Probabilistic Sequential Patterns for Singing Transcription

Eita Nakamura*, Ryo Nishikimi*, Simon Dixon† and Kazuyoshi Yoshii*
* Kyoto University, Kyoto, Japan
E-mail: [enakamura,nishikimi,yoshii]@sap.ist.i.kyoto-u.ac.jp
† Queen Mary University of London, London, United Kingdom
E-mail: s.dixon@qmul.ac.uk

***Abstract*—Statistical models of musical scores play an important role in various tasks of music information processing. It has been an open problem to construct a score model incorporating global repetitive structure of note sequences, which is expected to be useful for music transcription and other tasks. Since repetitions can be described by a sparse distribution over note patterns (segments of music), a possible solution is to consider a Bayesian score model in which such a sparse distribution is first generated for each individual piece and then musical notes are generated in units of note patterns according to the distribution. However, straightforward construction is impractical due to the enormous number of possible note patterns. We propose a probabilistic model that represents a cluster of note patterns, instead of explicitly dealing with the set of all possible note patterns, to attain computational tractability. A score model is constructed as a mixture or a Markov model of such clusters, which is compatible with the above framework for describing repetitive structure. As a practical test to evaluate the potential of the model, we consider the problem of singing transcription from vocal f0 trajectories. Evaluation results show that our model achieves better predictive ability and transcription accuracies compared to the conventional Markov model, nearly reaching state-of-the-art performance.**

I. Introduction

Computational models of musical scores, sometimes referred to as music language models, play a vital role in various tasks of music information processing [1]–[6]. For automatic music transcription [7], for example, a musical score model is necessary to induce an output score to be an appropriate one that respects musical grammar, style of the target music, playability, etc. Conventional musical score models include variants of Markov models [8], hidden Markov models (HMMs) and probabilistic context-free grammar models [9], recurrent neural networks [10], and convolutional neural networks [11]. These models are typically learned supervisedly and the obtained *generic* score model is applied to transcription of all target musical pieces, in combination with an acoustic model and/or a performance model.

A musical piece commonly consists of groups of musical notes that are repeated several times, and detecting repeated note patterns has been a topic of computational music analysis [12]. Repetitive structure can complement local sequential dependence of notes, on which conventional score models have focused, and thus can be useful for transcription. This is because if the repetitive structure can be inferred correctly, one can recover part of the score or can account for timing deviations, acoustic variations, and other types of "noise" in performances. To realize this scenario, we need a score model that induces repetitions in generated scores. Since repetitive structure is a global feature, it is challenging to construct a computationally tractable model with such a property.

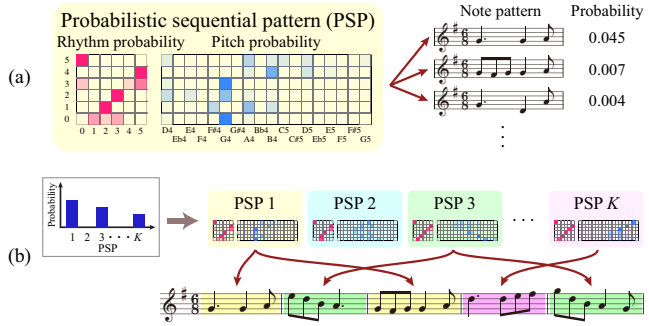


Fig. 1. The proposed sequential pattern model for musical scores. (a) A PSP generates note patterns for a bar based on a metrical Markov model (rhythm probability) and a distribution of pitches defined for each beat position (pitch probability). (b) A generative process of musical scores with multiple PSPs; for each bar a PSP is chosen according to a mixture/Markov model. See section III for details. (For clear illustration, the 6/8 time in this example is different from the 4/4 time considered in the main text.)

A framework for constructing a score model with repetitive structure has been proposed in [13]. The idea is to consider a distribution (or a Markov model) over a set of possible note patterns and implicitly represent repetitive structure with sparseness of the distribution. A note pattern is defined here as a subsequence of musical notes; if a piece consists of a small number of note patterns then it has repetitions and vice versa. The model is described as a Bayesian extension of a Markov model of note patterns where the Dirichlet parameters of the prior distributions are assumed to be small to induce repetitions. In addition, to deal with approximate repetitions (i.e. repetitions with modifications), which are common in music practice [14], an additional stochastic process of modifying patterns was incorporated. In this framework, an *individual* score model is learned unsupervisedly in the transcription process for each input signal, in contrast to the above supervised learning scheme. That study focused on monophonic music and only rhythms were modelled, and it has been shown that the framework is indeed effective for music transcription.

Although it is theoretically possible to extend this framework for more general forms of music, including pitches and multiple voices, the model becomes so large that computational tractability will be lost. First, with $\Omega \sim \mathcal{O}(10\text{–}10^2)$ unique pitches and ℓ notes, the number of possible pitch patterns is Ω^ℓ, which soon becomes intractable as ℓ increases. Second, by adding the process of modifying patterns, the complexity increases further: with Λ unique patterns one should consider Λ^2 possible ways of modification in general. Third, joint modelling of pitches and rhythms further increases the size of the state space.

In this study, we propose a new approach for modelling note patterns including pitches and rhythms that is compatible with the above framework for describing repetitive structure. To realize this, we treat a set of patterns related by modifications as a cluster, named probabilistic sequential pattern (PSP), without explicitly dealing with the set of all possible patterns (Fig. 1). A PSP is formulated so that it can naturally accommodate insertion, deletion, and substitution of notes as modifications of patterns. We construct score models based on a mixture (possibly with a Markov structure on the mixture weights) of such PSPs and its Bayesian extension. As a practical problem, we consider the problem of singing transcription of vocal f0 trajectories [15], [16].

The main contributions of this study are:

- Construction of computationally tractable score models based on note patterns; a Bayesian framework for describing approximate repetitive structure of musical scores.
- Our model achieves better predictive ability and transcription accuracies than the conventional Markov model.

An additional contribution is proposing a framework for integrating a note-level score model and a beat-level f0 model. Such a framework is nontrivial due to different time units in acoustic/f0 signals and musical scores and has not been realized in most previous studies; in some studies frame-level musical score models have been considered [17], [18], which cannot properly describe musical rhythms, and in other studies an unrealistic situation that the number of musical notes is known in advance has been assumed [19].

