	Number of chord symbols	III-A
T	Number of 16th-note-level time units	III-A
$\mathbf{z}$	Chord symbols	III-A
$\mathbf{y}$	Chord onset positions	III-A
$\mathbf{x}$	Pitches of melody notes	III-A
$\mathbf{o}$	Onset positions of melody notes	III-A
$\mathbf{t}$	Tree	III-B1
$\hat{\mathbf{t}}$	Tree structure without node labels	III-C2
K	Number of distinct non-terminal symbols	III-B1
$\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\lambda}$	Parameters of PCFG	III-B1
$\boldsymbol{\pi}$	Transition probabilities over chord onset positions	III-B2
$\boldsymbol{\tau}$	Transition probabilities over melody pitches	III-B3

A. Mathematical Notation

A list of the mathematical symbols is provided in Table I. In this paper, we assume that musical pieces have the time signature of 4/4 and the onset positions of chords and melody notes are on the 16th-note-level grid. Let L be the number of measures of a musical piece ($L = 8$ in this paper) and $T = 16L$ be the total number of time units. A sequence of chord symbols and their onset positions are denoted by $\mathbf{z} = \{z_n\}_{n=1}^{N}$ and $\mathbf{y} = \{y_n\}_{n=1}^{N}$, respectively, where N is the number of chords and each y_n takes an integer in $[0, T)$. Similarly, the pitches and onset positions of melody notes in the region of chord z_n are denoted by $\mathbf{x}_n = \{x_{n,i}\}_{i=1}^{I_n}$ and $\mathbf{o}_n = \{o_{n,i}\}_{i=1}^{I_n}$, respectively, where I_n is the number of melody notes in the region, $x_{n,i}$ is a MIDI note number from 33 (A1) to 92 (G#6), and $o_{n,i}$ takes an integer in $[y_n, y_{n+1})$. The whole melody is denoted by $\mathbf{x} = \{\mathbf{x}_n\}_{n=1}^{N}$ and $\mathbf{y}$. Let $I = \sum_{n=1}^{N} I_n$ be the total number of melody notes. We use the integer pitch representation instead of the spelled pitch representation and treat enharmonic notes equivalently. Since most notes in the

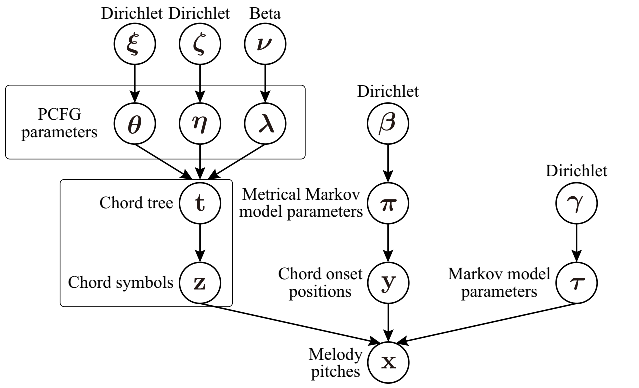


Fig. 3. Dependencies of variables and parameters of the generative model.

popular music data we use in this study are on the diatonic scale, the information lost in the integer pitch representation is little and is considered to have a small effect.

B. Model Formulation

We formulate a unified hierarchical generative model of chord symbols $\mathbf{z}$, chord onset positions $\mathbf{y}$, and melody pitches $\mathbf{x}$ (Fig. 3). This model consists of three sub-models: (1) a PCFG model of $\mathbf{z}$, (2) a metrical Markov model of $\mathbf{y}$, and (3) a Markov model of $\mathbf{x}$ conditioned by $\mathbf{z}$ and $\mathbf{y}$.

1) PCFG Model of Chord Symbols: Following the idea of previous studies [18], [19], [21], we assume that chord symbols $\mathbf{z}$ are derived from a latent tree $\mathbf{t}$ in a similar way that words are derived from a syntactic tree in the phrase structure grammar of natural language. A derivation tree $\mathbf{t}$ and chord symbols $\mathbf{z}$ are generated in this order according to a PCFG $G = (V, \Sigma, R, S)$ defined by a set of non-terminal symbols $V = \{S_k\}_{k=1}^{K}$, a set of terminal symbols (chord symbols) Σ, a set of rule probabilities R, and a start symbol S (a non-terminal symbol at the root of $\mathbf{t}$), where K is the number of distinct non-terminal symbols. The non-terminal symbols are expected to represent the syntactic roles and hierarchical structures of chord symbols. There are three types of rule probabilities. $\theta(A \rightarrow BC)$ is the probability that a non-terminal symbol $A \in V$ branches into non-terminal symbols $B \in V$ and $C \in V$. $\eta(A \rightarrow \alpha)$ is the probability that $A \in V$ emits terminal symbol $\alpha \in \Sigma$. $\lambda_A \in [0,1]$ is the probability that a non-terminal symbol $A \in V$ emits a terminal symbol, and otherwise it branches. The first two probabilities are normalized as follows:

$$\sum_{B,C \in V} \theta(A \rightarrow BC) = 1, \quad \sum_{\alpha \in \Sigma} \eta(A \rightarrow \alpha) = 1. \tag{1}$$

A tree and a chord sequence are generated by a cascading process of generating non-terminal symbols from the start symbol and finally terminal symbols from the non-terminal symbols. The probability of the tree and chord sequence is given as a product of relevant branching and emitting probabilities. We write $\boldsymbol{\theta}_A = \{\theta(A \rightarrow BC)\}_{B,C \in V}$, $\boldsymbol{\theta} = \{\boldsymbol{\theta}_A\}_{A \in V}$, $\boldsymbol{\eta}_A = \{\eta(A \rightarrow \alpha)\}_{\alpha \in \Sigma}$, $\boldsymbol{\eta} = \{\boldsymbol{\eta}_A\}_{A \in V}$, and $\boldsymbol{\lambda} = \{\lambda_A\}_{A \in V}$. Similar notations are used throughout this paper.

A subtree of $\mathbf{t}$ that derives $z_{m:n} = \{z_m, z_{m+1}, ..., z_n\}$ is denoted by $t_{m:n}$ (Fig. 2). In particular, we have $\mathbf{t} = t_{1:N}$. We also use the notation $t_{m:n}$ to indicate the root node of the subtree for simplicity. In this paper, we refer to the nodes $\{t_{n:n}\}_{n=1}^{N}$ that emit chord symbols as *leaf nodes*.

2) Metrical Markov Model of Chord Onset Positions: The metrical Markov model [23], [24] of chord onset positions $\mathbf{y}$ on the 16th-note-level grid is defined as follows:

$$p(y_n | y_{n-1}) = \pi_{y_{n-1} \mathrm{mod} 16, y_n - y_{n-1}}, \tag{2}$$

where $\pi_{a,b}$ indicates the probability that a chord has an onset at the a-th position in a measure ($0 \leq a < 16$) and a duration of b time units ($0 < b \leq T$). We write $\boldsymbol{\pi}_a = \{\pi_{a,b}\}_{b=0}^{T}$

3) Markov Model of Melody Pitches: The Markov model of melody pitches $\mathbf{x}$ conditioned by chord symbols $\mathbf{z}$ and chord onset positions $\mathbf{y}$ is defined as follows:

$$p(x_{n,1} | x_{n-1,I_{n-1}}, z_n) = \tau^{z_n}(x_{n-1,I_{n-1}}, x_{n,1}), \tag{3}$$

$$p(x_{n,i} | x_{n,i-1}, z_n) = \tau^{z_n}(x_{n,i-1}, x_{n,i}) \ (2 \leq i \leq I_n), \tag{4}$$

where $\tau^c(a,b)$ is the transition probability from pitch a to pitch b under chord symbol c, that is, the probability that pitch a would be generated following pitch b in the time span of chord symbol c. We write $\boldsymbol{\tau}_a^c = \{\tau^c(a,b)\}_{b=33}^{92}$ (33 = A1 and 92 = G#6).

