A FEEDBACK FRAMEWORK FOR IMPROVED CHORD RECOGNITION BASED ON NMF-BASED APPROXIMATE NOTE TRANSCRIPTION

Satoshi Maruo[1] *Kazuyoshi Yoshii*[1] *Katsutoshi Itoyama*[1] *Matthias Mauch*[2] *Masataka Goto*[3]

[1]Graduate School of Informatics, Kyoto University [2]Queen Mary University of London
[3]National Institute of Advanced Industrial Science and Technology (AIST)

ABSTRACT

This paper presents a feedback framework that can improve chord recognition for music audio signals by performing approximate note transcription with Bayesian non-negative matrix factorization (NMF) using prior knowledge on chords. Although the names and note compositions of chords are intrinsically linked with each other (*e.g.*, C major chords are highly likely to include C, E, and G notes, and those notes are highly likely to be in C major chords), chord recognition and note transcription (multipitch analysis) have been studied independently. To solve this chicken-and-egg problem, our framework iterates chord recognition and approximate note transcription using each other's results. More specifically, we first perform approximate note transcription based on Bayesian NMF that forces basis spectra to respectively correspond to different semitone-level pitches covering the whole range. We then execute chord recognition based on Bayesian hidden Markov models (HMMs) that use chroma features obtained from the activation patterns of those pitches. To improve note transcription, we again perform Bayesian NMF that encourages certain kinds of pitches in each chord region to be activated. Experimental results showed that our feedback framework gradually improved the accuracy of chord recognition.

Index Terms— Chord recognition, note transcription, Bayesian inference, nonnegative matrix factorization (NMF), hidden Markov model (HMM).

1. INTRODUCTION

Automatic chord recognition for music audio signals is one of the most fundamental tasks in the field of music information processing [1, 2], in part because the chord patterns used in musical pieces are clues useful in composer identification [3] and genre classification [4]. And because chord patterns are closely related to the musical mood, automatic chord recognition is indispensable for finding users' favorite pieces in large music collections [5].

Conventional methods of chord recognition generally consist of two parts: extraction of acoustic feature vectors and classification of those vectors. The most popular way of feature extraction is to calculate a 12-dimensional chroma vector at each frame or short segment (*e.g.*, half beat) [6]. The chroma vector represents an energy distribution over the twelve traditional pitch classes C, C#, D, D#, ..., B. The chroma vectors extracted from a region of a C major chord, for example, tend to take large values in the dimensions corresponding to pitch classes C, E, and G. A standard way of feature classification is to use a hidden Markov model (HMM) that represents the transition probabilities between adjacent chords and the emission probabilities of chroma vectors for each kind of chords [7–9].

This study was partially supported by JSPS KAKENHI 26700020, 24220006, 24700168 and CREST OngaCREST project.

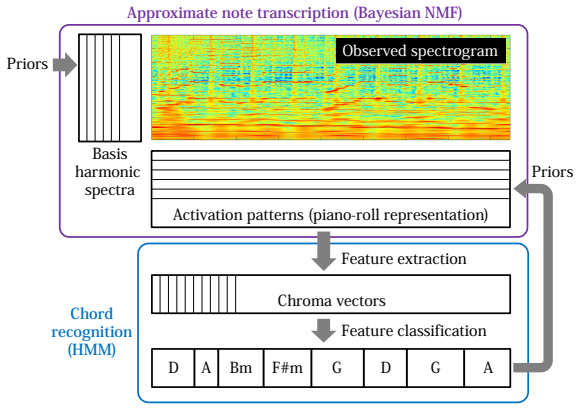

Fig. 1. Overview of our feedback framework combining chord recognition with approximate note transcription.

The main limitation of most conventional methods of chord recognition is that they extract shallow chroma vectors from music audio signals without performing any note transcription or multipitch analysis. Since the names and note compositions of chords are intrinsically linked with each other (*e.g.*, a C major chord is highly likely to include musical notes with pitch classes C, E, and G, and vice versa), a few studies have focused on the connection between chord recognition and note transcription. Sumi *et al.* [10] improved chord recognition by using a multipitch analysis method called PreFEst [11] to estimate the pitch trajectory of a bass line and using that trajectory as a clue when estimating the root notes of chords. Mauch and Dixon [12] calculated reliable chroma vectors insensitive to the energy of overtones by using approximate note transcription based on nonnegative least squares (NNLS), and Raczyński *et al.* [13] improved note transcription by considering a sequence of chords as prior knowledge about pitch distributions. Note that those studies have dealt with only *one-way* dependency between chord recognition and note transcription.

