MUSICAL NOTE ESTIMATION FOR F0 TRAJECTORIES OF SINGING VOICES BASED ON A BAYESIAN SEMI-BEAT-SYNCHRONOUS HMM

Ryo Nishikimi[1] **Eita Nakamura**[1] **Katsutoshi Itoyama**[1] **Kazuyoshi Yoshii**[1]

[1] Graduate School of Informatics, Kyoto University , Japan

{nishikimi, enakamura}@sap.ist.i.kyoto-u.ac.jp, {itoyama, yoshii}@kuis.kyoto-u.ac.jp

ABSTRACT

This paper presents a statistical method that estimates a sequence of discrete musical notes from a temporal trajectory of vocal F0s. Since considerable effort has been devoted to estimate the frame-level F0s of singing voices from music audio signals, we tackle musical note estimation for those F0s to obtain a symbolic musical score. A naïve approach to musical note estimation is to quantize the vocal F0s at a semitone level in every time unit (*e.g.*, half beat). This approach, however, fails when the vocal F0s are significantly deviated from those specified by a musical score. The onsets of musical notes are often delayed or advanced from beat times and the vocal F0s fluctuate according to singing expressions. To deal with these deviations, we propose a Bayesian hidden Markov model that allows musical notes to change in semi-synchronization with beat times. Both the semitone-level F0s and onset deviations of musical notes are regarded as latent variables and the frequency deviations are modeled by an emission distribution. The musical notes and their onset and frequency deviations are jointly estimated by using Gibbs sampling. Experimental results showed that the proposed method improved the accuracy of musical note estimation against baseline methods.

1. INTRODUCTION

Singing voice analysis is one of the most important topics in the field of music information retrieval because singing voice usually forms the melody line of popular music and it has a strong impact on the mood and impression of a musical piece. The widely studied tasks in singing voice analysis are fundamental frequency (F0) estimation [1,4,5,8,10,15,22] and singing voice separation [9,13] for music audio signals. These techniques can be used for singer identification [11,23], Karaoke systems based on singing voice suppression [2,20], and a music listening system that helps a user focus on a particular musical element (*e.g.*, vocal part) for deeper music understanding [7].

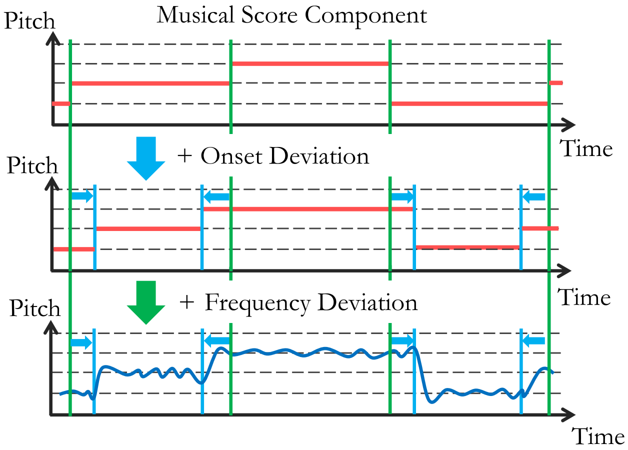


Figure 1: The process generating F0 trajectories of singing voices.

In this study we tackle a problem called *musical note estimation* that aims to recover a sequence of musical notes from an F0 trajectory of singing voices. While a lot of effort has been devoted to F0 estimation of singing voices, musical note estimation should be investigated additionally to complete automatic music transcription, *i.e.*, convert the estimated F0 trajectory to a musical score containing only discrete symbols. If beat information is available, a naïve approach to this problem is to quantize the vocal F0s contained in every time unit (*e.g.*, half beat) into a semitone-level F0 with a majority vote [7]. This approach, however, often fails to work when the vocal F0s are significantly deviated from exact semitone-level F0s specified by a musical score or the melody is sung in a tight or lazy singing style such that the onsets of musical notes are significantly advanced or delayed from exact beat times.

To solve this problem, we propose a statistical method based on a hidden Markov model (HMM) that represents how a vocal F0 trajectory is generated from a sequence of latent musical notes (Fig. 1). The F0s of musical notes in a musical score can take only discrete values with the interval of semitones and tend to vary at a beat, half-beat, or quarter-beat level. The vocal F0 trajectory in an actual performance, on the other hand, is a continuous signal that can dynamically and smoothly vary over time. To deal with both types of F0s from a generative viewpoint, we formulate a semi-beat-synchronous HMM (SBS-HMM) allowing the continuous F0s of a sung melody to deviate from the discrete F0s of written musical notes along the time and frequency directions. In the proposed HMM, the semitone-level F0s and onset deviations of musical notes are encoded as latent variables and the F0 deviations of musical notes are modeled by emission probability distributions. Given an F0 trajectory and beat times, all the variables and distributions are estimated jointly using Gibbs sampling.

2. RELATED WORK

This section introduces related work on singing voices.