4) Hierarchical Bayesian Integration of Three Sub-Models: Let $\boldsymbol{\Omega} = \{\mathbf{t}, \mathbf{z}, \mathbf{y}, \mathbf{x}\}$ be the set of the latent and observed variables and $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\lambda}, \boldsymbol{\pi}, \boldsymbol{\tau}\}$ be the set of model parameters. Assuming that chord symbols and melody pitches are generated independently, the whole model is given by

$$\begin{aligned} p(\boldsymbol{\Omega}, \boldsymbol{\Theta}) &= p(\boldsymbol{\Omega}|\boldsymbol{\Theta}) p(\boldsymbol{\Theta}) \\ &= p(\mathbf{t}, \mathbf{z} | \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\lambda}) p(\mathbf{y}|\boldsymbol{\pi}) p(\mathbf{x}|\mathbf{z}, \mathbf{y}, \boldsymbol{\tau}) p(\boldsymbol{\Theta}), \end{aligned} \tag{5}$$

where $p(\mathbf{t}, \mathbf{z} | \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\lambda})$ is the probability of chords and a tree given by the PCFG model, $p(\mathbf{y}|\boldsymbol{\pi})$ is the probability of chord onset positions given by the metrical Markov model, $p(\mathbf{x}|\mathbf{z}, \mathbf{y}, \boldsymbol{\tau})$ is the probability of melody pitches given by the Markov model, and $p(\boldsymbol{\Theta}) = p(\boldsymbol{\theta}) p(\boldsymbol{\eta}) p(\boldsymbol{\lambda}) p(\boldsymbol{\pi}) p(\boldsymbol{\tau})$ is a prior distribution over the model parameters. We consider prior distributions on the model parameters to make the learning process more efficiently (avoiding getting stuck in bad local optima) [22] and to induce the model's probability parameters sparser, which tends to represent syntactic structure similar to human's interpretation [36]. To make Bayesian inference tractable, we use conjugate Dirichlet and beta priors:

$$\boldsymbol{\theta}_A \sim \mathrm{Dir}(\boldsymbol{\xi}_A), \ \boldsymbol{\eta}_A \sim \mathrm{Dir}(\boldsymbol{\zeta}_A), \ \lambda_A \sim \mathrm{Beta}(\boldsymbol{\nu}_A), \tag{6}$$

$$\boldsymbol{\pi}_a \sim \mathrm{Dir}(\boldsymbol{\beta}_a), \ \boldsymbol{\tau}_a^c \sim \mathrm{Dir}(\boldsymbol{\gamma}_a^c), \tag{7}$$

where $\boldsymbol{\xi}_A$, $\boldsymbol{\zeta}_A$, $\boldsymbol{\nu}_A$, $\boldsymbol{\beta}_a$, and $\boldsymbol{\gamma}_a^c$ are hyperparameters. It is known that smaller values of these hyperparameters make the probability parameters sparser.

C. Model Training

The three sub-models described in Section III-B can be trained separately using different types of training data. For training the PCFG model, the basic method [12] of unsupervised learning is explained in Section III-C1. As refinements to this method, we propose a semi-supervised learning method using tree structure annotations on $\mathbf{z}$ in Section III-C2 and an

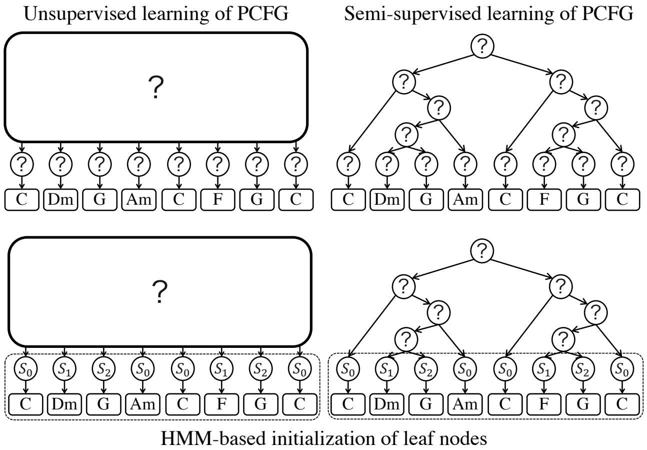

Unsupervised learning of PCFG
Semi-supervised learning of PCFG
?
?
C Dm G Am C F G C
S_0 S_1 S_2 S_0 S_0 S_1 S_2 S_0
HMM-based initialization of leaf nodes

Fig. 4. Four possible learning methods for the PCFG model. The left and right columns show unsupervised and semi-supervised learning methods, respectively. The top and bottom rows show the cases with and without an HMM-based initialization method. Question marks indicate the variables estimated in unsupervised learning.

initialization method based on an HMM of $\mathbf{z}$ in Section III-C3, to make a tree $\mathbf{t}$ estimated by the PCFG model close to human interpretation. Combining these two refinements, we have four possible learning methods shown in Fig. 4. In Sections III-C4 and III-C5, we explain how to train the metrical Markov model of chord onset positions $\mathbf{y}$ and the Markov model of melody pitches $\mathbf{x}$.

1) Unsupervised Learning of PCFG: Our goal is to obtain the maximum-a-posteriori (MAP) estimate of the model parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\lambda}, \boldsymbol{\pi}, \boldsymbol{\tau}\}$. To estimate the parameters $\boldsymbol{\theta}$, $\boldsymbol{\eta}$, and $\boldsymbol{\lambda}$ from chord symbols $\mathbf{z}$ in an unsupervised manner, we draw samples from the posterior distribution $p(\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\lambda}, \mathbf{t}|\mathbf{z})$ by using a Gibbs sampling method based on the inside filtering-outside sampling algorithm [12], [36]. More specifically, the latent tree $\mathbf{t}$ and the parameters $\boldsymbol{\theta}$, $\boldsymbol{\eta}$, and $\boldsymbol{\lambda}$ are alternately updated according to the conditional posterior distributions $p(\mathbf{t}|\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\lambda}, \mathbf{z})$ and $p(\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\lambda}|\mathbf{t}, \mathbf{z})$, respectively.

In the inside filtering step, the conditional probability (inside message) that a subsequence $z_{n:m}$ is derived from a subtree whose root node is A is denoted by

$$p^A_{n,m} = p(z_{n:m}|t_{n:m} = A). \tag{8}$$

This probability can be calculated recursively from the leaf nodes to the root node as follows:

$$p^A_{n,n} = \lambda_A \, \eta(A \to z_n), \tag{9}$$

$$p^A_{n,n+k} = \sum_{B,C \in V} (1-\lambda_A)\theta(A \to BC) \sum_{1 \le l \le k} p^B_{n,n+l-1} p^C_{n+l,n+k}. \tag{10}$$

In the outside sampling step, a latent tree $\mathbf{t}$ is obtained by recursively branching paths from the start symbol S to the leaf nodes according to $p(\mathbf{t}|\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\lambda}, \mathbf{z})$. Suppose that we already have a node $t_{n:n+k} = A$. Two non-terminal symbols B and C derived from $t_{n:n+k}$ are then sampled according to

$$\begin{aligned} &p(l, B, C) \\ &= p(t_{n:n+l-1} = B, t_{n+l:n+k} = C \,|\, t_{n:n+k} = A, z_{n:n+k}) \\ &= (1-\lambda_A)\theta(A \to BC)\, p^B_{n,n+l-1}\, p^C_{n+l,n+k} / p^A_{n,n+k}, \end{aligned} \tag{11}$$

where $l \in \{1, \ldots, k\}$ indicates a split position.

Finally, we sample parameters $\boldsymbol{\theta}$, $\boldsymbol{\eta}$, and $\boldsymbol{\lambda}$ according to $p(\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\lambda}|\mathbf{t}, \mathbf{z}) = p(\boldsymbol{\theta}|\mathbf{t}, \mathbf{z})p(\boldsymbol{\eta}|\mathbf{t}, \mathbf{z})p(\boldsymbol{\lambda}|\mathbf{t}, \mathbf{z})$ as follows:

$$\boldsymbol{\theta}_A \mid \mathbf{t}, \mathbf{z} \sim \text{Dir}(\boldsymbol{\xi}_A + \boldsymbol{u}_A), \tag{12}$$

$$\boldsymbol{\eta}_A \mid \mathbf{t}, \mathbf{z} \sim \text{Dir}(\boldsymbol{\zeta}_A + \boldsymbol{v}_A), \tag{13}$$

$$\lambda_A \mid \mathbf{t}, \mathbf{z} \sim \text{Beta}(\boldsymbol{\nu}_A + \boldsymbol{w}_A), \tag{14}$$

where $u(A \to BC)$ is the number of times that a binary production rule $A \to BC$ is used, $v(A \to \alpha)$ is the number of times that an emission rule $A \to \alpha$ is used, $w_{A,0}$ is the number of times that a non-terminal symbol A branches into two non-terminal symbols, and $w_{A,1}$ is the number of times that a non-terminal symbol A emits a chord.