The rest of the paper is organized as follows. In the next section, we review a simple Markov model generating melodies (pitches and rhythms) and its extension for use in singing transcription. In section III, we formulate our PSP model and explain an inference method. Numerical experiments and evaluations are presented in section IV. The last section is dedicated to conclusions and discussion.

II. Melody Markov Model

Here we specify the problem of singing transcription considered in this study. A simple model for statistical singing transcription is presented as a baseline method.

A. Singing Transcription from F0 Trajectories

Singing transcription is a task of converting a singing voice signal into a musical score representation [19], [20]. Here, we consider a situation that a singing-voice f0 trajectory is given in advance as well as the information about the location of beats and bar onsets (downbeats) (Fig. 2). The former can be estimated by using an f0 estimation method for singing voices (e.g. [21]) and the latter can be estimated by a beat tracking method (e.g. [22]). In the following, time frames are indexed by $t \in \{1, \ldots, T\}$ and score times in units of 16th notes are indexed by $\tau \in \{\tau_s, \ldots, \tau_e\}$, where T represents the signal length and τ_s and τ_e respectively denote the first and the last beats. For each score time τ, its relative position to the bar onset is denoted by $[\tau] \in \{0, \ldots, B-1\}$ and called the beat position. The symbol B denotes the number of beats in each bar and it is fixed to $B = 16$ in what follows, as we only consider pieces in $4/4$ time for simplicity. Each score time τ is associated with a time frame denoted by $t(\tau)$.

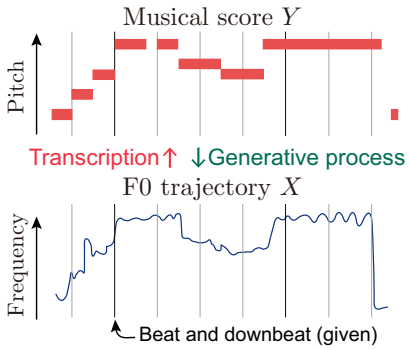


Fig. 2. Singing transcription from a vocal f0 trajectory and the corresponding generative process.

The input signal is a sequence of f0s denoted by $X = (x_t)_{t=1}^T$. The output is a musical score represented as a sequence of pitches and onset score times $Y = (p_n, \tau_n)_{n=1}^N$, where N is the number of notes and p_n is the n th note's pitch in units of semitones. Note that the number N is unknown in advance and must be estimated from the input signal.

In the generative modelling approach, we first construct a score model that yields the probability $P(Y)$ and combine it with an f0 model that yields $P(X|Y)$. The output score can be obtained as the one that maximizes the probability $P(Y|X) \propto P(X|Y)P(Y)$.

B. Score Model

As a minimal model, we treat pitches and rhythms independently and consider a Markov model for pitches and a metrical Markov model [23], [24] for onset score times. It is described with the initial and transition probabilities for pitches

$$P(p_1 = p) = \chi_p^{\mathrm{ini}}, \quad P(p_n = p \,|\, p_{n-1} = p') = \chi_{p'p} \qquad (1)$$

and those for onset score times

$$P(\tau_1 = \tau) = \psi_{[\tau]}^{\mathrm{ini}}, \quad P(\tau_n = \tau \,|\, \tau_{n-1} = \tau') = \psi_{[\tau'][\tau]}. \qquad (2)$$

In the transition probability matrix for onset score times, we have implicitly assumed that the maximum interval between adjacent onsets is the bar length B and when $[\tau_n] \leq [\tau_{n-1}]$ we interpret that a bar line is crossed between these notes.

To apply the above note-level score model for singing transcription, one must compare all possible segmentations

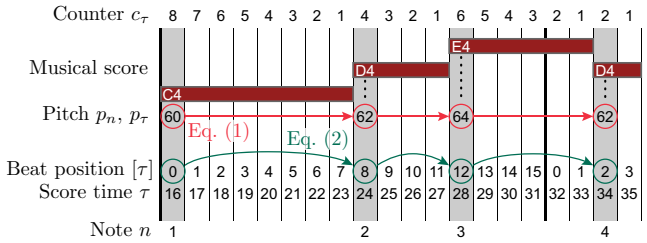


Fig. 3. Generative process of a melody Markov model.

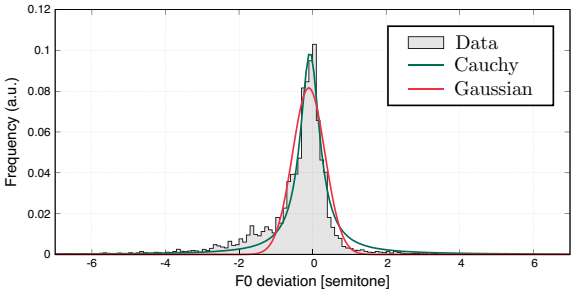


Fig. 4. Distribution of f0 deviations in singing voices.

of the input signal into a sequence of notes. To enable this, we reformulate the score model as an equivalent form that is described at the beat level. Specifically, the above model is embedded in a type of semi-Markov model called the residential time Markov model [25] in which a counter variable is introduced to memorize the residual duration of a note. In this beat-level melody Markov model (Fig. 3), all variables are defined for each beat; the pitch and the counter variable are denoted by p_τ and $c_\tau \in \{1, \ldots, B\}$. The counter variables are generated as

$$P(c_{\tau_s} = c) = \psi_{[\tau_s][\tau_s+c]}, \tag{3}$$

$$P(c_\tau | c_{\tau-1}) = \delta_{c_{\tau-1}(c_\tau+1)} + \delta_{c_{\tau-1}1}\psi_{[\tau][\tau+c_\tau]}, \tag{4}$$

where δ denotes the Kronecker delta. Here, $c_{\tau-1} = 1$ indicates that there is an onset at score time τ and the corresponding note has a length c_τ; otherwise the pitch remains constant. Thus, the generative process for pitches is described as

$$P(p_{\tau_s} = p) = \chi^{\text{ini}}_p, \tag{5}$$

$$P(p_\tau = p \,|\, p_{\tau-1} = p') = \delta_{c_{\tau-1}1}\chi_{p'p} + (1 - \delta_{c_{\tau-1}1})\delta_{pp'}. \tag{6}$$