In this paper we propose a feedback framework that can improve chord recognition by focusing on the *mutual* dependency of chord recognition and note transcription (Fig. 1). More specifically, a target music audio signal is first transcribed into a piano-roll representation via Bayesian nonnegative matrix factorization (NMF) that forces basis spectra to have harmonic structures corresponding to different semitone-level pitches. Overtone-insensitive chroma vectors are obtained from the piano roll as in [12], and then a chord sequence is estimated by using Bayesian HMMs. A key feature of our framework is to again perform Bayesian NMF that encourages particular kinds of pitches (*e.g.*, C, E, and G) to be activated in each chord region (*e.g.*, C major). This feedback enables us to calculate more reliable chroma vectors from the refined piano roll.

2. RELATED WORK

This section introduces related work on chord recognition in terms of feature extraction and classification.

2.1. Feature extraction

The basic method of calculating the chroma vector is to accumulate the energy of frequency bins corresponding to each pitch class [6]. Since this method cannot distinguish the energy of fundamental frequencies (F0s) from that of harmonic partials, the obtained chroma vector does not precisely represent the energy distribution over the twelve pitch classes. Several methods for reducing the power of overtones before calculating chroma vectors have been proposed. Lee [14], for example, proposed a harmonic product spectrum (HPS), whereas Mauch and Dixon [12] used approximate note transcription based on nonnegative least squares (NNLS). Saito *et al.* [15] used specmurt analysis for roughly extracting the power of F0s by assuming a common harmonic structure, and Ueda *et al.* [16] reduced the power of percussive components by using a harmonic/percussive sound separation (HPSS) method.

Several variants of chroma vectors have been proposed. Müller and Ewert [17] proposed a CRP chromagram that is insensitive to the pitches and timbres of music audio signals. Ni *et al.* [18] proposed a loudness-based chromagram that takes into account the fact that perception of loudness is not linearly proportional to the power or amplitude spectrum, and Harte *et al.* [19] proposed a method of tonal centroid transformation that converts a 12-dimensional chroma vector into a 6-dimensional tonal-centroid vector.

2.2. Feature classification

HMMs can be used to model a vocabulary of chords as latent states; decoding such an HMM then corresponds to transcribing the optimal chord sequence. An important extension of this approach is to estimate chords and keys at the same time. Lee and Slaney [20] trained 24 key-specific HMMs corresponding to the 24 major/minor keys and selected the best model with high probability for a given audio signal. To deal with key changes, some studies tried to take into account the transition between adjacent keys [12,21,22]. Deep neural networks (DNNs) and recurrent neural networks (RNNs), which have significantly improved the accuracy of speech recognition, have recently been used for chord recognition [23,24]. This approach can unify feature extraction and classification into the same network.

3. PROPOSED METHOD

This section describes the proposed method of chord recognition based on approximate note transcription. Our method consists of Bayesian NMF-based feature extraction (inspired by [12]) and Bayesian HMM-based feature classification (inspired by [20]).

3.1. Problem specification

The goal of chord recognition is to transcribe a target music signal into a sequence of chord labels. Let $\hat{\boldsymbol{X}} = \{\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \cdots, \hat{\boldsymbol{x}}_T\}$ be a sequence of chroma vectors extracted from the target signal and $\hat{\boldsymbol{Z}} = \{\hat{z}_1, \hat{z}_2, \cdots, \hat{z}_T\}$ be a sequence of the corresponding chord labels, where T is the length of those sequences. The chord boundaries are determined by detecting chord change positions from $\hat{\boldsymbol{Z}}$. We aim to convert $\hat{\boldsymbol{X}}$ to $\hat{\boldsymbol{Z}}$ by using some kind of classifiers.

To train a statistical classifier, we use a chord-label-annotated music signal. Let $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T\}$ be a sequence of chroma vectors extracted from the music signal and $\boldsymbol{Z} = \{z_1, z_2, \cdots, z_T\}$ be a sequence of the corresponding chord labels. Both $\boldsymbol{X}$ and $\boldsymbol{Z}$ are given as the training data. For simplicity, we assume that the training data consists of only one musical piece having the same length as the sequence $\hat{\boldsymbol{X}}$ (the extension to dealing with multiple pieces is straightforward).