2) Semi-Supervised Learning of PCFG: The PCFG model can be trained in a semi-supervised manner by using tree-structure annotations. Such annotations can be represented as S-expressions on $\mathbf{z}$ (*e.g.*, ((C, ((Dm, G), Am)))) that specify the shape of a tree $\mathbf{t}$. The tree shapes of $\mathbf{t}$ and $t_{m:n}$ without information about non-terminal symbols are denoted by $\hat{\mathbf{t}}$ and $\hat{t}_{m:n}$, respectively.

To estimate the parameters $\boldsymbol{\theta}$, $\boldsymbol{\eta}$, and $\boldsymbol{\lambda}$ from chord symbols $\mathbf{z}$ with a tree structure $\hat{\mathbf{t}}$, we extend the inside-filtering algorithm. More specifically, we calculate the posterior distribution $p(\mathbf{t}|\hat{\mathbf{t}}, \mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\lambda})$ of a tree $\mathbf{t}$ having a fixed shape $\hat{\mathbf{t}}$ by modifying Eq. (10) as follows:

$$p^A_{n,n+k} = \sum_{B,C \in V} (1-\lambda_A)\theta(A \to BC) q(B, C, n, k) \tag{15}$$

$$q(B, C, n, k) = \begin{cases} p^B_{n,n+l-1} p^C_{n+l,n+k} \quad \text{if } \exists l \in [1, k] \\ \quad \text{s.t. } \hat{t}_{n:n+l-1} \Rightarrow z_{n:n+l-1},\ \hat{t}_{n+l:n+k} \Rightarrow z_{n+l:n+k}, \\ 0 \quad \text{otherwise}, \end{cases}$$

where $\hat{t}_{m:n} \Rightarrow z_{m:n}$ means that a subtree $\hat{t}_{m:n}$ derives chord symbols $z_{m:n}$. Because of the constraints of the probabilities in the above expression, the trees sampled in the outside sampling step match the tree shape $\hat{\mathbf{t}}$.

3) HMM-Based Initialization of PCFG: In the unsupervised learning of the PCFG model (Section III-C1), non-terminal symbols given to the leaf nodes of a tree $\mathbf{t}$ tend to mismatch the musically meaningful harmonic functions. To solve this, we propose a technique for initializing the PCFG model by using an HMM. This technique is motivated by the fact that the HMM is able to automatically learn syntactic categories corresponding to the harmonic functions [22].

First, an HMM that represents chords and syntactic functions as observed and latent variables, respectively, is trained by using Gibbs sampling (see [22] for details). The most likely sequence of latent variables $\{y_n\}_{n=1}^N$ is then estimated by using the Viterbi algorithm. These latent variables are used to specify the non-terminal symbols of the leaf nodes $\{t_{n:n}\}_{n=1}^N$

of a latent tree. Specifically, in the first inside-filtering step of Gibbs sampling for the PCFG, we use the following formula instead of Eq. (9) to initialize the leaf nodes:

$$p^A_{n,n} = \begin{cases} \lambda_A \text{ (if } A = y_n), \\ 0 \text{ (otherwise)}. \end{cases} \tag{16}$$

The probability parameters are randomly initialized. This technique can be used with the semi-supervised learning of the PCFG in Section III-C2.

4) Bayesian Learning of Metrical Markov Model: Given chord onset positions $\mathbf{y}$, the posterior distribution of $\boldsymbol{\pi}$ can be analytically calculated as follows:

$$\boldsymbol{\pi}_a \mid \mathbf{y} \sim \mathrm{Dir}(\boldsymbol{\beta}_a + \mathbf{q}_a), \tag{17}$$

where $q_{a,b}$ is the number of times that a chord has an onset at the a-th position in a measure and has a duration of b time units in the training data.

5) Bayesian Learning of Pitch Markov Model: Given chord symbols $\mathbf{z}$, onset positions $\mathbf{y}$, and melody pitches $\mathbf{x}$, the posterior distribution of $\boldsymbol{\tau}$ can be analytically calculated as follows:

$$\boldsymbol{\tau}^c_a \mid \mathbf{z}, \mathbf{y}, \mathbf{x} \sim \mathrm{Dir}(\gamma^c_a + \mathbf{r}^c_a), \tag{18}$$

where $r^c(a,b)$ is the number of times that pitch a transits to pitch b in the region of chord symbol c in the training data.

IV. Melody Harmonization

This section explains how to generate chord sequences for a given melody based on the proposed model.

A. Problem Specification

Given a melody with pitches $\mathbf{x}$ and onset positions $\mathbf{o}$ and model parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\lambda}, \boldsymbol{\pi}, \boldsymbol{\tau}\}$, our goal is to estimate an appropriate number of chords with symbols $\mathbf{z}$ and onset positions $\mathbf{y}$, where a latent tree $\mathbf{t}$ should also be estimated. In this paper, we calculate the expectations of $\boldsymbol{\Theta}$ under the posterior distributions given by Eqs. (12), (13), (14), (17), and (18) and use them as the point estimates of $\boldsymbol{\Theta}$.

To draw samples of $\mathbf{t}$, $\mathbf{z}$, and $\mathbf{y}$ from the posterior distribution $p(\mathbf{t}, \mathbf{z}, \mathbf{y} | \mathbf{x}, \mathbf{o}, \boldsymbol{\Theta})$, we propose a Metropolis-Hastings (MH) sampler with four types of proposals:

- **Global update:** Update $\mathbf{z}$ and $\mathbf{t}$ by using a stochastic or deterministic method while keeping the number of chords and $\mathbf{y}$ unchanged.
- **Chord split:** Update $\mathbf{t}$, $\mathbf{z}$, and $\mathbf{y}$ by choosing one of the chords and split it into two adjacent chords.
- **Chord merge:** Update $\mathbf{t}$, $\mathbf{z}$, and $\mathbf{y}$ by choosing two adjacent chords derived from a node of $\mathbf{t}$ and merge them into a single chord.
- **Rhythm update:** Update $\mathbf{y}$ by choosing a chord n and move its onset position y_n back or forth.

One of these operations is randomly selected and a new sample $s^* = (\mathbf{t}^*, \mathbf{z}^*, \mathbf{y}^*)$ is proposed by referring to a current sample $s = (\mathbf{t}, \mathbf{z}, \mathbf{y})$. The acceptance ratio of s^* is given by

$$g(s^*, s) = \min\left\{1, \frac{p(s^*)\bar{p}(s|s^*)}{p(s)\bar{p}(s^*|s)}\right\}, \tag{19}$$

where $p(s) = p(\mathbf{t}, \mathbf{z}, \mathbf{y}, \mathbf{x}, \mathbf{o} | \boldsymbol{\Theta})$ is the complete joint probability of s based on the proposed model and $\bar{p}(s^*|s)$ is a proposal distribution. If the proposal is rejected, $\mathbf{t}$, $\mathbf{z}$, and $\mathbf{y}$ are not updated. In the global update, a proposed chord sequence is always accepted. This process is iterated until the posterior probability reaches a plateau. In the history of generated samples, we use a sample that maximizes the posterior distribution $p(\mathbf{t}, \mathbf{z}, \mathbf{y} | \mathbf{x}, \mathbf{o}, \boldsymbol{\Theta})$ as the final estimate of chords.

B. Global Update

We propose two methods for jointly updating chord symbols $\mathbf{z}$ and a latent tree $\mathbf{t}$ while fixing the chord onset positions $\mathbf{y}$. One is to use Gibbs sampling for stochastically drawing $\mathbf{z}$ and $\mathbf{t}$ from the posterior distribution $p(\mathbf{z}, \mathbf{t} | \mathbf{y}, \mathbf{x}, \mathbf{o}, \boldsymbol{\Theta})$. The other is to use a Viterbi algorithm for deterministically estimating $\mathbf{z}$ and $\mathbf{t}$ that maximize $p(\mathbf{z}, \mathbf{t} | \mathbf{y}, \mathbf{x}, \mathbf{o}, \boldsymbol{\Theta})$.