C. F0 Model and Inference Method

The f0 model describes a stochastic process in which observed f0s are generated from a given score. As a minimal model, we define the output probability for each beat as

$$P(x_{t(\tau)}, \ldots, x_{t(\tau+1)-1} \,|\, p_\tau) = \prod_{t'=t(\tau)}^{t(\tau+1)-1} P_{\text{f0}}(x_{t'} | p_\tau), \tag{7}$$

where we have assumed that each frame has independent and identical probability. It is further assumed that the probabilities $P_{\text{f0}}(x|p)$ for different ps share the same functional form:

$$P_{\text{f0}}(x|p) = F(x - p) \tag{8}$$

for some distribution F. The distribution of f0 deviations $(x - p)$ extracted from annotated f0 data [26] as well as its best fit Cauchy and Gaussian distributions are shown in Fig. 4. The width of the best fit Cauchy distribution is 0.32 (semitones) and the standard deviation of the best fit Gaussian is 0.44 (semitones). We see that the empirical distribution in the data is long-tailed and skewed, and the Cauchy distribution is better fitted than the Gaussian. As in a previous study [16], a Cauchy distribution is used as the distribution F in this study. One would benefit from using a more elaborated distribution incorporating the skewness for improving the transcription accuracy. On the other hand, we have confirmed that the transcription accuracy is not very sensitive to the value of the width parameter for the Cauchy distribution.

Given musical score data, one can estimate parameters of the score model by the maximum-likelihood method. Once the model parameters are given, one can estimate the score from an input signal by maximizing the probability $P(\boldsymbol{p}, \boldsymbol{c} \,|\, X) \propto P(X|\, \boldsymbol{p}, \boldsymbol{c})P(\boldsymbol{p}, \boldsymbol{c})$, where $\boldsymbol{p} = (p_\tau)$ and $\boldsymbol{c} = (c_\tau)$. This can be computed by the standard Viterbi algorithm [27]. Note that, by looking at the obtained counter variables, one can determine the most likely sequence of onset score times as well as the number of notes in the output score.

III. Probabilistic Sequential Pattern Model

We here formulate probabilistic sequential pattern (PSP) models. First, we formalize the idea of representing note patterns related by modifications as a PSP. Second, we explain a score model constructed by using multiple PSPs. In the last subsection, inference methods are explained.

A. Probabilistic Sequential Patterns

For definiteness, a subsequence of notes spanning a bar length is considered as a note pattern in this study. This can be represented as a segment $(p_i, b_i)_{i=1}^I$ where b_i denotes the beat position of the ith note satisfying $0 \le b_1 < \cdots < b_I < B$. Instead of considering a distribution over the set of all possible such patterns explicitly, we consider a probabilistic model that stochastically generates note patterns. The onset beat positions are generated in the same way as the metrical Markov model:

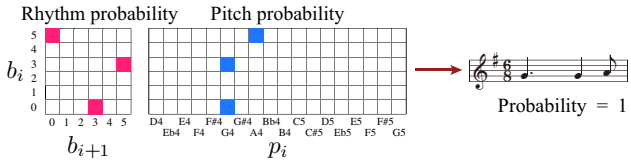
$$P(b_1) = \rho^{\text{ini}}_{b_1}, \quad P(b_i | b_{i-1}) = \rho_{b_{i-1}b_i}, \tag{9}$$

which are referred to as rhythm probabilities. Next, pitches are generated conditionally on the beat position as

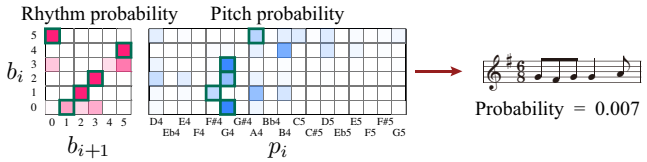
$$P(p_i | b_i) = \phi_{b_i p_i}, \tag{10}$$

which are referred to as pitch probabilities.

This model, named a PSP, can be regarded as a generalization of a note pattern (Fig. 5). This is because in the limit of binary probabilities, i.e. all entries of ρ^{ini}, ρ, and ϕ are either 0 or 1, the model can generate only one note pattern with probability 1. As long as these probability distributions are sparse, a PSP can effectively generate only a limited number of note patterns. Furthermore, these note patterns tend to be related to each other by certain modifications: the metrical Markov model can naturally describe deletions and insertions



(a) A PSP with binary probabilities generates only one note pattern.



(b) A note pattern generated by a (non-deterministic) PSP.

Fig. 5. Examples of note patterns generated by PSPs. In (b), relevant probabilities are marked with a bold green box. (For simpler illustration, the 6/8 time is used here instead of the 4/4 time considered in the text.)

and the pitch probability can express pitch substitutions. The number of parameters of a PSP is $B(B+1+\Omega)$, which is much smaller than the number of unique note patterns as discussed in the Introduction. Note also that pitches and rhythms are not independently generated in a PSP so that it can be a more precise model than the melody Markov model in general.

B. PSP-Based Score Models

1) PSP Mixture Model: The real advantage of considering PSPs becomes clear when we consider multiple PSPs as a generative model of musical scores. The simplest model is described by K PSPs parameterized by $(\rho_b^{(k),\mathrm{ini}}, \rho_{b'b}^{(k)}, \phi_{bp}^{(k)})_{k=1}^K$ and mixture probabilities σ_k obeying $\sigma_1 + \cdots + \sigma_K = 1$. The generative process is described as follows.

1) When a new bar is entered a component k is drawn from the mixture probability.
2) Pitches and onset beat positions in that bar are generated by the k th PSP.
3) Once an onset beat position b_i such that $b_i \le b_{i-1}$ is drawn, we move to the next bar and continue the process.