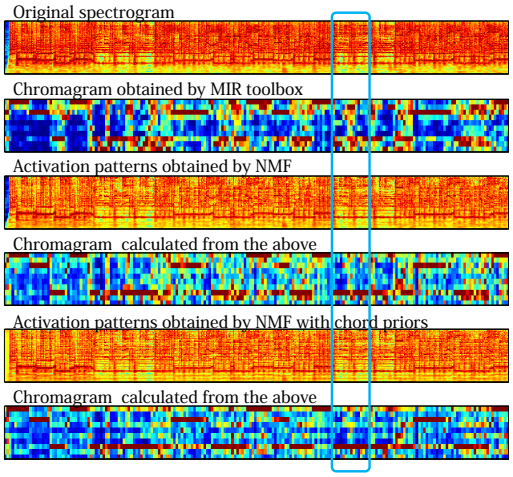

Fig. 2. Comparison of three variants of chromagrams: The bottom chromagram shows the clearest distributions over the 12 pitch classes.

The chord label is represented as a combination of a root note (C, C#, $\cdots$, B) and a type ("maj" or "min"). In addition, the special symbol N is used for representing "no chord." Let K be the size of the chord vocabulary ($K = 25$). Since the main focus of this paper is to show the effectiveness of mutually combining chord recognition with note transcription, we tackle the essential part of chord recognition by posing the following assumptions:

- Correct beat positions are given in advance. This assumption is not critical because many promising methods of beat tracking have recently been proposed [25,26].
- Chord boundaries are on beat positions (quarter-note level) or half-beat positions (eight-note level). This assumption holds true for the vast majority of popular music.
- Only two types of chords, "maj" and "min" chords, are taken into account. Other types of chords (*e.g.*, "maj7" and "dim") are forcibly categorized into those two types as in [27].
- The key of a musical piece remains the same from the beginning to the end (*i. e.*, key changes are not taken into account).

To perform reliable chord recognition, the tuning of each musical piece should be adjusted in advance as in [28].

3.2. Chord recognition based on Bayesian HMM

We explain our method of chord recognition that represents the generative process of chroma vectors by using Bayesian HMMs.

3.2.1. Feature extraction

We propose a method that extracts robust chroma features from temporal activation patterns of musical notes in a way similar to that in [12]. A key difference is that our method uses Bayesian NMF instead of NNLS because it enables us to take prior knowledge on harmonic structures and chord labels into account in a principled manner (see Section 3.3).

At each frame t, we calculate a 12-dimensional chroma vector $\boldsymbol{x}_t$ from the activation patterns of different pitches obtained

by Bayesian NMF (Fig. 2). In this paper the "frame" indicates a short half-beat segment. More specifically, we focus on 60 different pitches from C2 to B6 (MIDI note numbers from 36 to 95) in five octaves. To calculate the value of each dimension of $\boldsymbol{x}_t$, we accumulate the amplitude values of five pitches (*e.g.*, C2, C3, C4, C5, and C6) corresponding to the same pitch class (*e.g.*, C) at frame t. Finally, each dimension of chroma vectors is standardized over all frames such that the mean and variance of the dimension are equal to 0 and 1.

3.2.2. Model formulation

We propose a method that classifies chroma vectors by using 24 key-specific HMMs that respectively correspond to 24 major/minor keys in a way similar to that in [20]. A key difference is to formulate *Bayesian* HMMs for avoiding overfitting to the training data. Each HMM encodes the generative process of the observed data $\boldsymbol{X}$ via the latent variables $\boldsymbol{Z}$ under a condition that a particular key is assumed. For a set of model parameters, $\boldsymbol{\Theta}$ ($= \{\boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\}$ explained later), the joint probability over $\boldsymbol{X}$ and $\boldsymbol{Z}$ is given by

$$p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\Theta}) = p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{\Theta})p(\boldsymbol{Z}|\boldsymbol{\Theta}) = \prod_{t=1}^{T} p(\boldsymbol{x}_t|z_t)p(z_t|z_{t-1}), \quad (1)$$

where $p(z_t|z_{t-1})$ is the transition probability indicating how likely the previous chord z_{t-1} is to make a transition to the current chord z_n and $p(\boldsymbol{x}_t|z_t)$ is the emission probability indicating how likely the chroma vector $\boldsymbol{x}_t$ is to be generated from the chord z_n. $p(z_1|z_0) = p(z_1)$ is the initial state probability. More specifically, $p(z_t|z_{t-1})$ is represented as a discrete distribution as follows:

$$p(z_t = k'|z_{t-1} = k, \boldsymbol{\phi}_k) = \phi_{kk'}, \quad (2)$$

where $\boldsymbol{\phi}_k = [\phi_{k1}, \cdots, \phi_{kK}]^T$ is a set of transition probabilities that sum to unity ($1 \leq k \leq K$). Note that the transition probabilities $\boldsymbol{\phi}$ heavily depend on the key. $p(\boldsymbol{x}_t|z_t)$, on the other hand, is represented as a Gaussian mixture model (GMM) as follows:

$$p(\boldsymbol{x}_t|z_t = k, \boldsymbol{\pi}_k, \boldsymbol{\mu}_{kl}, \boldsymbol{\Lambda}_{kl}) = \sum_{l=1}^{L} \pi_{kl}\mathcal{N}(\boldsymbol{x}_t|\boldsymbol{\mu}_{kl}, \boldsymbol{\Lambda}_{kl}^{-1}), \quad (3)$$

where L is the number of Gaussians, $\boldsymbol{\pi}_k = [\pi_{k1}, \cdots, \pi_{kL}]^T$ is a set of mixing weights, and $\boldsymbol{\mu}_{kl}$ and $\boldsymbol{\Lambda}_{kl}$ are respectively the mean vector and precision matrix of the l-th component Gaussian. This GMM represents a probability distribution over chroma vectors of chord k ($1 \leq k \leq K$). To complete the Bayesian formulation, we put conjugate priors on unknown parameters $\boldsymbol{\Theta}$ as follows:

$$p(\boldsymbol{\phi}_k) = \mathcal{D}(\boldsymbol{\phi}_k|\boldsymbol{\alpha}_0), \quad p(\boldsymbol{\pi}_k) = \mathcal{D}(\boldsymbol{\pi}_k|\boldsymbol{\gamma}_0), \quad (4)$$

$$p(\boldsymbol{\mu}_{kl}, \boldsymbol{\Lambda}_{kl}) = \mathcal{N}(\boldsymbol{\mu}_{kl}|\boldsymbol{u}_0, (\beta_0\boldsymbol{\Lambda}_k)^{-1})\mathcal{W}(\boldsymbol{\Lambda}_{kl}|\boldsymbol{W}_0, \nu_0), \quad (5)$$

where $\mathcal{D}$ and $\mathcal{W}$ indicate the Dirichlet and Wishart distributions and $*_0$'s are hyperparameters given in advance; $\boldsymbol{\alpha}_0$ and $\boldsymbol{\gamma}_0$ are K-dimensional nonnegative vectors, $\boldsymbol{u}_0$ is a 12-dimensional vector, β_0 is a nonnegative scalar, $\boldsymbol{W}_0$ is a 12×12 scale matrix, and ν_0 is a degree of freedom. Using Eqs. (1)–(5), we obtain the complete probability over all random variables: $p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Theta}) = p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})p(\boldsymbol{Z}|\boldsymbol{\phi})p(\boldsymbol{\phi})p(\boldsymbol{\pi})p(\boldsymbol{\mu}, \boldsymbol{\Lambda})$.

3.2.3. Training

Given the training data $\boldsymbol{X}$ and $\boldsymbol{Z}$, we aim to calculate a posterior distribution over the model parameters, $p(\boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{X}, \boldsymbol{Z})$, according to the Bayes rule. In this setting, the posterior can be factorized as $p(\boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{X}, \boldsymbol{Z}) = p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{X}, \boldsymbol{Z})p(\boldsymbol{\phi}|\boldsymbol{Z})$. Using the conjugacy between Eq. (2) and Eq. (4), we obtain $p(\boldsymbol{\phi}|\boldsymbol{Z})$ in the same form as the prior distribution:

$$p(\boldsymbol{\phi}_k|\boldsymbol{Z}) = \mathcal{D}(\boldsymbol{\phi}_k|\boldsymbol{\alpha}_0 + \boldsymbol{n}_k), \quad (6)$$

where $\boldsymbol{n}_k$ is a K-dimensional vector in which each element $n_{kk'}$ is the number of transitions from chord k to chord k' in $\boldsymbol{Z}$.