The stochastic method is similar to the inside filtering-outside sampling algorithm described in Section III-C1 except that both $\mathbf{t}$ and $\mathbf{z}$ are estimated. The inside messages are calculated recursively from the terminal symbols $\mathbf{z}$ to the start symbol S as follows:

$$p^A_{n,n} = \lambda_A \sum_{z_n \in \Sigma} \eta(A \rightarrow z_n)\ p(\mathbf{x}_n | z_n), \tag{20}$$

$$p^A_{n,n+k} = (1 - \lambda_A) \sum_{\substack{B,C \in V \\ 1 \le l \le k}} \theta(A \rightarrow BC) p^B_{n,n+l-1} p^C_{n+l,n+k}, \tag{21}$$

where $p(\mathbf{x}_n | z_n)$ is the probability that pitches $\mathbf{x}_n$ are generated from chord z_n and is given by

$$p(\mathbf{x}_n | z_n) = \prod_{i=1}^{I_n} \tau^{z_n}(x_{n,i-1}, x_{n,i}), \tag{22}$$

where $x_{n,0}$ is interpreted as $x_{n-1,I_{n-1}}$. A latent tree $\mathbf{t}$ is obtained by recursively sampling paths from the start symbol S to the leaf nodes as in the outside-sampling algorithm. Each chord symbol z_n is then sampled as follows:

$$p(z_n) \propto \eta(t_{n:n} \rightarrow z)\ p(\mathbf{x}_n | z_n). \tag{23}$$

In the deterministic method, the sum operations in Eqs. (20) and (21) are replaced with max operations. Instead of using the outside sampling, a latent tree $\mathbf{t}$ is obtained by recursively back-tracking the most likely paths from the start symbol S to the chord symbols.

C. Chord Split and Merge

A chord is split or adjacent chords are merged by considering both a latent tree $\mathbf{t}$ and a melody. Tree $\mathbf{t}$ is locally updated by these operations (Fig. 5) and the split operation has an inverse relationship with the merge operation.

In the split operation, a sample s^* is given by splitting a randomly selected chord z_n into z^L and z^R, estimating a new onset position $y^* \in [y_n + 1, y_{n+1} - 1]$, and splitting a non-terminal symbol $t_{n:n}$ into two non-terminal symbols t^L and t^R. The proposal distribution is thus given by

$$\bar{p}(s^*|s) = \begin{cases} \frac{\theta(t_{n:n} \rightarrow t^L t^R)\eta(t^L \rightarrow z^L)\eta(t^R \rightarrow z^R)}{N(y_{n+1} - y_n - 1)} & y_{n+1} \ge y_n + 1, \\ 0 & \text{otherwise}. \end{cases} \tag{24}$$

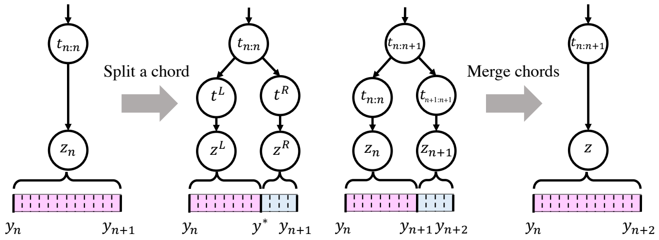


Fig. 5. Chord split and merge operations.

In the merge operation, a sample s is given by merging a randomly selected pair of adjacent chords z^L and z^R into z according to the probability $\eta(t_{n:n} \rightarrow z)$. Thus we have

$$\bar{p}(s|s^*) = \frac{\eta(t_{n:n} \rightarrow z)}{\#\text{MergeableNodes}(s^*)}, \tag{25}$$

where #MergeableNodes(s^*) is the number of pairs of adjacent chords that can be merged into s^*, *i.e.*, those chords forming a subtree with two leaves. The inverse proposal distribution for the split operation is the same as the proposal distribution for the merge operation and vice versa. In addition, we have

$$\begin{aligned} \frac{p(s^*)}{p(s)} &= (1 - \lambda_{t_{n:n}})\theta(t_{n:n} \rightarrow t^L t^R) \\ &\quad \cdot \frac{\lambda_{t^L}\eta(t^L \rightarrow z^L)\lambda_{t^R}\eta(t^R \rightarrow z^R)}{\lambda_{t_{n:n}}\ \eta(t_{n:n} \rightarrow z_n)} \\ &\quad \cdot \frac{p(\mathbf{x}^L|z^L)p(\mathbf{x}^R|z^R)p(y^*|y_n)p(y_{n+1}|y^*)}{p(\mathbf{x}_n|z_n)p(y_{n+1}|y_n)}, \end{aligned} \tag{26}$$

where $\mathbf{x}_n^L$ and $\mathbf{x}_n^R$ are sequences of pitches obtained by splitting $\mathbf{x}_n$ at a position y^*.

In the split operation, the acceptance ratio of s^* given by Eq. (19) is calculated by using Eqs. (24), (25), and (26). In the merge operation, the acceptance ratio of s^* given by Eq. (19) is calculated by exchanging s and s^* in Eqs. (24), (25), and (26). Through the split and merge operations, the number of chords N is optimized stochastically.

D. Rhythm Update

Chord onset positions $\mathbf{y}$ are sampled from the conditional posterior distribution $p(\mathbf{y}|\mathbf{t}, \mathbf{z}, \mathbf{x}, \mathbf{o}, \mathbf{\Theta})$. A new sample s^* is given by moving a randomly selected onset position y_n to a position $y_n^* \in [y_{n-1}+1, y_{n+1}-1]$. The proposal distribution $\bar{p}(s^*|s)$ and its inverse version $\bar{p}(s|s^*)$ are given by

$$\bar{p}(s^*|s) = \bar{p}(s|s^*) = \frac{1}{N-1}\frac{1}{y_{n+1} - y_{n-1} - 1}. \tag{27}$$

The likelihood ratio is given by

$$\frac{p(s^*)}{p(s)} = \frac{p(\mathbf{x}_{n-1}^*|z_{n-1})p(\mathbf{x}_n^*|z_n)p(y_n^*|y_{n-1})p(y_{n+1}|y_n^*)}{p(\mathbf{x}_{n-1}|z_{n-1})p(\mathbf{x}_n|z_n)p(y_n|y_{n-1})p(y_{n+1}|y_n)}, \tag{28}$$

where $\mathbf{x}_{n-1}^*$ and $\mathbf{x}_n^*$ are the sequences of pitches in the regions of chords $n-1$ and n with the new onset position y_n^*. The acceptance ratio of s^* given by Eq. (19) is calculated by using Eqs. (27) and (28).

V. Evaluation

This section reports three objective experiments for evaluating the unified model of chords and melody notes and another experiment for evaluating the melody harmonization method based on the unified model.

A. Experimental Conditions

To train the PCFG model, 399 chord sequences corresponding to musical sections of seven or eight measures (*e.g.*, verse, bridge, and chorus sections) were extracted from the Billboard data [37] using its section annotations. Due to the limited amount of available data, we limit the vocabulary of chord symbols used for the experiments to the set of major and minor triads. The vocabulary of chord symbols consists of the combinations of the 12 root notes {C, C#, ..., B} and the two chord types {major, minor}, and other chord types are reduced to triads according to the root, third, and fifth notes (diminished and augmented triads are discarded in the analysis). We manually made tree structure annotations data[2]. In Eqs. (6) and (7), the elements of hyperparameter $\boldsymbol{\nu}$ were all set to 1.0 and those of $\boldsymbol{\xi}$, $\boldsymbol{\zeta}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ were all set to 0.1.

To train the parameters of an HMM used for initializing the leaf nodes of a tree $\mathbf{t}$, a set (called J-pop data) of chord sequences of 4,872 Japanese popular songs obtained from a public web page[3] was used. More specifically, an HMM with four latent states was trained such that these states correspond to the three harmonic functions (tonic, dominant, subdominant) and an extra function as in [22]. Fig. 6 shows the emission probabilities of the HMM. To train the metrical Markov model described in Section III-B2 and the Markov model described in Section III-B3, 9,902 pairs of melodies and chord sequences were extracted from 194 pieces of popular music included in the Rock Corpus [38].