To put this into equations, let $m = 1, \ldots, M$ be an index for bars and $n = 1, \ldots, N_m$ be an index for note onsets in each bar m. The pitch and onset beat position of the n th note in the m th bar are denoted by p_{mn} and b_{mn}. For each bar m, a PSP k_m is chosen according to the mixture probability

$$P(k_m = k) = \sigma_k. \tag{11}$$

Beat positions $b_{m2}, \ldots, b_{mN_m}$ are generated by

$$P(b_{m(n+1)} = b \,|\, b_{mn} = b', k_m = k) = \rho_{b'b}^{(k)}. \tag{12}$$

The first beat position is generated by

$$P(b_{(m+1)1} = b \,|\, b_{mN_m} = b', k_m = k) = \rho_{b'b}^{(k)}, \tag{13}$$

except for the first bar, for which case the following holds:

$$P(b_{(m=1)1} = b \,|\, k_{(m=1)} = k) = \rho_b^{(k),\mathrm{ini}}. \tag{14}$$

Pitches are generated as

$$P(p_{mn} = p \,|\, b_{mn} = b, k_m = k) = \phi_{bp}^{(k)}. \tag{15}$$

We call this model a PSP mixture model.

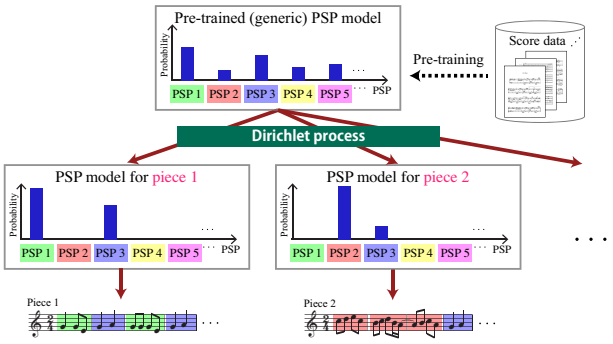


Fig. 6. In the Bayesian extension of the PSP model, an individual PSP model is generated for each musical piece, which is assumed to have sparse distributions to induce repetitions. (For simpler illustration, transition probabilities for PSPs are illustrated as a mixture distribution and the 2/4 time is used here instead of the 4/4 time considered in the text.)

2) Markov PSP Model: We can easily extend the PSP mixture model by introducing a Markov structure on the mixture weights. Instead of the mixture probability, we consider initial and transition probabilities for the PSP components:

$$P(k_1 = k) = \sigma_k^{\mathrm{ini}}, \quad P(k_{m+1} = k' \,|\, k_m = k) = \sigma_{kk'}, \tag{16}$$

$$P(\mathbf{end} \,|\, k_M = k) = \sigma_{k\,\mathbf{end}}, \tag{17}$$

where the last probability is used to end the generative process. The probabilities obey the following normalization conditions:

$$1 = \sum_{k'} \sigma_{k'}^{\mathrm{ini}} = \sum_{k'} \sigma_{kk'} + \sigma_{k\,\mathbf{end}} \quad (\forall k). \tag{18}$$

We call this model a Markov PSP model.

The Markov PSP model has advantages over the PSP mixture model. Since the sequential structure of note patterns is incorporated, it can potentially describe repetitive structure better. In addition, there is a nice theoretical property that with a sufficiently large number of PSPs it can completely reproduce a given piece of music. We thus focus on Markov PSP models in the following and simply call them PSP models.

3) Bayesian Extension: To apply the Bayesian framework for describing repetitive structure explained in the Introduction, we extend the PSP model to a Bayesian model by putting conjugate priors on the model parameters:

$$\boldsymbol{\sigma}^{\mathrm{ini}} \sim \mathrm{Dir}(\alpha_\sigma^{\mathrm{ini}} \bar{\boldsymbol{\sigma}}^{\mathrm{ini}}), \quad \boldsymbol{\sigma}_k \sim \mathrm{Dir}(\alpha_\sigma \bar{\boldsymbol{\sigma}}_k), \tag{19}$$

$$\boldsymbol{\rho}^{\mathrm{ini}} \sim \mathrm{Dir}(\alpha_\rho^{\mathrm{ini}} \bar{\boldsymbol{\rho}}^{\mathrm{ini}}), \quad \boldsymbol{\rho}_b \sim \mathrm{Dir}(\alpha_\rho \bar{\boldsymbol{\rho}}_b), \tag{20}$$

$$\boldsymbol{\phi}_b \sim \mathrm{Dir}(\alpha_\phi \bar{\boldsymbol{\phi}}_b), \tag{21}$$

where we have introduced the notation $\boldsymbol{\sigma}^{\mathrm{ini}} = (\sigma_k^{\mathrm{ini}})$, $\boldsymbol{\sigma}_k = (\sigma_{kk'})$, etc. and $\mathrm{Dir}(\cdot)$ denotes a Dirichlet distribution. The concentration parameters $\alpha_\sigma^{\mathrm{ini}}$, α_σ, etc. are chosen to be small to induce sparse distributions.

The above generative process can be interpreted as a process of choosing a set of note patterns like motives for composing a particular piece (Fig. 6). Eq. (19) says that a limited set of PSPs are chosen, which induces repetitive structure for the generated piece. Eqs. (20) and (21) induce each PSP to become more specific in both rhythm and pitch, which enhances the repetitive structure.

C. Inference Methods for Transcription

In the application of the PSP model to music transcription, there are three inference problems: (i) parameter estimation for a pre-trained (generic) score model; (ii) Bayesian inference of an individual score model given an input signal; and (iii) final estimation of the output score for the input signal. To enable inference, we should reformulate the PSP model as a beat-level model. As explained in Appendix A, the note-level and beat-level Markov PSP models can be reformulated in forms of Markov models.

Unlike conventional score models and the model studied in [13], PSPs are supposed to be pre-trained in an unsupervised manner. As explained in section III-C, the model parameters can be learned by the maximum-likelihood method, similarly as a Gaussian mixture model (GMM). Similarly as the variance of each component Gaussian of a GMM becomes smaller as we increase the number of mixtures, the perplexity of each PSP becomes smaller as we increase the number of PSPs, leading to each PSP generating a more specific subset of note patterns. Therefore, a PSP model will spontaneously find clusters in the space of note patterns that best explain the training data. By varying the number of PSPs K, we can control the preciseness of the model, which is in general in a trade-off relation with the computational cost for inference.