2.1 Pitch Estimation of Singing Voice

Many studies on estimating a vocal F0 trajectory in a music audio signal have been conducted [1,4,5,8,10,15,22]. Subharmonic summation (SHS) [8] is a method in which the fundamental frequency for each time is determined by calculating the sum of the powers of the harmonic components of each candidate fundamental frequency $\{f_0, \ldots, f_M\}$. PreFEst [5] is a method that estimates the F0 trajectories of a melody and a bass line by extracting the most predominant harmonic structure from a polyphonic music audio signal. Ikemiya et al. [10] proposed a method in which singing voice separation and F0 estimation are performed mutually. First a singing voice is separated, from the spectrogram obtained by the short-time Fourier transform (STFT) for a music audio signal, by using a robust principal component analysis (RPCA) [9], and then a vocal F0 trajectory is obtained with the Viterbi algorithm by using SHS for a separated singing voice. Salamon et al. [22] used the characteristics of vocal F0 contours for melody extraction. Durrieu et al. [4] proposed a method for melody extraction in which the main melody is represented as a source-filter model and the accompaniment of the music mixture is represented as a non-negative matrix factorization (NMF)-based model. De Cheveigné et al. [1] proposed a autocorrelation based method for fundamental frequency estimation which is expanded to decrease an error rate. This method is called YIN. Mauch et al. [15] extended YIN in a probabilistic way to output multiple pitch candidates. This method is called pYIN.

2.2 Note Estimation of Singing Voice

A method for estimating the sequence of musical notes by quantizing pitches of a vocal F0 trajectory has been proposed. A majority-vote method described in Sec. 1 was implemented in Songle [7]. The method has a limit because it doesn't consider the singing expression nor the typical occurrence of pitches in succession. Paiva et al. [17] proposed a method that has five stages and detects melody notes in polyphonic musical signals, and Raphael [19] proposed an HMM-based method that simultaneously estimates rhythms, tempos, and notes from a solo singing voice acoustic signal. Poliner et al. [18] proposed a method based on a support vector machines (SVM) classifier which doesn't need the assumption that a musical pitch is realized as a set of harmonics of a particular fundamental. Laaksonen [12] proposed a melody transcription method that uses chord information, and Ryynänen et al. [21] proposed a method for transcribing the melody, bass line, and chords in polyphonic music. A software tool called Tony developed by Mauch et al. [14] estimates musical notes from the output of pYIN by Viterbi-decoding of an HMM.

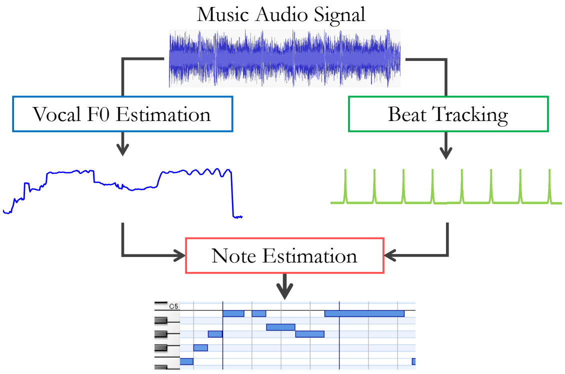


Figure 2: Overview of the proposed musical note estimation method based on a semi-beat-synchronous HMM.

2.3 Analysis of Vocal F0 Trajectories

Studies on extracting the personality and habit of singing expression from vocal F0 trajectories have been conducted. Ohishi et al. [16] proposed a model that represents the generating process of vocal F0 trajectories in consideration of the time and frequency deviations. In that model the vocal F0 trajectory consists of three components: note, expression, and fine deviation components. The note component contains the note transition and overshoot, and the expression component contains vibrato and portamento. The note and expression components are represented as the outputs of second-order linear systems driven by the note and expression commands. The note and expression commands represent the sequence of musical notes and the musical expressive intentions, respectively. The note command and the expression command are represented with HMMs. Although the method can extract the personality of the singing expression from vocal F0 trajectories, it assumes that the music score is given in advance and cannot be directly applied for note estimation.

3. PROPOSED METHOD

This section explains the proposed method for estimating a sequence of latent musical notes from the observed vocal F0 trajectories by formulating an SBS-HMM which represents the generating process of the observations. An observed F0 trajectory is stochastically generated by imparting frequency and onset deviations to a step-function-like F0 trajectory that varies exactly on a 16th-note-level grid according to a music score. The semitone-level F0s (called pitches for simplicity in this paper) between adjacent grid lines and the onset deviations are represented as latent variables (states) of the HMM. Since the frequency deviations are represented by emission probability distributions of the HMM, a semi-beat-synchronous step-function-like F0 trajectory is generated in the latent space and its finely-fluctuated version is then generated in the observed space.

3.1 Problem Specification

The problem of musical note estimation is formally defined (Fig. 2) as follows:

Input: a vocal F0 trajectory $\boldsymbol{X} = \{x_t\}_{t=1}^T$ and 16th-note-level beat times $\boldsymbol{\psi} = \{\psi_n\}_{n=1}^N$ automatically estimated from a music audio signal.
Output: a sequence of pitches $\boldsymbol{Z} = \{z_n\}_{n=1}^N$.