As shown in Fig. 4, we tested the following four learning configurations of the PCFG model:

(i) **US-PCFG**: Unsupervised learning (Section III-C1).
(ii) **SS-PCFG:** Semi-supervised learning based on tree structure annotations (Section III-C2).
(iii) **US-HMM-PCFG**: Unsupervised learning with HMM-based initialization (Sections III-C1 and III-C3).
(iv) **SS-HMM-PCFG**: Semi-supervised learning based on tree structure annotations with HMM-based initialization (Sections III-C2 and III-C3).

All data (Billboard data, J-pop data, and Rock Corpus) were transposed to the C major or C minor key (we used the local key information when it is given, to deal with possible modulations within each piece). To put emphasis on the Markov model of a melody, each element of the trained $\boldsymbol{\tau}$ was squared and then normalized.

B. Statistical Chord Modeling

We investigated the performance of the PCFG model in predicting chord sequences, for varying numbers of non-terminal symbols K. The number K was changed from 1

[2] The data is available at http://sap.ist.i.kyoto-u.ac.jp/members/tsushima/chord_tree_annotation/

[3] J-Total Music: http://music.j-total.net

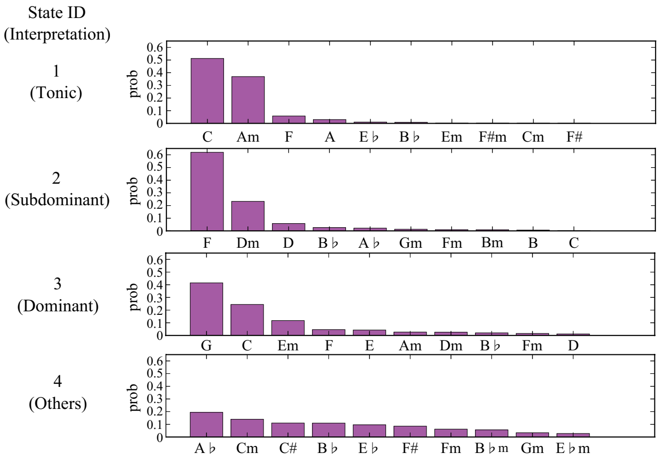


Fig. 6. The emission probabilities of an HMM. Ten chord symbols with highest emission probabilities are shown for each state.

to 25 in the framework of five-fold cross validation. As an evaluation measure, we calculated the perplexity for an unseen chord sequence $\mathbf{z}$ as follows:

$$\mathcal{P}_{\mathrm{chord}} = \exp\left(-\frac{1}{N}\ln\frac{p(\mathbf{z}|\boldsymbol{\lambda},\boldsymbol{\theta},\boldsymbol{\eta})}{\sum_{\mathbf{z}':|\mathbf{z}'|=N} p(\mathbf{z}'|\boldsymbol{\lambda},\boldsymbol{\theta},\boldsymbol{\eta})}\right), \qquad (29)$$

where $p(\mathbf{z}|\boldsymbol{\lambda},\boldsymbol{\theta},\boldsymbol{\eta}) = p^S_{1,N}$ is the marginal probability obtained by using the inside algorithm given by Eqs. (9) and (10).

As shown in Fig. 7, US-HMM-PCFG achieved the lowest (best) perplexities and outperformed US-PCFG for almost all K. This indicates that the HMM-based initialization method improves the generalization capability of the PCFG model for unseen chords. Interestingly, SS-HMM-PCFG underperformed US-HMM-PCFG. This means that the semi-supervised learning method using tree structure annotations does not contribute to chord prediction. This is presumably because there are many possible structural interpretations for chord sequences and the unsupervised learning method is better at dealing with such ambiguity in a probabilistic manner. Nonetheless, SS-HMM-PCFG can be considered useful for *uniquely* estimating the most likely tree behind chords (Section V-D).

The improvement of the test-set perplexity was saturated before K reached 20. Considering the computational cost, the optimal values of K were estimated as 12, 17, 16, and 18 for learning methods (i)–(iv), respectively, which were used for the following evaluations.

C. Statistical Melody Modeling

We investigated the performance of the unified model in predicting melodies by using 172 melodies extracted from the RWC music database [39]. These melodies were disjoint with the datasets (Billboard data, J-pop data, and Rock Corpus) used for training the unified model. In this experiment, we assumed that the chord onset positions $\mathbf{y}$ were fixed to bar lines, to enable analytically marginalizing out chord symbols and the tree in the calculation of the melody perplexity defined below. As an evaluation measure, we calculated the perplexity for an unseen melody $(\mathbf{x}, \mathbf{o})$ (per note) as follows:

$$\mathcal{P}_{\mathrm{melody}} = \exp\left(-\frac{1}{I}\ln\frac{p(\mathbf{x},\mathbf{o}|\mathbf{y},\boldsymbol{\lambda},\boldsymbol{\theta},\boldsymbol{\eta})}{\sum_{\mathbf{z}:|\mathbf{z}|=N} p(\mathbf{z}|\boldsymbol{\lambda},\boldsymbol{\theta},\boldsymbol{\eta})}\right), \qquad (30)$$

where $p(\mathbf{x},\mathbf{o}|\mathbf{y},\boldsymbol{\lambda},\boldsymbol{\theta},\boldsymbol{\eta}) = p^S_{1,N}$ is the marginal probability obtained by using the inside algorithm given by Eqs. (20) and (21). Note that chord symbols $\mathbf{z}$ and a tree $\mathbf{t}$ are both marginalized out.

For comparison, we tested a measure-wise HMM that represents chords and melody notes as latent and observed variables, respectively. To conduct a fair comparison with the PCFG model, chords were allowed to change only at bar lines in this model. As a generative process, one chord for each measure is generated according to a Markov model, and melody notes in each measure is generated by a Markov model conditionally dependent on the generated chord as in Eq. (4). The transition probabilities of chords were learned from the Billboard data. Given a melody, the most likely chord sequence was estimated using the Viterbi algorithm.

As shown in Fig. 8, the results of melody modeling were consistent with those of chord modeling shown in Section V-B. US-HMM-PCFG attained the lowest perplexity and US-PCFG, US-HMM-PCFG, and SS-HMM-PCFG outperformed the conventional HMM (6.09). The HMM-based initialization of the PCFG model improved the ability of melody prediction.

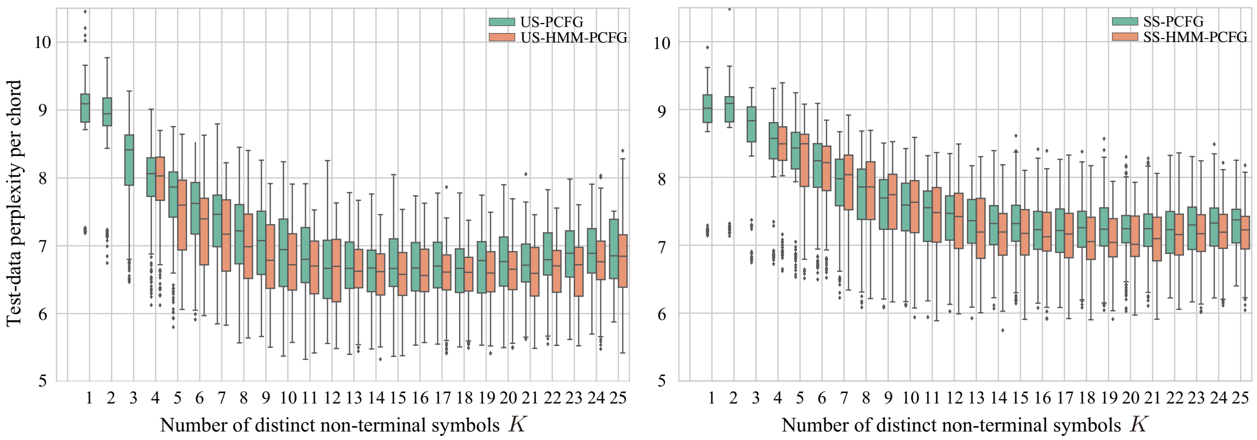
D. Tree-Structured Parsing

We investigated the performance of the unified model in tree-structured parsing of chord sequences. 5-fold cross validation on the Billboard dataset [37] was conducted and the unsupervised learning method (Section III-C1) was compared with the semi-supervised learning method (Section III-C2). As an evaluation measure, we calculated the tree edit distance (TED) [40] between an estimated tree and the ground-truth tree. Since the non-terminal symbols of the ground-truth tree are not given, only the estimated tree shape was evaluated.