Writing $\theta = (\sigma_k^{\mathrm{ini}}, \sigma_{k'k}, \rho_b^{(k),\mathrm{ini}}, \rho_{b'b}^{(k)}, \phi_{bp}^{(k)})$ for the model parameters and Y for the training score data, the optimal θ is estimated by maximizing the likelihood $P(Y|\theta)$. We can apply the expectation-maximization (EM) algorithm [28] for the maximum-likelihood estimation, by regarding (k_m) as latent variables. Update equations for the EM algorithm are provided in Appendix B. The pre-trained parameters are denoted by $\bar{\theta} = (\bar{\sigma}_k^{\mathrm{ini}}, \bar{\sigma}_{k'k}, \bar{\rho}_b^{(k),\mathrm{ini}}, \bar{\rho}_{b'b}^{(k)}, \bar{\phi}_{bp}^{(k)})$.

The first step of singing transcription by the proposed method is to carry out Bayesian inference for learning an individual score model for the input signal. We can apply the Gibbs sampling method for inferring the parameters θ given their pre-trained values $\bar{\theta}$ and the input signal X. Denoting the latent variables as $Z = (\boldsymbol{k}, \boldsymbol{b}, \boldsymbol{p})$ where $\boldsymbol{k} = (k_m)$, $\boldsymbol{b} = (b_{mn})$, and $\boldsymbol{p} = (p_{mn})$, we sample from the distribution $P(\theta, Z \mid X, \bar{\theta}) \propto P(X|Z,\theta)P(Z|\theta)P(\theta|\bar{\theta})$. For sampling the latent variables, the forward filtering-backward sampling method can be applied. Then the parameters can be sampled from the posterior Dirichlet distributions. In practice, after a certain number of Gibbs samplings, we choose the sampled parameters θ_* that maximize the probability $P(X|\theta_*)$.

The final step of transcription is to estimate the output score Y that maximizes the probability $P(Y|X,\theta_*)$. As in the case of the melody Markov model, this can be done with the Viterbi algorithm. We can simply apply the Markov-model formulation of PSP models explained in Appendix A.

IV. Evaluation

We conduct numerical experiments to compare the PSP model and the melody Markov model. First, they are evaluated as musical score models in terms of perplexity. Next, they are compared in terms of transcription accuracy using real data.

TABLE I
Test-data perplexities.

Model	Test-data perplexity
Melody Markov model	43.9
PSP model ($K = 10$)	46.9
PSP model ($K = 30$)	36.1
PSP model ($K = 50$)	32.8

A. Setup

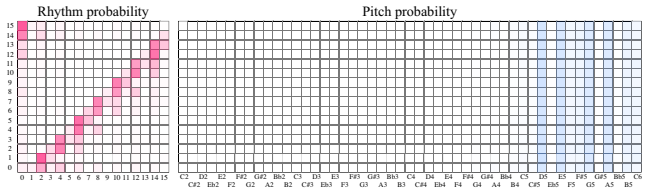
We use the popular musical pieces in the RWC database [26], [29] for evaluation. For the sake of simplicity in data preparation, we only use pieces that are in $4/4$ time without intermediate changes of time signature and remove pieces that have more than two voices in the vocal part or for which the beat and f0 annotation data contain significant errors. 63 pieces remained according to these criteria and are used as test data. The training data consist of the other pieces in the RWC database, 193 pieces by the Beatles, and 135 other pop music pieces, which have no overlap with the test data. To alleviate the problem of data sparseness, the training data is augmented: all pieces are transposed by intervals in the range of $[-12, 12]$ semitones and used for training.

All the concentration parameters of the PSP models are set to unity. For pre-training PSP models, the EM algorithm is run until convergence. For Bayesian inference, Gibbs sampling is iterated 100 times. In general, we can introduce a parameter to weight the relative importance of the score model and the f0 model. Formally, the logarithm of the f0 output probability in Eq. (7) is multiplied by a factor w during inference. We use $w = 0.1$ for Bayesian inference and $w = 1$ in other places, which have been roughly optimized in a preliminary stage.

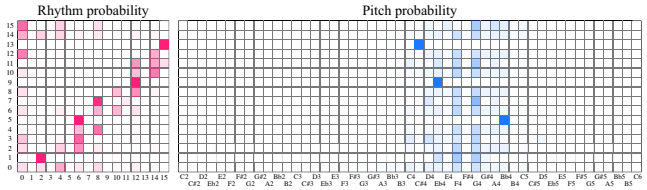
B. Evaluation of Score Models

After parameters of the PSP model are learned with the training data, the perplexity on the test data is computed. The results for PSP models with $K = 10$, 30, and 50 PSPs and for the melody Markov model are given in Table I. We see that the test-data perplexity decreases as K increases and the cases $K = 30$ and 50 outperform the melody Markov model. This indicates that even without a Bayesian extension the PSP model can be useful as a score model with higher predictive ability than the melody Markov model.

Looking at the learned parameters of the PSP models reveals what aspects of musical note sequences they capture with different model sizes. When the model size is small (e.g. $K = 5$), each PSP represents note patterns in different pitch ranges. For $K = 10$, the model begins to learn the structure of musical scales (Fig. 7(a)). With this level of model size, no significant correlation between the probabilities of pitches and rhythms is learned: the pitch probability is almost independent of beat positions and the metrical transition probability is the statistical average of all note patterns. For a larger K (e.g. $K = 50$), each PSP begins to represent a more specific cluster of note patterns. Both the probabilities of pitches and rhythms become sparser in this case (Fig. 7(b)).



(a) The case of 10 PSPs



(b) The case of 50 PSPs

Fig. 7. Examples of learned PSPs. Learned probabilities are visualized for a typical PSP in the case of (a) $K = 10$ and (b) $K = 50$. The left and right boxes represent transition probabilities in Eq. (9) and the pitch probability in Eq. (10). The vertical axis indicates the current beat position and the horizontal axis indicates the next beat position for the rhythm probability and the output pitch for the pitch probability.

TABLE II
AVERAGES AND STANDARD ERRORS OF ERROR RATES EVALUATED ON THE REAL DATA. THE P-VALUE MEASURES STATISTICAL SIGNIFICANCE OF THE DIFFERENCE BETWEEN EACH MODEL AND THE BAYESIAN PSP MODEL WITH $K = 30$, WHICH IS CALCULATED FROM THE DISTRIBUTION OF PIECE-WISE DIFFERENCES IN ERROR RATES .