Since the true posterior $p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{X}, \boldsymbol{Z})$ cannot be computed analytically, we use a variational Bayesian (VB) method [29] that approximates it as a factorized distribution as follows:

$$p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{X}, \boldsymbol{Z}) \approx q(\boldsymbol{\pi})q(\boldsymbol{\mu}, \boldsymbol{\Lambda}), \quad (7)$$

where two factors $q(\boldsymbol{\pi})$ and $q(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ are iteratively optimized such that the Kullback-Leibler divergence between $p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{X}, \boldsymbol{Z})$ and $q(\boldsymbol{\pi})q(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ is minimized. As a result, $q(\boldsymbol{\pi})$ and $q(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ are found to have the same forms as the corresponding priors:

$$q(\boldsymbol{\pi}_k) = \mathcal{D}(\boldsymbol{\pi}_k|\boldsymbol{\gamma}_0 + \boldsymbol{m}_k), \quad (8)$$

$$q(\boldsymbol{\mu}_{kl}, \boldsymbol{\Lambda}_{kl}) = \mathcal{N}(\boldsymbol{\mu}_{kl}|\boldsymbol{u}_{kl}, (\beta_{kl}\boldsymbol{\Lambda}_{kl})^{-1})\mathcal{W}(\boldsymbol{\Lambda}_{kl}|\boldsymbol{W}_{kl}, \nu_{kl}). \quad (9)$$

To optimize these parameters, we alternate VB-E and VB-M steps as in the expectation-maximization (EM) algorithm. Because of space limitation, we omit the updating formula (see Ch. 10 in [29]).

We utilize the circular characteristics of the twelve pitch classes to make the maximum use of $\boldsymbol{X}$ and $\boldsymbol{Z}$ for model training [16, 30].

- Training $p(\boldsymbol{\phi}|\boldsymbol{Z})$: The key underlying each chord z_t is transposed to C by shifting the root note of z_t. We thus have to train chord transitions for only two keys (C major and C minor). The other 22 key-specific HMMs can be obtained by permuting the elements of $\boldsymbol{\phi}$.
- Training $p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{X}, \boldsymbol{Z})$: The root note of each chord z_t is shifted to C by circular-shifting of the elements of the chroma vector x_t. We thus have to independently train GMMs for only two different "types" of chords (C "maj" and C "min"). The other 22 GMMs can be obtained by shifting the elements of the Gaussian parameters. Note that the 24 GMMs are shared over all the 24 key-specific HMMs.

3.2.4. Prediction

Using a trained HMM, we aim to obtain the optimal $\hat{\boldsymbol{Z}}$ maximizing a joint predictive probability given by

$$p(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Z}}|\boldsymbol{X}, \boldsymbol{Z}) = \int p(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Z}}|\boldsymbol{\Theta})p(\boldsymbol{\Theta}|\boldsymbol{X}, \boldsymbol{Z})d\boldsymbol{\Theta}. \quad (10)$$

An approximate solution could be obtained without integral computation by simply getting the optimal $\hat{\boldsymbol{\Theta}}$ that maximizes $p(\boldsymbol{\Theta}|\boldsymbol{X}, \boldsymbol{Z})$ and then finding the optimal $\hat{\boldsymbol{Z}}$ that maximizes $p(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Z}}|\hat{\boldsymbol{\Theta}})$ by using the Viterbi algorithm [31]. Instead, since $p(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Z}}|\boldsymbol{X}, \boldsymbol{Z})$ is more robust to outliers, we approximate it as a *predictive* HMM that consists of predictive transition and emission distributions (see Eqs. (2) and (3) for comparison) obtained by marginalizing out $\boldsymbol{\Theta}$ as follows:

$$p(\hat{z}_t = k'|\hat{z}_{t-1} = k, \boldsymbol{Z}) = \mathbb{E}[\phi_{kk'}], \quad (11)$$

$$p(\hat{\boldsymbol{x}}_t|\hat{z}_t = k, \boldsymbol{X}, \boldsymbol{Z}) = \sum_{l=1}^{L} \mathbb{E}[\pi_{kl}]\mathrm{St}(\hat{\boldsymbol{x}}_t|\boldsymbol{u}_{kl}, \boldsymbol{V}_{kl}, \nu_{kl} - 11), \quad (12)$$

where St indicates the Student-t distribution and $\boldsymbol{V}_{kl}$ is given by

$$\boldsymbol{V}_{kl} = \frac{(\nu_{kl} - 11)\beta_{kl}}{1 + \beta_{kl}}\boldsymbol{W}_{kl}. \quad (13)$$

To obtain the optimal $\hat{\boldsymbol{Z}}$, we can use the Viterbi algorithm. This decoding procedure is performed in parallel with respect to the 24 key-specific HMMs, and the HMM with the highest $p(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Z}}|\boldsymbol{X}, \boldsymbol{Z})$ is selected for estimating the key of $\hat{\boldsymbol{Z}}$.