Here, t is the time frame index, T is the number of time frames in the target signal, x_t indicates a log-frequency in cents at frame t, N is the number of beat times, ψ_n is the n-th beat time, and z_n indicates a pitch between ψ_{n-1} and ψ_n taking one of $\{\mu_1, \ldots, \mu_K\}$, where K is the number of kinds of pitches that can appear in a music score. The beginning and end of music are represented as $\psi_0 = 1$ and $\psi_N = T+1$ respectively. In this paper we assume that each z_n corresponds to a 16th note for simplicity. A longer musical note is represented by a subsequence of $\{z_n\}_{n=1}^N$ having the same pitch in succession.

3.2 Model Formulation

We explain how to formulate the SBS-HMM that simultaneously represents the pitch transitions and onset and frequency deviations of musical notes.

3.2.1 Modeling Pitch Transitions

A sequence of latent pitches $\boldsymbol{Z}$ forms a first-order Markov chain given by

$$z_n | z_{n-1}, \boldsymbol{A} \sim \mathrm{Categorical}(z_n | \boldsymbol{a}_{z_{n-1}}), \qquad (1)$$

where $\boldsymbol{A} = [\boldsymbol{a}_1^T, \cdots, \boldsymbol{a}_K^T]$ is a K-by-K transition probability matrix such that $\sum_{k=1}^K a_{jk} = 1$ for any j. The initial latent state z_1 is determined as follows:

$$z_1 | \boldsymbol{\pi} \sim \mathrm{Categorical}(z_1 | \boldsymbol{\pi}), \qquad (2)$$

where $\boldsymbol{\pi} = [\pi_1, \cdots, \pi_K]^T$ is a K-dimensional vector such that $\sum_{k=1}^K \pi_k = 1$.

3.2.2 Modeling Onset Deviations

The onset deviations of musical notes $\boldsymbol{\tau} = \{\tau_n\}_{n=1}^N$ are represented as discrete latent variables taking integer values between $-G$ and G. Let $\phi_n = \psi_n + \tau_n$ be the actual onset time of the n-th musical note. Note that $\tau_0 = 0$ and $\tau_N = 0$ at the beginning and end of a vocal F0 trajectory. We assume that τ_n is stochastically generated as follows:

$$\tau_n | \boldsymbol{\rho} \sim \mathrm{Categorical}(\tau_n | \boldsymbol{\rho}), \qquad (3)$$

where $\boldsymbol{\rho} = [\rho_{-G}, \ldots, \rho_G]^T$ is a $(2G+1)$-dimensional vector such that $\sum_{g=-G}^G \rho_g = 1$.

3.2.3 Modeling Frequency Deviations

The observed F0 x_t $(\phi_{n-1} \leq t < \phi_n)$ is stochastically generated by imparting a probabilistic frequency deviation to the semitone-level pitch μ_{z_k} assigned to each beat interval. Assuming that x_t is independently generated at each frame, the emission probability of the n-th beat interval in the case of $z_n = k, \tau_{n-1} = f, \tau_n = g$ is given by

$$b_{nkfg} \equiv \left\{ \prod_{t=\phi_{n-1}}^{\phi_n - 1} p(x_t | z_n = k) \right\}^{\frac{1}{\phi_n - \phi_{n-1}}}, \qquad (4)$$

where $p(x_t | z_n = k)$ is the emission probability of each frame. To balance the effects of transition probabilities and emission probabilities, we exponentiate the product of emission probabilities of frames in a beat interval by the number of frames in a beat interval. We use as $p(x_t|z_n)$ the Cauchy distribution, which is robust against outliers and is defined by

$$\mathrm{Cauchy}(x; \mu, \lambda) = \frac{\lambda}{\pi \left\{ (x-\mu)^2 + \lambda^2 \right\}}, \qquad (5)$$

where μ is a location parameter that defines the mode of the distribution and λ is a scale parameter. When the pitch of the n-th beat interval is $z_n = k$, μ takes the value μ_k. The scale parameter takes a value that does not depend on the pitch z_n.

Since actual vocal F0s are significantly deviated from those specified by a musical score, the scale parameter of a Cauchy distribution is allowed to change according to the difference of adjacent F0s; *i.e.*, $\Delta x_t \equiv x_t - x_{t-1}$. The scale parameter is set to be proportional to the absolute value of Δx_t and defined for each frame t as follows:

$$\lambda_t = c \cdot |\Delta x_t| + d, \qquad (6)$$

where c is an coefficient and d is a constant term. If $d = 0$, $p(x_t|z_n)$ cannot be calculated when $\lambda_t = 0$ and $\Delta x_t = 0$. To avoid this problem, we introduce the constant term d.