Model	Error rate (%)	p-value
Melody Markov model	34.2 ± 1.6	$< 10^{-5}$
PSP model ($K = 10$)	31.0 ± 1.7	$< 10^{-5}$
PSP model ($K = 30$)	30.3 ± 1.7	$< 10^{-5}$
Bayesian PSP model ($K = 10$)	30.4 ± 1.6	$< 10^{-5}$
Bayesian PSP model ($K = 30$)	28.6 ± 1.6	—
HHSMM [16]	27.6 ± 1.7	5.7×10^{-3}

C. Evaluation of Transcription Accuracy

Next we evaluate the PSP model and the melody Markov model in terms of transcription accuracy using real data of f0 trajectories. We use the annotated f0 and beat tracking data for the RWC data [26] as input. For the PSP models, we compare the cases $K = 10$ and $K = 30$ and both cases of using and not using Bayesian inference. As an evaluation measure, a beat-level error rate of estimated pitches is used.

The average error rates in Table II show that the PSP models significantly outperform the melody Markov model. For both $K = 10$ and 30, the Bayesian extension yields better results, even though the difference is slight for $K = 10$. To see the effect of the PSP model in more detail, we plot the improvement of error rates for each piece for the case $K = 30$ (Fig. 8). We see that for all pieces except one the non-Bayesian PSP model improves the error rate. The piece for which the PSP model has a worse error rate (piece ID 7/RWC No. 16) has an f0 trajectory that deviates significantly from the musical score. For most pieces the Bayesian PSP model further improves the error rate, but it sometimes yields a worse error rate. This implies the possibility of further improving the result if we are able to adjust the parameters (e.g. concentration parameters and the width parameter of the f0 model) for

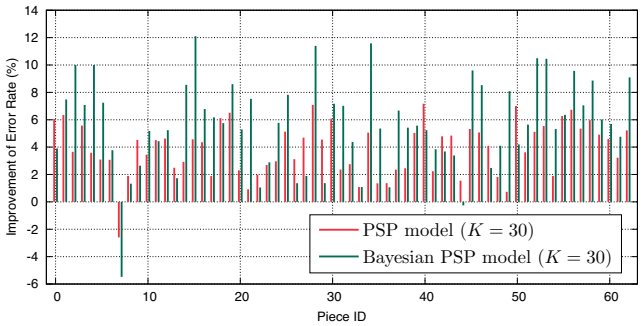


Fig. 8. Improvements in the error rate compared to the melody Markov model.

individual signals. For reference, the error rate for a state-of-the-art model [16] is also shown in Table II, which is slightly better than the best case for the PSP model. This result is encouraging given that the compared model has a much more elaborated f0 model [16], which can be incorporated into the present model in principle.

Transcription results for an example piece (piece ID 32/RWC No. 55) are shown in Fig. 9 together with the ground truth. We see that the repetitive structure in the ground-truth data is better reproduced in the transcription by the Bayesian PSP model than the other two cases. We can find some incomplete repetitions of the first and third bars, which lead to transcription errors in this example. This shows the potential of the Bayesian PSP model to capture approximate repetitions, which often appear in other pieces. Musical naturalness is also improved by the Bayesian PSP model, as we can see from the absence of out-of-scale notes that are present in the transcriptions by the other methods. The example also reveals a limitation of the present model that it is hard to recognize repeated notes, as in the first and fifth bars. To solve this problem, it would be necessary to incorporate some feature, like the spectral flux or the magnitude of percussive spectral components, that can indicate locations of onsets.

V. CONCLUSION

We have formulated a probabilistic description of musical note patterns named probabilistic sequential pattern (PSP) and constructed musical score models that generate note sequences in units of PSPs. The model enables a statistical description of repetitive structure through a Bayesian extension. We have confirmed that even with a moderate number of PSPs (e.g. $K = 30$), the PSP model yields better test-data perplexities and transcription accuracies than the conventional Markov model. The PSPs can be learned unsupervisedly and automatically capture aspects of music that provide more relevant information depending on the model size. Within the range of model sizes we studied, first the pitch range is captured, second musical-scale structure, and then more specific patterns in which pitches and rhythms are correlated.

The description of the PSP models given here is a minimal one; several ways of extension are expected to improve the models. First, whereas it is interesting that the PSP model spontaneously learns the structure of key/musical scale, intro-

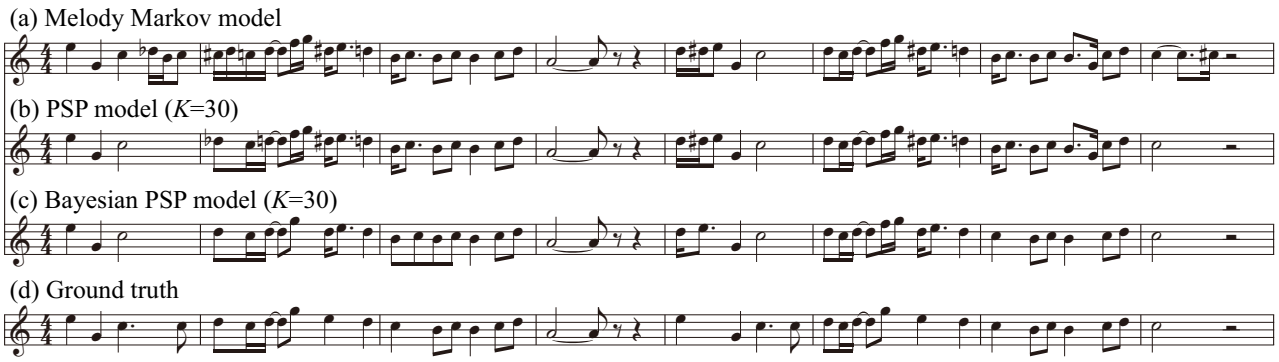


Fig. 9. Example transcription results (RWC No. 55).

ducing transposition invariance in the model can lead to more efficient learning, leading to a model in which PSPs focus on clustering note patterns in one key. Second, a limitation of the current model that sequential dependence between succeeding pitches is not explicitly incorporated can be overcome by an autoregressive extension of the pitch probability.