As shown in Fig. 9, the semi-supervised learning method turned out to be effective for improving the performance of tree-structured parsing. For the unsupervised learning methods, the HMM-based initialization also led to better tree-structured parsing results. Interestingly, applying both the semi-supervised learning and the HMM-based initialization of the PCFG model slightly degraded the performance. When the leaf nodes of a tree were guided to represent harmonic functions, the production rules used in a higher level were often disjointed from those used in a lower level. This might prevent the model from inducing a globally consistent tree from a chord sequence. Overall, we confirmed the efficacy of the proposed learning techniques for improving the similarity between human structural annotations of chord sequences and those generated by the PCFG model.

E. Melody Harmonization

We investigated the performance of the proposed melody harmonization method based on the unified probabilistic model



(a) (b)

Fig. 7. Test-set perplexities per chord symbol obtained by the four different learning configurations of the PCFG model: (a) US-PCFG and US-HMM-PCFG, (b) SS-PCFG and SS-HMM-PCFG. The band and boxes indicate the median and interquartile range (IQR), and the whiskers indicate the 1.5 times wider range as IQR. Points outside this range are identified as outliers.

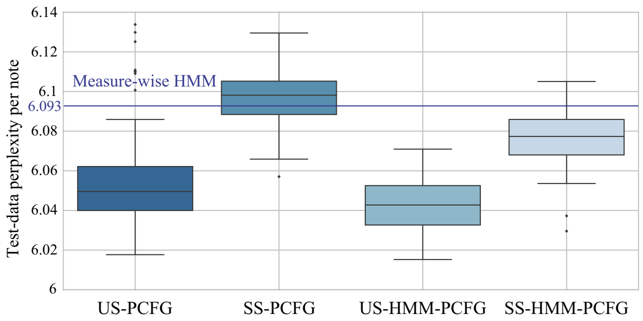


Fig. 8. Test-set perplexities per melody note. Indicators in the box plots are the same as those in Fig. 7.

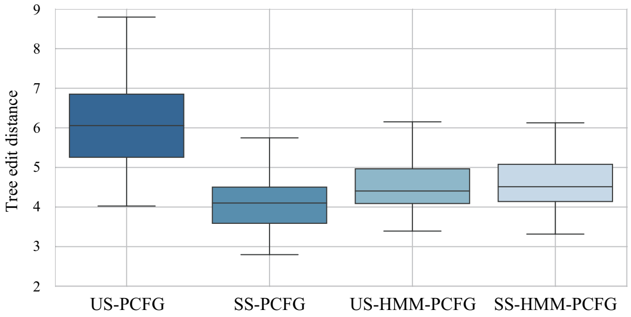


Fig. 9. Tree edit distances between estimated and ground-truth trees. Indicators in the box plots are the same as those in Fig. 7.

of a chord sequence and a melody. For evaluation, we extracted 172 pairs of human-composed melodies and chord sequences from the RWC music database [39]. We tested eight configurations by combining the four learning configurations (i)–(iv) with the sampling method and the Viterbi algorithm described in Section IV-B. For comparison, we tested a standard HMM-based method, similar to the model of [7], that represents chord transitions at the 16th-note level. We also tested a rule-based method implemented in the Melisma Music Analyzer [41] that generates a sequence of chord root notes for a given melody.

For the proposed method, we ran the MCMC sampler to generate a sufficient number ($= 5000$) of chord sequences for a given melody. From the generated chord sequences, one with the highest posterior probability was chosen as the final chord sequence. We also used a set of five samples of chord sequences with the highest posterior probabilities in the calculation of the mean reciprocal rank (MRR) explained later. Since the learning methods of the unified model (i)–(iv) and the proposed melody harmonization method are all based on MCMC sampling, we took the average score of each measure over 500 trials, *i.e.*, we ran the harmonization method five times using 100 different parameter sets of the PCFG model. For the HMM-based method, we similarly sampled a sufficient number ($= 1000$) of chord sequences using the forward filtering-backward sampling algorithm [43] and chose five samples with the highest posterior probabilities for the calculation of the MRR. For the other metrics, we used the most probable chord sequence obtained by the Viterbi algorithm.

The methods were evaluated by comparing the generated chord sequences with the original human-composed chord sequence. To evaluate the local correctness of generated chord sequences, we calculated the accuracy by comparing the generated sequence with the highest posterior probability with the original sequence at the 16th-note level. We also calculated the tonal pitch step distance [42], which measures the dissimilarity between two chord sequences based on a cognitive model of tonality (see [42] for details). We also calculated the mean reciprocal rank (MRR) by comparing the five chord sequences with the original sequence at the 16th-note level. To evaluate the global correctness of generated chord sequences, we calculated the Levenshtein edit distance between the generated sequence and the original sequence. To compare the proposed method with the rule-based method, we also calculated the Levenshtein distance between the generated and original sequences of root notes by ignoring the chord types.

As shown in Fig. 10, the Viterbi algorithm consistently outperformed the sampling method and US-HMM-PCFG per-

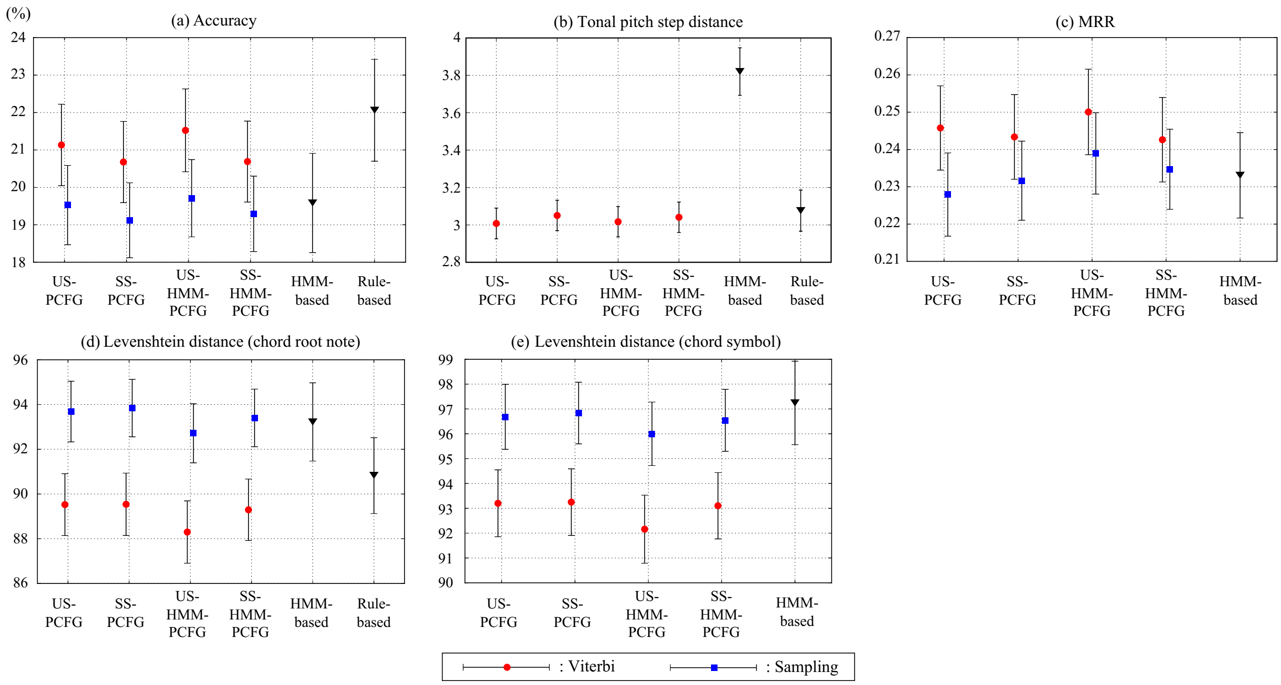


Fig. 10. Evaluation results of melody harmonization: (a) accuracy, (b) tonal pitch step distance [42], (c) MRR, (d) Levenshtein distance for chord root notes, and (e) Levenshtein distance for chord symbols. For accuracy and MRR, the higher, the better. For Levenshtein distance, the lower, the better. We plot the piece-wise mean and standard error corresponding to a 68% confidence interval.