3.2.4 Incorporating Prior Distributions

We put conjugate Dirichlet priors on model parameters $\boldsymbol{A}$, $\boldsymbol{\pi}$, and $\boldsymbol{\rho}$ as follows:

$$\boldsymbol{a}_j \sim \mathrm{Dirichlet}(\boldsymbol{a}_j | \boldsymbol{\xi}_j), \qquad (7)$$

$$\boldsymbol{\pi} \sim \mathrm{Dirichlet}(\boldsymbol{\pi} | \boldsymbol{\zeta}), \qquad (8)$$

$$\boldsymbol{\rho} \sim \mathrm{Dirichlet}(\boldsymbol{\rho} | \boldsymbol{\eta}), \qquad (9)$$

where $\boldsymbol{\xi}_j = [\xi_1, \ldots, \xi_K]^T$ and $\boldsymbol{\zeta} = [\zeta_1, \ldots, \zeta_K]^T$ are K-dimensional vectors and $\boldsymbol{\eta} = [\eta_{-G}, \ldots, \eta_G]^T$ is a $(2G+1)$-dimensional vector.

We then put on gamma priors on nonnegative Cauchy parameters c and d as follows:

$$c \sim \mathrm{Gamma}(c | c_0, c_1), \qquad (10)$$

$$d \sim \mathrm{Gamma}(d | d_0, d_1), \qquad (11)$$

where c_0 and d_0 are shape parameters and c_1 and d_1 are rate parameters.

3.3 Bayesian Inference

The goal of Bayesian inference is to calculate the posterior distribution $p(\boldsymbol{Z}, \boldsymbol{\tau}, \boldsymbol{A}, \boldsymbol{\pi}, \boldsymbol{\rho}, c, d | \boldsymbol{X})$. Since this computation is analytically intractable, we use Markov chain Monte Carlo (MCMC) methods for sampling the values of those variables. Let $\boldsymbol{\Theta} = \{\boldsymbol{A}, \boldsymbol{\pi}, \boldsymbol{\rho}\}$ be a set of model parameters. To get samples of $\boldsymbol{\Theta}$, the Gibbs sampling algorithm is used. To get samples of sequential latent variables $\boldsymbol{Z}$ and $\boldsymbol{\tau}$, on the other hand, a kind of blocked Gibbs sampling algorithms called a forward filtering-backward sampling algorithm is used. These steps are iterated in a similar way to an expectation maximization (EM) algorithm called the Baum-Welch algorithm used for unsupervised learning of an HMM. Since the conjugacy is not satisfied for the distributions regarding c and d, we use the Metropolis-Hastings (MH) algorithm.

3.3.1 Inferring Latent Variables $\boldsymbol{Z}$ and $\boldsymbol{\tau}$

We explain how to sample a sequence of latent variables $\boldsymbol{Z}$ and $\boldsymbol{\tau}$. For each beat interval, we calculate the probability given by

$$\beta_{njf} = p(z_n = j, \tau_n = f | z_{n+1:N}, \tau_{n+1:N}, x_{1:T}), \quad (12)$$

where $z_{n+1:N}$, $\tau_{n+1:N}$, and $x_{1:T}$ represent $z_{n+1}, \ldots, z_N$, $\tau_{n+1}, \ldots, \tau_N$, and $x_1, \ldots, x_T$, respectively. The latent variables of the n-th beat interval (z_n, τ_n) are sampled in accordance with β_{njf}. The calculation of Eq. (12) and the sampling of the latent variables are performed by using forward filtering-backward sampling.

In forward filtering, we recursively calculate the probability given by

$$\alpha_{nkg} = p(X_{10\tau_1}, \ldots, X_{(n-1)\tau_{n-2}f}, X_{nfg}, z_n{=}k, \tau_n{=}g),$$

where X_{nfg} represents the observations x_t in the beat interval from ϕ_{n-1} to ϕ_n when $\tau_{n-1} = f$ and $\tau_n = g$. α_{nkg} is calculated as follows:

$$\begin{aligned}
\alpha_{1kg} &= p(X_{10g}, z_1 = k, \tau_1 = g) \\
&= p(X_{10g} | z_1 = k, \tau_1 = g) p(z_1 = k) p(\tau_1 = g) \\
&= b_{1k0g} \pi_k \rho_g, \quad (13) \\
\alpha_{nkg} &= p(X_{10\tau_1}, \ldots, X_{nfg}, z_n = k, \tau_n = g) \\
&= \sum_{f=-G}^{G} p(X_{nfg} | z_n = k, \tau_{n-1} = f, \tau_n = g) \\
&\quad \cdot \sum_{j=1}^{K} p(X_{10\tau_1}, \ldots, X_{(n-1)\tau_{n-2}f}, z_{n-1}{=}j, \tau_{n-1}{=}f) \\
&\quad \cdot p(z_n{=}k | z_{n-1}{=}j) p(\tau_n{=}g) \\
&= \sum_{f=-G}^{G} b_{nkfg} \sum_{j=1}^{K} \alpha_{(n-1)jf} a_{jk} \rho_g. \quad (14)
\end{aligned}$$

In backward sampling, Eq. (12) is calculated in the n-th beat interval by using the value of α_{nkg}, and the states (z_n, τ_n) are sampled recursively. When the $(n{+}1)$-th sampled states are $(z_{n+1}, \tau_{n+1}) = (k, g)$, β_{njf} is calculated as follows:

$$\begin{aligned}
\beta_{njf} &\propto p(X_{(n+1)fg} | z_{n+1} = k, \tau_n = f, \tau_{n+1} = g) \\
&\quad \cdot p(z_{n+1} = k | z_n = j) p(\tau_{n+1} = g) \\
&\quad \cdot p(X_{10\tau_1}, \ldots, X_{n\tau_n f}, z_n = j, \tau_n = f) \\
&= b_{(n+1)kfg} a_{jk} \rho_g \alpha_{njf}. \quad (15)
\end{aligned}$$