For further improving the accuracy of singing transcription, several directions of model adaptation would also be effective. To adapt the f0 model for individual singers, one can infer the parameters, especially the width parameter, in a Bayesian manner [30]. We have also observed that the optimal values of the concentration parameters are different for individual pieces. Extension to a hierarchical Bayesian model for inferring these parameters is thus another possibility.

The present model can be applied to other tasks including automatic composition and arrangement. Extension for polyphonic music is of great importance for extending the application of the approach, which is currently under investigation.

ACKNOWLEDGMENT

This study was partially supported by JSPS KAKENHI Grant Numbers 26700020, 16H01744, 16H02917, 16J05486, and 16K00501 and JST ACCEL No. JPMJAC1602. E.N. is supported by the JSPS research fellowship (PD).

APPENDIX

A. Markov PSP Models Formulated as Markov Models

We first formulate a Markov PSP model as a Markov model. We introduce stochastic variables (k_n, b_n, p_n) that are defined for each note n (indexed throughout a piece). The initial and transition probabilities are

$$P(k_1, b_1, p_1) = \sigma^{\mathrm{ini}}_{k_1} \rho^{(k_1),\mathrm{ini}}_{b_1} \phi^{(k_1)}_{b_1 p_1}, \tag{22}$$

$$P(k_n, b_n, p_n \,|\, k_{n-1}, b_{n-1}, p_{n-1}) = \begin{Bmatrix} \delta_{k_n k_{n-1}} & (b_n > b_{n-1}) \\ \sigma_{k_{n-1} k_n} & (b_n \le b_{n-1}) \end{Bmatrix} \cdot \rho^{(k_{n-1})}_{b_{n-1} b_n} \phi^{(k_n)}_{b_n p_n}. \tag{23}$$

This reproduces the complete-data probability for the Markov PSP model.

A Markov PSP model can be represented as a beat-level score model by using the same framework as in section II-B. We define variables (k_τ, p_τ, c_τ) for each score time $\tau \in \{\tau_s, \ldots, \tau_e\}$. The initial and transition probabilities are then given as

$$P(k_{\tau_s} = k, p_{\tau_s} = p, c_{\tau_s} = c) = \sigma^{\mathrm{ini}}_k \rho^{(k)}_{[\tau_s][\tau_s + c]} \phi^{(k)}_{[\tau_s]p} \tag{24}$$

$$P(k_\tau = k, p_\tau = p, c_\tau = c \,|\, k_{\tau-1} = k', p_{\tau-1} = p', c_{\tau-1} = c') = \begin{Bmatrix} \delta_{k'k} & ([\tau] \neq 0) \\ \sigma_{k'k} & ([\tau] = 0) \end{Bmatrix} \cdot \begin{Bmatrix} \delta_{c(c'-1)} \delta_{pp'} & (c_{\tau-1} > 1) \\ \rho^{(k_\tau)}_{[\tau][\tau+c]} \phi^{(k_\tau)}_{[\tau]p} & (c_{\tau-1} = 1) \end{Bmatrix}. \tag{25}$$

B. EM Algorithm for Markov PSP models

Readers are reminded the notation introduced in sections III-B and III-C. Update equations for the EM algorithm can be derived by differentiating the following function with respect to θ with constraints for normalizations [28]:

$$\mathcal{F} = -\sum_{\boldsymbol{K}} P(\boldsymbol{K} |\, \boldsymbol{B}, \boldsymbol{P}, \theta') \ln P(\boldsymbol{K}, \boldsymbol{B}, \boldsymbol{P}, \theta), \tag{26}$$

where θ' denotes the parameters before an update and we have introduced variables $\boldsymbol{B} = (\boldsymbol{b}^l)$ and $\boldsymbol{P} = (\boldsymbol{p}^l)$ representing the training data consisting of multiple musical scores indexed by l, and $\boldsymbol{K} = (\boldsymbol{k}^l)$ is the corresponding mixture variable. The results are summarized as follows:

$$\sigma^{\mathrm{ini}}_k = \frac{1}{\lambda} \sum_l P(k^l_1 = k \,|\, \boldsymbol{b}^l, \boldsymbol{p}^l, \theta'), \tag{27}$$

$$\sigma_{kk'} = \frac{1}{\lambda_k} \sum_{l,m} P(k^l_m = k, k^l_{m+1} = k' \,|\, \boldsymbol{b}^l, \boldsymbol{p}^l, \theta'), \tag{28}$$

$$\rho^{(k)}_{bb'} = \frac{1}{\lambda_{kb}} \sum_{l,m,n} \delta_{bb^l_{mn}} \delta_{b' b^l_{m(n+1)}} P(k^l_m = k \,|\, \boldsymbol{b}^l, \boldsymbol{p}^l, \theta'), \tag{29}$$

$$\phi^{(k)}_{bp} = \frac{1}{\xi_{kb}} \sum_{l,m,n} \delta_{bb^l_{mn}} \delta_{pp^l_{mn}} P(k^l_m = k \,|\, \boldsymbol{b}^l, \boldsymbol{p}^l, \theta'). \tag{30}$$

Here, λ, λ_k, λ_{kb}, and ξ_{kb} are normalization constants.