formed best in terms of the MRR and the Levenshtein distance. In terms of accuracy, the proposed method outperformed the HMM-based method (19.6%), but underperformed the rule-based method (22.1%). In terms of the MRR, the proposed method outperformed the HMM-based method (0.233). As for the tonal pitch step distance and Levenshtein distance for root notes, the proposed method with the Viterbi algorithm outperformed both the HMM-based method and the rule-based method. In the case of the Levenshtein distance for chord symbols, the proposed method significantly outperformed the HMM-based method (97.2). These results show that the proposed method was inferior to the rule-based method in terms of the accuracy, but outperformed it in terms of the other compared metrics. A possible reason is that while the PCFG is able to capture the global structure with a derivation tree, it cannot explicitly model the sequential dependency of chords, unlike Markov models.

F. Example Results

Fig. 11 shows examples of chord sequences generated by the proposed and conventional methods. The numbers of distinct non-terminal symbols of US-PCFG and SS-HMM-PCFG were set to 12 and 17, respectively. The HMM-based method often generated a sequence consisting of a very few chord symbols. The rule-based method generated a chord sequence consisting of a reasonable number of symbols but could not generate musically natural rhythms.

In contrast, the proposed method (US-PCFG and SS-HMM-PCFG) were able to successfully generate a chord sequence with a reasonable number of symbols with natural rhythms. This demonstrates the effectiveness of the metrical Markov model of chord onset positions. In US-PCFG, leaf nodes above different chord symbols were given different labels, whereas in SS-HMM-PCFG, the leaf nodes above C-major chord and A-minor chord, which are usually categorized as tonic chords, were given the same label S_1. In addition, in the latter case, the generated chord sequence had cadences and chord rhythms that were similar to those of the ground truth. We observed that the two models had a similar tendency for other melodies as well (see online supplemental material[4]). This indicates that the SS-HMM-PCFG can successfully reflect syntactic functions and hierarchical structure of chords in the generated results.

G. Open Problems

The proposed melody harmonization method tends to generate conservative chord symbols (*e.g.*, C major and G major) that frequently appear in the training data because they tend to maximize the posterior probabilities of chords for given melodies. To increase the diversity and musical attractiveness of the generated chord sequences, it would be effective to incorporate prior knowledge about chord patterns (idioms and phrases) that frequently appear in actual musical pieces into our MCMC sampler. Another solution is to replace the PCFG model that splits a non-terminal node into two child nodes with a tree-substitution grammar (TSG) model that splits a non-terminal node into two variable-depth subtrees. This enables us to automatically discover tree-structured patterns frequently found in chord sequences. We also plan to replace the generative model of a melody (Markov model) with a deep

[4] https://pcfgharmonization.github.io/

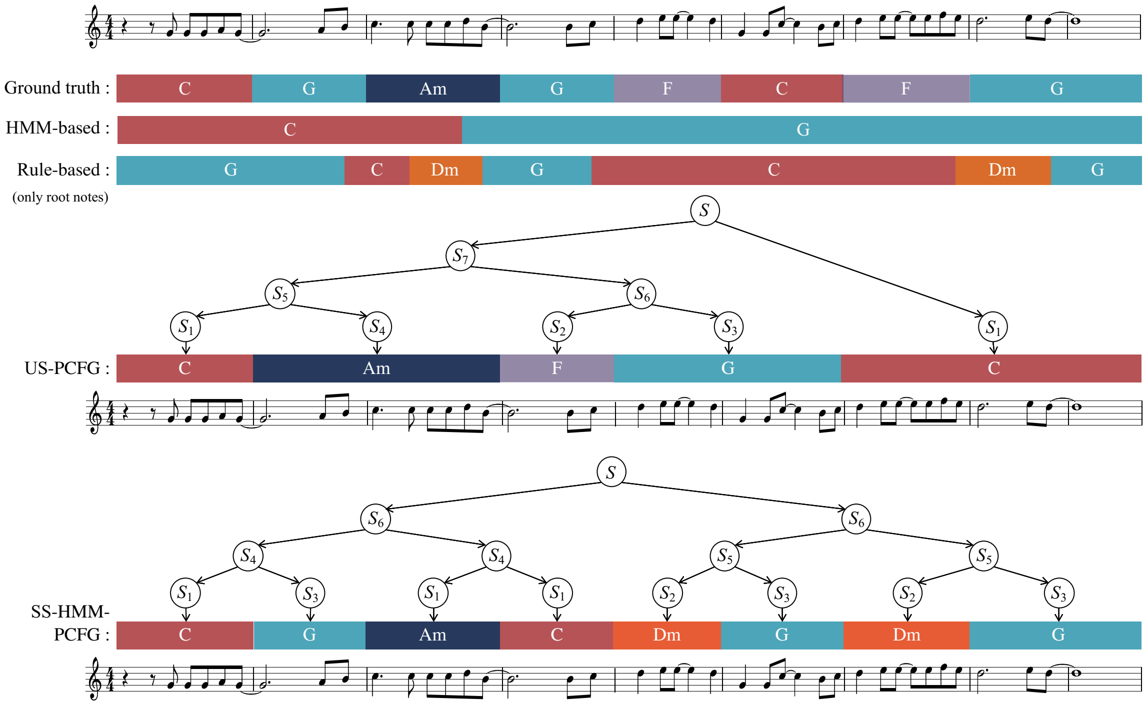


Fig. 11. Examples of generated chord sequences. From top to bottom, the input melody, the human-composed chord sequence, and chord sequences generated by the HMM-based method, the rule-based method, US-PCFG, and SS-HMM-PCFG are shown.

generative model (*e.g.*, recurrent neural network), which is effective for learning complex dependency in time series.

While our method can capture syntactic functions and the local structure behind chords, it is still weak in capturing global structure such as phrase structure. A possible solution is to use the combinatory categorial grammar (CCG) [44] for modeling chord sequences. A CCG can explicitly describe the global syntactic structure (*e.g.*, phrase, clause) by using symbols constructed from the combination of simple categories of words (*e.g.*, “noun”, “verb”). By inferring the CCG from the data of chord sequences, it would be possible to directly analyze the phrase structure in chord sequences. It is also important to improve the naturalness of chord rhythms. Since chord rhythms depend on rhythms of melody notes, we plan to formulate a model of melody note rhythms and integrate it with our unified model.

Acknowledgement

We are grateful to Ryo Nishikimi for helping the computer program implementation of our method.

VI. Conclusion

This paper has presented a Bayesian melody harmonization method that aims to generate a chord sequence with natural rhythms for a given melody[5]. This method is based on a unified generative model of a chord sequence and a melody involving a metrical Markov model describing chord rhythms and a PCFG model describing the tree structure behind chords. Experimental results showed that the proposed method outperformed the conventional HMM-based method in terms of accuracy of generated chord sequences. We also confirmed that the semi-supervised learning of the PCFG model is superior to the completely unsupervised learning, in terms of the ability to predict tree structures behind chords, and that the HMM-based initialization of the PCFG model improved the performance of melody harmonization.

Our next direction is to construct an interactive music composition/arrangement system. Interactive systems are useful for those people who want to incorporate their preference in the generated music and can go beyond the reach of fully automated systems. The ability of the proposed model to learn and represent the grammatical structure of music in a similar way as humans do is essential for enhancing the directability of the system, as partly shown in [13]. Through large-scale user studies we plan to reveal human’s music creation process from the computational perspective.

[5] The source code of our melody harmonization method is available in the online supplementary page (https://pcfgharmonization.github.io/).