Specifically, the latent variables (z_N, τ_N) are sampled in accordance with the probability given by

$$\beta_{Njf} \propto \alpha_{Njf}. \quad (16)$$

3.3.2 Learning Model Parameters $\boldsymbol{A}$, $\boldsymbol{\pi}$, and $\boldsymbol{\rho}$

We explain how to learn the values of $\boldsymbol{\Theta}$. In a sequence of latent variables $\{z_n, \tau_n\}_{n=1}^{N}$ which are sampled in backward sampling, the number of transitions such that $z_n = j$ and $z_{n+1} = k$ is represented as s_{jk} and the number of onset deviations such that $\tau_n = g$ is represented as u_g. The value of v_k is 1 at $z_1 = k$, and else where is 0. The parameters a_{jk}, ρ_g and π_k are updated by sampling from the posterior distributions given by

$$\begin{aligned}
p(\boldsymbol{a}_j | \boldsymbol{\xi}_j + \boldsymbol{s}_j) &= \mathrm{Dirichlet}(\boldsymbol{a}_j | \boldsymbol{\xi}_j + \boldsymbol{s}_j), \quad (17) \\
p(\boldsymbol{\rho} | \boldsymbol{\eta} + \boldsymbol{u}) &= \mathrm{Dirichlet}(\boldsymbol{\rho} | \boldsymbol{\eta} + \boldsymbol{u}), \quad (18) \\
p(\boldsymbol{\pi} | \boldsymbol{\zeta} + \boldsymbol{v}) &= \mathrm{Dirichlet}(\boldsymbol{\pi} | \boldsymbol{\zeta} + \boldsymbol{v}), \quad (19)
\end{aligned}$$

where $\boldsymbol{s}_j = [s_{j1}, \ldots, s_{jK}]^T$, $\boldsymbol{u} = [u_{-G}, \ldots, u_G]^T$, and $\boldsymbol{v} = [v_1, \ldots, v_K]^T$.

3.3.3 Learning Cauchy Parameters c and d

To estimate the parameters c and d, we use the MH algorithm. It is hard to analytically calculate the posterior distributions of c and d because a Cauchy distribution doesn't have conjugate prior distributions. When the values of c and d are respectively c_i and d_i, we define proposal distributions of c and d as follows:

$$\begin{aligned}
q_c(c | c_i) &= \mathrm{Gamma}(c | \gamma c_i, \gamma), \quad (20) \\
q_d(d | d_i) &= \mathrm{Gamma}(d | \delta d_i, \delta), \quad (21)
\end{aligned}$$

where γ and δ are hyperparameters of the proposal distributions. Using the value of c^* sampled from $q_c(c|c_i)$, we calculate the acceptance ratio given by

$$g_c(c^*, c_i) = \min\left\{1, \frac{f_c(c^*) q_c(c_i | c^*)}{f_c(c_i) q_c(c^* | c_i)}\right\}, \quad (22)$$

where $f_c(c)$ is a likelihood function given by

$$\begin{aligned}
f_c(c) &\equiv p(c | x_{1:T}, z_{1:N}, \tau_{1:N}, \Theta, d_i) \\
&\propto \prod_{n=1}^{N} \rho_n b_{n z_n \tau_{n-1} \tau_n} \prod_{n=2}^{N} a_{z_{n-1} z_n} \pi_{z_1} q(c) \\
&= \prod_{n=1}^{N} \rho_n \left\{ \prod_{t=\phi_{n-1}}^{\phi_n - 1} \mathrm{Cauchy}(x_t | \mu_{z_n}, \lambda_t^c) \right\}^{\frac{1}{\phi_n - \phi_{n-1}}} \\
&\quad \cdot \prod_{n=2}^{N} a_{z_{n-1} z_n} \pi_{z_1} \mathrm{Gamma}(c | c_0, c_1), \quad (23) \\
\lambda_t^c &= c_i \cdot \Delta x_t + d_i, \quad (24)
\end{aligned}$$

Finally, if the value of $g_c(c^*, c_i)$ is larger than the random number r sampled from a uniform distribution on the interval $[0, 1]$, then $c_{i+1} = c^*$, and otherwise $c_{i+1} = c_i$, where c_0 is sampled from the prior distribution $q(c)$.