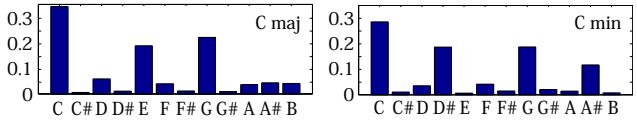

Fig. 3. Pitch-class distributions indicating how likely each pitch class is to be used in a C maj or C min chord.

3.3. Approximate note transcription based on Bayesian NMF

Here we explain Bayesian NMF for note transcription and how to incorporate prior knowledge on harmonic structures and chord labels.

3.3.1. Model formulation and Bayesian inference

NMF has been the most popular choice for source separation and multipitch analysis of music audio signals [32, 33]. It approximates a nonnegative matrix (constant-Q spectrogram) $\boldsymbol{X} \in \mathbb{R}^{M \times N}$ as the product of two nonnegative matrices $\boldsymbol{W} \in \mathbb{R}^{M \times R}$ and $\boldsymbol{H} \in \mathbb{R}^{R \times N}$ such that $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}$, where $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_R]$ is a set of R basis spectra over M frequency bins and $\boldsymbol{H} = [\boldsymbol{h}_1, \cdots, \boldsymbol{h}_R]^T$ is a set of the corresponding activation patterns over N frames. To evaluate the approximation error, we use the Kullback-Leibler (KL) divergence in an element-wise manner. Minimizing the KL divergence is equivalent to maximizing the Poisson likelihood given by

$$p\Big(X_{mn}\Big|\{w_{rm}, h_{rn}\}_{r=1}^{R}\Big) = \mathcal{P}\left(X_{mn}\middle|\sum_{r=1}^{R} w_{rm}h_{rn}\right). \quad (14)$$

To formulate a Bayesian model of NMF, we put independent gamma priors on individual elements of $\boldsymbol{W}$ and $\boldsymbol{H}$ as follows:

$$p(w_{rm}) = \mathcal{G}(w_{rm}|a_{rm}^{w}, b_{rm}^{w}), \quad (15)$$

$$p(h_{rn}) = \mathcal{G}(h_{rn}|a_{rn}^{h}, b_{rn}^{h}), \quad (16)$$

where a_* and b_* are the shape and rate hyperparameters.

Since the true posterior $p(\boldsymbol{W}, \boldsymbol{H}|\boldsymbol{X})$ cannot be computed analytically, we also use a VB method that approximates it as a factorized distribution as follows:

$$p(\boldsymbol{W}, \boldsymbol{H}|\boldsymbol{X}) \approx q(\boldsymbol{W})q(\boldsymbol{H}). \quad (17)$$

As a result, optimal $q(\boldsymbol{W})$ and $q(\boldsymbol{H})$ are found to have the same forms of the prior distributions:

$$q(w_{rm}) = \mathcal{G}(w_{rm}|a_{rm}^{w} + \textstyle\sum_n \lambda_{mnk} X_{mn}, b_{rm}^{w} + \sum_n \mathbb{E}[h_{rn}]),$$

$$q(h_{rn}) = \mathcal{G}(h_{rn}|a_{rn}^{h} + \textstyle\sum_m \lambda_{mnk} X_{mn}, b_{rn}^{h} + \sum_m \mathbb{E}[w_{rm}]),$$

where $\lambda_{mnk} \propto \exp(\mathbb{E}[\log w_{rm}] + \mathbb{E}[\log h_{rn}])$ is an auxiliary variable such that $\sum_k \lambda_{mnk} = 1$.

3.3.2. Hyperparameter adjustment

We adjust $p(\boldsymbol{W})$ such that basis spectra have harmonic structures corresponding to prefixed semitone-level pitches. We assume that the power of overtones decays exponentially with a decaying ratio of 0.5 in a typical harmonic structure. A hyperparameter vector $\boldsymbol{a}_k^w = [a_{k1}^w, \cdots, a_{kM}^w]^T$ over frequency bins is set to be proportional to the typical harmonic structure. Note that the larger the vector $\boldsymbol{a}_k^w$ is scaled, the stronger the impact of the prior is.

Similarly, we can adjust $p(\boldsymbol{H})$ such that particular kinds of pitch classes are encouraged to be activated in the region of each chord. As shown in Fig. 3, we learn how likely each pitch class is to be used in a C maj or C min chord by using music scores (MIDI files) with chord annotations in which the musical notes of each chord region are transposed such that the root note of the chord is C. A hyperparameter a_{rn}^h is set to be proportional to the probability of the corresponding pitch class.

Table 1. Experimental results.

Feature	Chroma	NMF	NMF with priors
Recognition rate (%)	69.0	71.6	71.9

4. EVALUATION

Here we describe our experiments evaluating the effectiveness of our framework.