References

[1] K. Ebcioğlu, “An expert system for harmonizing four-part chorales,” *Computer Music Journal*, vol. 12, no. 3, pp. 43–51, 1988.
[2] G. Papadopoulos and G. Wiggins, “AI methods for algorithmic composition: A survey, a critical view and future prospects,” in *AISB Symposium on Musical Creativity*, 1999, pp. 110–117.
[3] M. Towsey, A. Brown, S. Wright, and J. Diederich, “Towards melodic extension using genetic algorithms,” *Educational Technology & Society*, vol. 4, no. 2, pp. 54–65, 2001.
[4] M. Allan and C. Williams, “Harmonising chorales by probabilistic inference,” in *NIPS*, 2005, pp. 25–32.
[5] J. F. Paiement, D. Eck, and S. Bengio, “Probabilistic melodic harmonization,” in *CSCSI*, 2006, pp. 218–229.
[6] C. H. Chuan and E. Chew, “A hybrid system for automatic generation of style-specific accompaniment,” in *IJWCC*, 2007, pp. 57–64.
[7] I. Simon, D. Morris, and S. Basu, “Mysong: automatic accompaniment generation for vocal melodies,” in *SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2008, pp. 725–734.

[8] R. D. Prisco and R. Zaccagnino, "An evolutionary music composer algorithm for bass harmonization," in *Applications of Evolutionary Computing*. Springer, 2009, pp. 567–572.

[9] R. D. Prisco, A. Eletto, A. Torre, and R. Zaccagnino, "A neural network for bass functional harmonization," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2010, pp. 351–360.

[10] R. Groves, "Automatic harmonization using a hidden semi-Markov model," in *AIIDE*, 2013, pp. 48–54.

[11] S. A. Raczyński, S. Fukayama, and E. Vincent, "Melody harmonization with interpolated probabilistic models," *Journal of New Music Research*, vol. 42, no. 3, pp. 223–235, 2013.

[12] H. Tsushima, E. Nakamura, K. Itoyama, and K. Yoshii, "Function- and rhythm-aware melody harmonization based on tree-structured parsing and split-merge sampling of chord sequences," in *ISMIR*, 2017, pp. 502–508.

[13] ——, "Interative arrangement of chords and melodies based on a tree-structured generative model," in *ISMIR*, 2017, pp. 502–508.

[14] G. Hadjeres and F. Pachet, "DeepBach: A steerable model for Bach chorales generation," in *ICML*, 2017, pp. 1362–1371.

[15] A. Cadwallader and D. Gagné, *Analysis of Tonal Music: A Schenkerian Approach (3rd ed.).* Oxford University Press, 2011.

[16] H. Riemann, *Harmony Simplified: Or the Theory of the Tonal Functions of Chords (2nd ed.).* Augener, 1986.

[17] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*. MIT press, 1985.

[18] M. J. Steedman, "A generative grammar for jazz chord sequence," *Music Perception*, vol. 2, no. 1, pp. 52–77, 1984.

[19] M. Rohrmeier, "Mathematical and computational approaches to music theory, analysis, composition and performance," *Journal of Mathematics and Music*, vol. 5, no. 1, pp. 35–53, 2011.

[20] W. B. De Haas, J. P. Magalhães, F. Wiering, and R. C. Veltkamp, "Automatic functional harmonic analysis," *Computer Music Journal*, vol. 37, no. 4, pp. 37–53, 2013.

[21] D. Quick, "Learning production probabilities for musical grammars," *Journal of New Music Research*, vol. 45, no. 4, pp. 295–313, 2016.

[22] H. Tsushima, E. Nakamura, K. Itoyama, and K. Yoshii, "Generative statistical models with self-emergent grammar of chord sequences," *Journal of New Music Research*, 2018.

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

[24] M. Hamanaka, M. Goto, H. Asoh, and N. Otsu, "A learning-based quantization: Unsupervised estimation of the model parameters," in *ICMC*, 2003, pp. 369–372.

[25] H. Lim, S. Rhyu, and K. Lee, "Chord generation from symbolic melody using BLSTM networks," in *Proc. ISMIR*, 2017, pp. 621–627.

[26] C. Ames, "The Markov process as a compositional model: a survey and tutorial," *Leonardo*, vol. 22, no. 2, pp. 175–187, 1989.

[27] F. Pachet, R. Roy, and G. Barbieri, "Finite-length markov processes with constraints," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[28] M. Hamanaka, K. Hirata, and S. Tojo, "Implementing 'A generative theory of tonal music'," *Journal of New Music Research*, vol. 35, no. 4, pp. 249–277, 2006.

[29] ——, "Musical structural analysis database based on GTTM," in *ISMIR*, 2014, pp. 325–330.

[30] E. Nakamura, M. Hamanaka, K. Hirata, and K. Yoshii, "Tree-structured probabilistic model of monophonic written music based on the generative theory of tonal music," in *IEEE ICASSP*, 2016, pp. 276–280.

[31] R. Scholz, E. Vincent, and F. Bimbot, "Robust modeling of musical chord sequences using probablistic N-grams," in *IEEE ICASSP*, 2009, pp. 53–56.

[32] K. Yoshii and M. Goto, "A vocabulary-free infinity-gram model for nonparametric bayesian chord progression analysis." in *ISMIR*, 2011, pp. 645–650.

[33] J. F. Paiement, D. Eck, and S. Bengio, "A probabilistic model for chord progressions," in *ISMIR*, 2005, pp. 312–319.

[34] S. Fukayama, K. Yoshii, and M. Goto, "Chord-sequence-factory: A chord arrangement system modifying factorized chord sequence probabilities," in *ISMIR*, 2013, pp. 457–462.

[35] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *ACL*, 1996, pp. 310–318.

[36] M. Johnson, T. L. Griffiths, and S. Goldwater, "Bayesian inference for PCFGs via Markov chain Monte Carlo," in *NAACL-HLT*, 2007, pp. 139–146.

[37] J. B. L. Smith, J. A. Burgoyne, I. Fujinaga, D. D. Roure, and J. S. Downie, "Design and creation of a large-scale database of structural annotations," in *ISMIR*, 2011, pp. 555–560.

[38] T. D. Clercq and D. Temperley, "A corpus analysis of rock harmony," *Popular Music*, vol. 30, no. 01, pp. 47–70, 2011.

[39] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical and jazz music databases." in *ISMIR*, 2002, pp. 287–288.

[40] K. Zhang and D. Shasha, "Simple fast algorithms for the editing distance between trees and related problems," *SIAM journal on computing*, vol. 18, no. 6, pp. 1245–1262, 1989.

[41] D. Temperley and D. Sleator, "The Melisma Music Analyzer," 2001, http://www.link.cs.cmu.edu/melisma/.

[42] W. B. de Haas, F. Wiering, and R. C. Veltkamp, "A geometrical distance measure for determining the similarity of musical harmony," *International Journal of Multimedia Information Retrieval*, vol. 2, no. 3, pp. 189–202, 2013.

[43] J. Van Gael, Y. Saatci, Y. W. Teh, and Z. Ghahramani, "Beam sampling for the infinite hidden Markov model," in *Proc. ICMC*, 2008, pp. 1088–1095.

[44] M. Steedman and J. Baldridge, "Combinatory categorial grammar," *Non-Transformational Syntax: Formal and Explicit Models of Grammar. Wiley-Blackwell*, 2011.

Hiroaki Tsushima He received a masters degree in Informatics from Kyoto University in 2019. He was a member of Speech and Audio Processing Group, Graduate School of Informatics, Kyoto University. His research interests include automatic music composition and machine learning.

Eita Nakamura Eita Nakamura He received a Ph.D. degree in physics from the University of Tokyo in 2012. He has been a post-doctoral researcher at the National Institute of Informatics, Meiji University, and Kyoto University. He is currently an Assistant Professor at the Hakubi Center for Advanced Research and Graduate School of Informatics, Kyoto University. His research interests include music modelling and analysis, music information processing and statistical machine learning.

Kazuyoshi Yoshii He received the Ph.D. degree in informatics from Kyoto University, Japan, in 2008. He is currently a Senior Lecturer at Kyoto University. His research interests include music signal processing and machine learning. He is a Member of the Information Processing Society of Japan and Institute of Electronics, Information, and Communication Engineers.