The value of d is updated in the same way as that of c. Using the value of d^* sampled from $q_d(d|d_i)$, we calculate the acceptance criteria given by

$$g_d(d^*, d_i) = \min\left\{1, \frac{f_d(d^*) q_d(d_i | d^*)}{f_d(d_i) q_d(d^* | d_i)}\right\}, \quad (25)$$

where $f_d(d)$ is a likelihood function given by

$$
\begin{aligned}
f_d(d) &\equiv p(d|x_{1:T}, z_{1:N}, \tau_{1:N}, \Theta, c_{i+1}) \\
&\propto \prod_{n=1}^{N} \rho_n b_{n z_n \tau_{n-1} \tau_n} \prod_{n=2}^{N} a_{z_{n-1} z_n} \pi_{z_1} q(c) \\
&= \prod_{n=1}^{N} \rho_n \left\{ \prod_{t=\phi_{n-1}}^{\phi_n - 1} \mathrm{Cauchy}(x_t|\mu_{z_n}, \lambda_t^d) \right\}^{\frac{1}{\phi_n - \phi_{n-1}}} \\
&\quad \cdot \prod_{n=2}^{N} a_{z_{n-1} z_n} \pi_{z_1} \mathrm{Gamma}(d|d_0, d_1), \qquad (26)
\end{aligned}
$$

$$\lambda_t^d = c_{i+1} \cdot \Delta x_t + d_i, \qquad (27)$$

Finally, if the value of $g_d(d^*, d_i)$ is larger than the random number r sampled from a uniform distribution on the interval $[0, 1]$, then $d_{i+1} = d^*$, and otherwise $d_{i+1} = d_i$, where d_0 is sampled from the prior distribution $q(d)$.

3.4 Viterbi Decoding

A latent sequence of musical notes is estimated by using the Viterbi algorithm that uses the parameters at the time when the likelihood given by

$$p(x_{1:T}) = \sum_{k=1}^{K} \sum_{g=-G}^{G} \alpha_{nkg} \qquad (28)$$

is the maximum in the learning process. The musical notes that we want to estimate are the latent variables that maximize the value given by $p(\boldsymbol{Z}, \boldsymbol{\tau}|\boldsymbol{X})$. In the Viterbi algorithm, we define ω_{nkg} as follows:

$$
\begin{aligned}
&\omega_{nkg} = \\
&\max_{\substack{z_{1:n-1} \\ \tau_{1:n-1}}} \ln p(X_{10\tau_1}, \ldots, X_{n\tau_{n-1}g}, z_{1:n-1}, z_n{=}k, \tau_{1:n-1}, \tau_n{=}g), \\
&\qquad (29)
\end{aligned}
$$

and ω_{nkg} is calculated recursively with the equations

$$\omega_{1kg} = \ln \rho_g + \ln b_{1k0g} + \ln \pi_k, \qquad (30)$$

$$\omega_{nkg} = \ln \rho_g + \max_f \left[\ln b_{nkfg} + \max_j \{ \ln a_{jk} + \omega_{(n-1)jf} \} \right]. \qquad (31)$$

In the recursive calculation of ω_{nkg}, when the states that maximize the value of ω_{nkg} are $z_{n-1} = j, \tau_{n-1} = f$, those states are memorized as $h_{nk}^{(z)} = j, h_{ng}^{(\tau)} = f$. After calculating $\{\omega_{Nkg}\}_{k=1,g=-G}^{K,G}$ with Eqs. (30) and (31), the sequence of latent variables $\{z_n, \tau_n\}_{n=1}^{N}$ is recursively estimated with the equations given by

$$(z_N, \tau_N) = \arg\max_{k,g} \{\omega_{nkg}\}, \qquad (32)$$

$$z_n = h_{(n+1) z_{n+1}}^{(z)}, \qquad (33)$$

$$\tau_n = h_{(n+1) \tau_{n+1}}^{(\tau)}. \qquad (34)$$

The note sequence is retrieved by revising the onset deviations represented by the estimated latent variables $\{\tau_n\}_{n=1}^{N}$.

Model	Concordance rate
SBS-HMM	66.3 ± 1.0
Majority vote	56.9 ± 1.1
Frame-based HMM	56.1 ± 1.1
BS-HMM	67.0 ± 1.0

Table 1: Average concordance rates and their standard errors

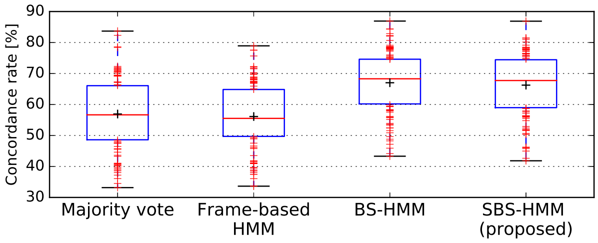


Figure 3: Concordance rates [%]. In the box plots, the red line indicates the median, the blue box indicates the range from the first to third quantile, the black cross indicates the mean, and the outliers are plotted with red crosses.

4. EVALUATION

We conducted an experiment to evaluate the proposed and previous methods in the accuracy of estimating musical notes from vocal F0 trajectories.

4.1 Experimental Conditions

The 100 pieces of popular music in RWC database [6] were used for the experiments. For each song, we trained model parameters, estimated the sequence of musical notes, and measured the accuracy of estimated musical notes. The input F0 trajectories were obtained from monaural music acoustic signals by the method of Ikemiya et al. [10]. We used the beat times obtained by a beat tracking system by Durand et al. [3]. This system estimates the beat times in units of a whole note, and the interval between adjacent beat times were divided into 16 equal intervals to obtain the beat times for 16th-note units were calculated.