4.1. Experimental conditions

We used the Beatles dataset [34], out of which 137 songs having major scales were used for 5-fold cross validation. We thus trained 12 key-specific HMMs corresponding to the 12 major keys in our experiments. The music audio signals were converted into constant-Q spectrograms with a frequency interval of 33.3 cents and a time interval of 64 ms. The hyperparameters were given by $L = 32$, $\boldsymbol{\alpha}_0 = \mathbf{1}$, $\boldsymbol{\gamma}_0 = \mathbf{1}$, $\boldsymbol{u}_0 = \mathbf{0}$, $\beta_0 = \nu_0 = 12$, $\boldsymbol{W}_0 = \boldsymbol{I}$, $R = 88$, and $a_{rm}^w = b_{rm}^w = a_{rn}^h = b_{rn}^h = 1$ if no prior knowledge was used. To evaluate the performance of chord recognition, we calculated the duration-based matching rate between the estimation results and the ground truth. For comparison, we tested a baseline method based on Bayesian HMMs trained by using chroma vectors [35] extracted via the MIR toolbox without any note transcription.

4.2. Experimental results

As shown in Table 1, the experimental results indicated the effectiveness of mutually combining chord recognition with approximate note transcription. We found that harmonic overtones were reduced by NMF, and the obtained chromagrams showed clearer pitch-class distributions than normal ones. Although the recognition rate was further improved by using prior knowledge about chords, the improvement was not as much as we had expected. Since relatively simple chords (*e.g.*, triad), which often appear in Beatles dataset, originally showed sparse chromagrams, the effectiveness of NMF was limited even if chord priors were used. The proposed method is expected to work more effectively for musical pieces with complex chords (*e.g.*, seventh and ninth).

If chord recognition and approximate note transcription are iterated in a mutually-dependent manner, the recognition rate is expected to be further improved. The computational time, however, is also increased in proportion to the number of iterations. To solve this problem, it is necessary to formulate a unified probabilistic model of chord recognition and note transcription.

5. CONCLUSION

This paper presented a feedback framework that combines chord recognition based on Bayesian HMMs with approximate note transcription based on Bayesian NMF. Those models can make use of each other's information. Experimental results showed the effectiveness of our framework, but there would be much room for significantly improving the chord recognition rate. To maximize the potential of the framework, we will try to appropriately adjust or automatically learn the hyperparameters of Bayesian NMF. On the other hand, the improvement of note transcription could be expected in our framework. We therefore plan to evaluate not only chord recognition but also note transcription. To guarantee the convergence theoretically, we aim to formulate a unified probabilistic model of chord recognition and note transcription that can be jointly optimized in a principled manner.

6. REFERENCES

[1] M. Mauch, *Automatic chord transcription from audio using computational models of musical context*, Ph.D. thesis, School of Electronic Engineering and Computer Science Queen Mary, University of London, 2010.

[2] C. Harte, *Towards automatic extraction of harmony information from music signals*, Ph.D. thesis, Department of Electronic Engineering, Queen Mary, University of London, 2010.

[3] M. Ogihara and T. Li, "N-gram chord profiles for composer style representation," in *ISMIR*, 2008, pp. 671–676.

[4] C. Pérez-Sancho, D. Rizo, and J. M. Inesta, "Genre classification using chords and stochastic language models," *Connection science*, vol. 21, no. 2-3, pp. 145–159, 2009.

[5] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano, "Songle: A web service for active music listening improved by user contributions," in *ISMIR*. Citeseer, 2011, pp. 311–316.

[6] T. Fujishima, "Realtime chord recognition of musical sound: A system using common lisp music," in *ICMC*, 1999, pp. 464–467.

[7] A. Sheh and D. P. W. Ellis, "Chord segmentation and recognition using EM-trained hidden Markov models," in *ISMIR*, 2003, pp. 185–191.

[8] K. Lee and M. Slaney, "A unified system for chord transcription and key extraction using hidden Markov models," in *ISMIR*. Citeseer, 2007, pp. 245–250.

[9] R. Chen, W. Shen, A. Srinivasamurthy, and P. Chordia, "Chord recognition using duration-explicit hidden Markov models," in *ISMIR*, 2012, pp. 445–450.

[10] K. Sumi, K. Itoyama, K. Yoshii, K. Komatani, T. Ogata, and H. G. Okuno, "Automatic chord recognition based on probabilistic integration of chord transition and bass pitch estimation," in *ISMIR*, 2008, pp. 39–44.