In the above equations, the relevant probabilities $P(k^l_m \,|\, \boldsymbol{b}^l, \boldsymbol{p}^l, \theta')$ and $P(k^l_m, k^l_{m+1} \,|\, \boldsymbol{b}^l, \boldsymbol{p}^l, \theta')$ can be computed by the forward-backward algorithm. Defining the

forward and backward variables and an additional variable as

$$\alpha_{lm}(k) = P(k^l_m = k, \boldsymbol{b}^l_{1:m}, \boldsymbol{p}^l_{1:m} \,|\theta'), \tag{31}$$

$$\beta_{lm}(k) = P(\boldsymbol{b}^l_{(m+1):M_l}, \boldsymbol{p}^l_{(m+1):M_l} \,|\, k^l_m = k, \theta'), \tag{32}$$

$$\Phi'_{klm} = P(\boldsymbol{b}^l_m, \boldsymbol{p}^l_m \,|\, k^l_m = k, \theta') = \prod_{n=1}^{N_m} \rho^{\prime(k)}_{b^l_{mn} b^l_{m(n+1)}} \phi^{\prime(k)}_{b^l_{mn} p^l_{mn}}, \tag{33}$$

the forward and backward algorithms go as

$$\alpha_{l1}(k) = \sigma'^{\mathrm{ini}}_k \Phi'_{kl1}, \tag{34}$$

$$\alpha_{lm}(k) = \sum_{k'} \alpha_{l(m-1)}(k') \sigma'_{k'k} \Phi'_{klm}, \tag{35}$$

$$\beta_{lM_l}(k) = \sigma'_{k\,\mathbf{end}}, \tag{36}$$

$$\beta_{lm}(k) = \sum_{k'} \sigma'_{kk'} \Phi'_{k'l(m+1)} \beta_{l(m+1)}(k'). \tag{37}$$

Then we have

$$P(k^l_m = k \,|\, \boldsymbol{b}^l, \boldsymbol{p}^l, \theta') \propto \alpha_{lm}(k) \beta_{lm}(k), \tag{38}$$

$$P(k^l_m = k, k^l_{m+1} = k' \,|\, \boldsymbol{b}^l, \boldsymbol{p}^l, \theta') \propto \alpha_{lm}(k) \sigma'_{kk'} \Phi'_{k'l(m+1)} \beta_{l(m+1)}(k'), \tag{39}$$

where the normalization factor can be computed as $P(\boldsymbol{b}^l, \boldsymbol{p}^l \,|\theta') = \sum_k \alpha_{lM_l}(k) \sigma'_{k\,\mathbf{end}}$.

REFERENCES

[1] H. Papadopoulos and G. Peeters. Large-scale study of chord estimation algorithms based on chroma representation and HMM. In *Proc. CBMI*, pages 53–60, 2007.

[2] M. McVicar, R. Santos-Rodríguez, Y. Ni, and T. De Bie. Automatic chord estimation from audio: A review of the state of the art. *IEEE/ACM TASLP*, 22(2):556–575, 2014.

[3] S. A. Raczynski, S. Fukayama, and E. Vincent. Melody harmonization with interpolated probabilistic models. *J. New Music Res.*, 42(3):223–235, 2013.

[4] C. Raphael and J. Stoddard. Functional harmonic analysis using probabilistic models. *Comp. Music J.*, 28(3):45–52, 2004.

[5] D. Temperley. A unified probabilistic model for polyphonic music analysis. *J. New Music Res.*, 38(1):3–18, 2009.

[6] E. Nakamura, K. Yoshii, and S. Sagayama. Rhythm transcription of polyphonic piano music based on merged-output HMM for multiple voices. *IEEE/ACM TASLP*, 25(4):794–806, 2017.

[7] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: Challenges and future directions. *J. Intelligent Information Systems*, 41(3):407–434, 2013.

[8] R. Scholz, E. Vincent, and F. Bimbot. Robust modeling of musical chord sequences using probabilistic N-grams. In *Proc. ICASSP*, pages 53–56, 2009.

[9] H. Tsushima, E. Nakamura, K. Itoyama, and K. Yoshii. Generative statistical models with self-emergent grammar of chord sequences. *J. New Music Res.*, 47, 2018. to appear.

[10] A. Ycart and E. Benetos. A study on LSTM networks for polyphonic music sequence modelling. In *Proc. ISMIR*, pages 421–427, 2017.

[11] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang. MIDINET: A convolutional generative adversarial network for symbolic-domain music generation. In *Proc. ISMIR*, pages 324–331, 2017.

[12] D. Meredith (ed.). *Computational Music Analysis*. Springer, 2016.

[13] E. Nakamura, K. Itoyama, and K. Yoshii. Rhythm transcription of MIDI performances based on hierarchical Bayesian modelling of repetition and modification of musical note patterns. In *Proc. EUSIPCO*, pages 1946–1950, 2016.

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

[15] R. Nishikimi, E. Nakamura, K. Itoyama, and K. Yoshii. Musical note estimation for F0 trajectories of singing voices based on a Bayesian semi-beat-synchronous HMM. In *Proc. ISMIR*, pages 461–467, 2016.

[16] R. Nishikimi, E. Nakamura, M. Goto, K. Itoyama, and K. Yoshii. Scale- and rhythm-aware musical note estimation for vocal F0 trajectories based on a semi-tatum-synchronous hierarchical hidden semi-Markov model. In *Proc. ISMIR*, pages 376–382, 2017.

[17] S. Raczynski, E. Vincent, and S. Sagayama. Dynamic Bayesian networks for symbolic polyphonic pitch modeling. *IEEE/ACM TASLP*, 21(9):1830–1840, 2013.

[18] S. Sigtia, E. Benetos, and S. Dixon. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM TASLP*, 24(5):927–939, 2016.

[19] C. Raphael. A graphical model for recognizing sung melodies. In *Proc. ISMIR*, pages 658–663, 2005.

[20] M. P. Ryynänen and A. P. Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music J.*, 32(3):72–86, 2008.

[21] Y. Ikemiya, K. Yoshii, and K. Itoyama. Singing voice analysis and editing based on mutually dependent F0 estimation and source separation. In *Proc. ICASSP*, pages 574–578, 2015.

[22] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer. madmom: A new python audio and music signal processing library. In *Proc. ACM Multimedia*, pages 1174–1178, 2006.

[23] C. Raphael. A hybrid graphical model for rhythmic parsing. *Artificial Intelligence*, 137:217–238, 2002.

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

[25] S.-Z. Yu. Hidden semi-Markov models. *Artificial Intelligence*, 174:215–243, 2010.

[26] M. Goto. AIST annotation for the RWC music database. In *Proc. ISMIR*, pages 359–360, 2006.

[27] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, 1989.

[28] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[29] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: Popular, classical and jazz music databases. In *Proc. ISMIR*, pages 287–288, 2002.

[30] R. Nishikimi, E. Nakamura, M. Goto, K. Itoyama, and K. Yoshii. Bayesian singing transcription based on integrated musical score and F0 trajectory models. In preparation.