For the proposed method, the sequence of musical notes was estimated with the Viterbi algorithm. The hyperparameters were $\boldsymbol{\xi} = \mathbb{1}, \boldsymbol{\zeta} = \mathbf{1}, \boldsymbol{\eta} = \mathbf{1}$, $c_0 = d_1 = d_0 = d_1 = 1$, where $\mathbb{1}$ and $\mathbf{1}$ respectively represent the matrix and vector whose elements are all ones. The parameters of the proposal distributions were $\gamma = \delta = 1$. The maximum value G that τ_n could take was $G = 5$ (i.e., 50 cents).

A majority-vote method was tested as a baseline. It estimates a musical note in each time unit corresponding to a 16th note by taking a majority vote for vocal F0s in the time unit. For comparison, a frame-based HMM and a beat-synchronous HMM (BS-HMM) were also tested. The frame-based HMM assumes that all beat intervals have only on frame. The BS-HMM is the same as SBS-HMM except that the onset deviation is not considered.

The estimated sequence of musical notes was compared with the ground-truth MIDI data synchronized to the vocal melody, and the concordance rate, i.e., the rate of frames in which pitches are correctly estimated, was used as the evaluation measure.

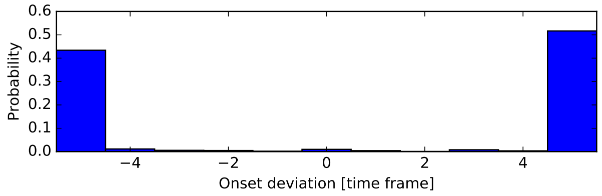


Figure 4: Example of a learned distribution of the model parameter $\boldsymbol{\rho}$.

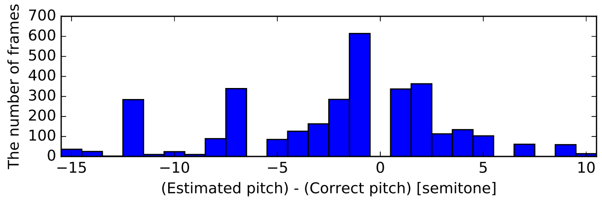


Figure 5: A example of pitch estimation error. The case that an estimated pitch is equal to a correct pitch is omitted.

4.2 Experimental Results

The results of note estimation are listed in Table 1 and Figure 3. The proposed model clearly outperformed the majority-vote method and the frame-based HMM in the average concordance rate. On the other hand, the average concordance rates for the proposed model and BS-HMM were similar and the difference was not statistically significant.

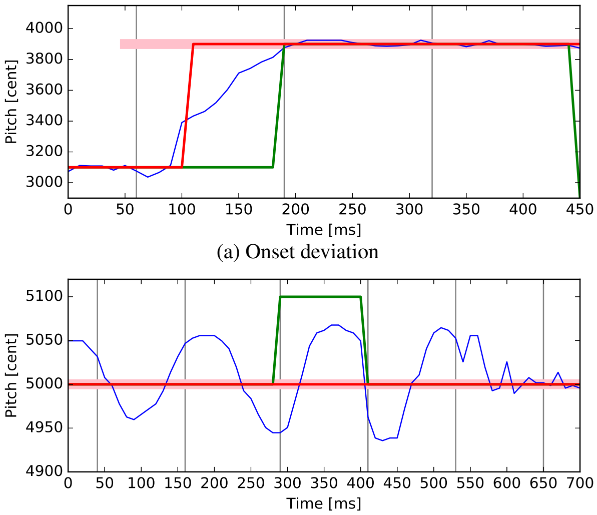
The results indicate that the model of music scores by transition probabilities and that of frequency deviations by output probabilities are both effective for improving the accuracy of musical note estimation. The cause of the result that the model of onset deviations did not improve the accuracy is probably that the model parameter ρ was not properly learned (Fig. 4). Capturing onset deviations by a single categorical distribution would be difficult, since onset depends on the duration of the pitches on either side of the onset, and on the over all tempo of the song. It would be necessary to model onset deviations in detail, for example by using a separate hidden state to represent the F0s during pitch transition.

4.2.1 Pitch Estimation Error

Two types of errors were mainly observed (Fig. 5). The first type was caused by singing styles of singers, and appears as errors that span one semitone and two semitones. This means that the frequency deviations affect the accuracy of note estimation. The second type was caused by the error of F0 estimation, and appears as the errors that span seven semitones and twelve semitones. The intervals of seven semitones and 12 semitones correspond to a perfect fifth and an octave, respectively.

4.2.2 Singing Style Extraction and Robustness

Example results of note estimation in Figure 6 show the potential of the proposed model to capture the singers' singing style. In the upper figure, the onset at the first beat is significantly delayed from the beat time. Whereas the proposed model correctly detected the delayed onset, the majority-vote method mis-identified the beat position of the pitch onset. The lower figure is an example of vibrato. With the majority-vote method, the estimation result was affected by the large frequency deviation. With the proposed method, on the other hand, the robustness due to the Cauchy distribution enabled the correct estimation of the pitch without being affected by the vibrato.