[11] M. Goto, "PreFEst: A predominant-F0 estimation method for polyphonic musical audio signals," *MIREX*, 2005.

[12] M. Mauch and S. Dixon, "Approximate note transcription for the improved identification of difficult chords," in *ISMIR*, 2010, pp. 135–140.

[13] S. A. Raczynski, E. Vincent, F. Bimbot, S. Sagayama, et al., "Multiple pitch transcription using DBN-based musicological models," in *ISMIR*, 2010, pp. 363–368.

[14] K. Lee, "Automatic chord recognition from audio using enhanced pitch class profile," in *ICMC*, 2006.

[15] S. Saito, H. Kameoka, K. Takahashi, T. Nishimoto, and S. Sagayama, "Specmurt analysis of polyphonic music signals," *IEEE Trans. on ALSP*, vol. 16, no. 3, pp. 639–650, 2008.

[16] Y. Ueda, Y. Uchiyama, T. Nishimoto, N. Ono, and S. Sagayama, "HMM-based approach for automatic chord detection using refined acoustic features," in *ICASSP*. IEEE, 2010, pp. 5518–5521.

[17] M. Muller, S. Ewert, and S. Kreuzer, "Making chroma features more robust to timbre changes," in *ICASSP*. IEEE, 2009, pp. 1877–1880.

[18] Y. Ni, M. McVicar, R. Santos-Rodriguez, and T. De Bie, "An end-to-end machine learning system for harmonic analysis of music," *IEEE Trans. on ASLP*, vol. 20, no. 6, pp. 1771–1783, 2012.

[19] C. Harte, M. Sandler, and M. Gasser, "Detecting harmonic change in musical audio," in *ACM*, 2006, pp. 21–26.

[20] K. Lee and M. Slaney, "Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio," *IEEE Trans. on ASLP*, vol. 16, no. 2, pp. 291–301, 2008.

[21] J. Pauwels and G. Peeters, "Segmenting music through the joint estimation of keys, chords and structural boundaries," in *ACM*, 2013, pp. 741–744.

[22] Y. Ueda, Y. Uchiyama, N. Ono, and S. Sagayama, "MIREX 2010: joint recognition of key and chord from music audio signals using key-modulation HMM," *MIREX*, 2010.

[23] E. J. Humphrey and J. P. Bello, "Rethinking automatic chord recognition with convolutional neural networks," in *ICMLA*. IEEE, 2012, vol. 2, pp. 357–362.

[24] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Audio chord recognition with recurrent neural networks," in *ISMIR*, 2013, pp. 335–340.

[25] S. Durand, B. David, and G. Richard, "Enhancing downbeat detection when facing different music styles," in *ICASSP*. IEEE, 2014, pp. 3132–3136.

[26] M. Khadkevich, T. Fillon, G. Richard, and M. Omologo, "A probabilistic approach to simultaneous extraction of beats and downbeats," in *ICASSP*. IEEE, 2012, pp. 445–448.

[27] L. Oudre, Y. Grenier, and C. Févotte, "Template-based chord recognition: Influence of the chord types," in *ISMIR*, 2009, pp. 153–158.

[28] C. Harte and M. Sandler, "Automatic chord identifcation using a quantised chromagram," in *AES*, 2005, pp. 245–250.

[29] Christopher M. Bishop et al., *Pattern recognition and machine learning*, vol. 1, springer New York, 2006.

[30] K. Itoyama, O. Tetsuya, and H. G. Okuno, "Automatic chord sequence recognition based on probabilistic integration of acoustic features and chord transition," in *IEA/AIE*, 2012.

[31] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *Information Theory, IEEE Trans. on*, vol. 13, no. 2, pp. 260–269, 1967.

[32] S. A. Raczyński, N. Ono, and S. Sagayama, "Multipitch analysis with harmonic nonnegative matrix approximation," in *ISMIR*. Citeseer, 2007.

[33] E. Vincent, N. Bertin, and R. Badeau, "Harmonic and inharmonic nonnegative matrix factorization for polyphonic pitch transcription," in *ICASSP*. IEEE, 2008, pp. 109–112.

[34] M. Mauch, C. Cannam, M. Davies, S. Dixon, C. Harte, S. Kolozali, D. Tidhar, and M. Sandler, "OMRAS2 metadata project 2009," in *ISMIR*, 2009.

[35] O. Lartillot and P. Toiviainen, "A Matlab toolbox for musical feature extraction from audio," in *DAFX*, 2007, pp. 237–244.