(a) Onset deviation

(b) Frequency deviation

Figure 6: Examples of note estimation results. The pink, blue, green, red, and black vertical lines respectively indicate a MIDI note which is the ground-truth, the F0 trajectory of a singing voice including onset deviations, the pitches estimated by the majority-vote method, the pitches estimated by the proposed method, and the beat times estimated in advance.

5. CONCLUSION

This paper presented a method for estimating the musical notes of music from the vocal F0 trajectory. When modeling the process generating the vocal F0 trajectory, we considered not only the musical score component but also onset deviation and frequency deviation. The SBS-HMM estimated pitches more accurately than the majority-vote method and the frame-based method.

The onset deviation and frequency deviation that were obtained using the proposed method are important for grasping the characteristics of singing expression. Future work includes precise modeling of vocal F0 trajectories based on second-order filters and extraction of individual singing expression styles. In the proposed method, F0 estimation, beat tracking, and musical note estimation are conducted separately. It is necessary to integrate these methods. The proposed method cannot deal with the non-vocal regions in actual music, so we plan to also appropriately deal with non-vocal regions.

Acknowledgement: This study was partially supported by JST OngaCREST Project, JSPS KAKENHI 24220006, 26700020, 26280089, 16H01744, and 15K16054, and Kayamori Foundation.

6. REFERENCES

[1] A. de Cheveigné and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[2] A. Dobashi, Y. Ikemiya, K. Itoyama, and K. Yoshii. A music performance assistance system based on vocal, harmonic, and percussive source separation and content visualization for music audio signals. *SMC*, 2015.

[3] S. Durand, J. P. Bello, B. David, and G. Richard. Downbeat tracking with multiple features and deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 409–413. IEEE, 2015.

[4] J.-L. Durrieu, G. Richard, B. David, and C. Févotte. Source/filter model for unsupervised main melody extraction from polyphonic audio signals. *IEEE Transactions on Audio, Speech, and Language Processing,*, 18(3):564–575, 2010.

[5] M. Goto. PreFEst: A predominant-F0 estimation method for polyphonic musical audio signals. *Proceedings of the 2nd Music Information Retrieval Evaluation eXchange*, 2005.

[6] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: Popular, classical and jazz music databases. In *The International Society for Music Information Retrieval (ISMIR)*, volume 2, pages 287–288, 2002.

[7] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T Nakano. Songle: A web service for active music listening improved by user contributions. In *The International Society for Music Information Retrieval (ISMIR)*, pages 311–316, 2011.

[8] D. J. Hermes. Measurement of pitch by subharmonic summation. *The journal of the acoustical society of America*, 83(1):257–264, 1988.

[9] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson. Singing-voice separation from monaural recordings using robust principal component analysis. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 57–60. IEEE, 2012.

[10] Y. Ikemiya, K. Yoshii, and K. Itoyama. Singing voice analysis and editing based on mutually dependent F0 estimation and source separation. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 574–578. IEEE, 2015.

[11] Y. E. Kim and B. Whitman. Singer identification in popular music recordings using voice coding features. In *Proceedings of the 3rd International Conference on Music Information Retrieval*, volume 13, page 17, 2002.

[12] A. Laaksonen. Automatic melody transcription based on chord transcription. In *The International Society for Music Information Retrieval (ISMIR)*, pages 119–124, 2014.

[13] Y. Li and D. Wang. Separation of singing voice from music accompaniment for monaural recordings. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1475–1487, 2007.

[14] M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J Salamon, J. Dai, J. Bello, and S Dixon. Computer-aided melody note transcription using the Tony software: Accuracy and efficiency. In *Proceedings of the First International Conference on Technologies for Music Notation and Representation - TENOR2015*, pages 23–30, Paris, France, 2015.

[15] M. Mauch and S. Dixon. pYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663. IEEE, 2014.

[16] Y. Ohishi, H. Kameoka, D. Mochihashi, and K. Kashino. A stochastic model of singing voice F0 contours for characterizing expressive dynamic components. In *INTERSPEECH*, pages 474–477, 2012.

[17] R. P. Paiva, T. Mendes, and A. Cardoso. On the detection of melody notes in polyphonic audio. In *The International Society for Music Information Retrieval (ISMIR)*, pages 175–182, 2005.

[18] G. E. Poliner and D. P. W. Ellis. A classification approach to melody transcription. *The International Society for Music Information Retrieval (ISMIR)*, pages 161–166, 2005.

[19] C. Raphael. A graphical model for recognizing sung melodies. In *The International Society for Music Information Retrieval (ISMIR)*, pages 658–663, 2005.

[20] M. Ryynänen, T. Virtanen, J. Paulus, and A. Klapuri. Accompaniment separation and karaoke application based on automatic melody transcription. In *2008 IEEE International Conference on Multimedia and Expo*, pages 1417–1420. IEEE, 2008.

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

[22] J. Salamon and E. Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1759–1770, 2012.

[23] W.-H. Tsai and H.-M. Wang. Automatic singer recognition of popular music recordings via estimation and modeling of solo vocal signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):330–341, 2006.