Generating Tablature of Polyphony Consisting of Melody and Bass Line

Shunsuke Sakai[1], Hinata Segawa[1] and Tetsuro Kitahara[1] *

College of Humanities and Sciences, Nihon University
{sakai, segawa, kitahara}@kthrlab.jp

Abstract. Our final goal is to develop a system that generates a tablature for a given lead sheet consisting of a melody and a chord progression. Generating a tablature for a lead sheet requires a complex solution search because plural possibilities exist in voicing each chord. As the first step, we address a system that generates a tablature consists of a melody and a bass line using the Viterbi algorithm. Polyphonic fingering states are modeled as 6-dimensional vectors and the playing difficulty is modeled as a cost function of such vectors. By minimizing the cost function, our system generates a playable tablature.

1 Introduction

Tablatures are helpful to play the guitar, so many non-professional guitarists use tablatures. Therefore, there have been attempts to automatically generate tablatures from audio signals or scores. Wiggins et al. used audio signals as inputs and estimated fingering positions using a neural network [1]. Yazawa et al. also used audio signals as inputs, and generated tablatures using fingering forms and note value-based costs [2]. Hori et al. proposed a web application that enables arrangement with transposition using hidden Markov models [3].

The solo guitar, which means playing both a melody and an accompaniment alone, is an attractive playing style for guitars, especially classic guitars. However, there are less commercially available tablatures for the solo guitar. Therefore, if one wants to play their favorite songs in the solo guitar, they must arrange those songs for the solo guitar. However, it is a difficult task for most amateur guitarists.

The final goal is to achieve a system that automatically generates such tablatures for the solo guitar. Inputs are assumed to be lead sheets, which describe melodies and chord progressions but no voicings. We need a complex solution search to generate tablatures from lead sheets because there are plural possibilities in the voicing of each chord. As the first step, we address a system that generates tablatures for simultaneously playing a melody and bass line.

* This work was supported by JSPS Kakenhi Nos. JP22H03711 and JP21H03572.

2 Proposed System

Given a lead sheet describing a melody and a chord progression, our system generates tablatures for simultaneously playing the melody and the bass line on the classic guitar.

2.1 Importing MusicXML data

Once a lead sheet given in the MusicXML format, a sequence of the melody notes and the bass notes is represented as:

$$X = \{(x_1, r_1), (x_2, r_2), \cdots, (x_N, r_N)\}$$

where x_n is the pitch (MIDI note number) of the n-th note and r_n is the chord's root note (pitch class from 0 to 11) when note x_n is being played. In the current implementation, the root note is played only at the timing of a chord change, and r_n is empty when there is no chord or the previous chord continues (represented by $r_n = \epsilon$).

2.2 Designing the Viterbi Algorithm

Given X, our system estimates the fingering positions for each element of X.

Set of fingering state vectors Let V be the set of vectors representing the playable fingering states. Each element $\boldsymbol{v}$ in V is represented by a 6-dimensional vector $\boldsymbol{v} = (f_1(\boldsymbol{v}), f_2(\boldsymbol{v}), f_3(\boldsymbol{v}), f_4(\boldsymbol{v}), f_5(\boldsymbol{v}), f_6(\boldsymbol{v}))$. Let $f_m(\boldsymbol{v})$ denote the fret number of string m ($f_m(\boldsymbol{v}) = -1, 0, \ldots, 14$), where $f_m(\boldsymbol{v}) = -1$ means not to play and $f_m(\boldsymbol{v}) = 0$ means open string. To ensure that V contains only playable fingering states, V only has elements that satisfy the following conditions.

1. $\max_m(f_m(\boldsymbol{v})) - \min_m(f_m(\boldsymbol{v})) \leq 3$,
2. The number of m satisfying $f_m(\boldsymbol{v}) > 0$ is 2 or less,
3. Multiple strings do not correspond to the same pitch.

Basic Mechanism Given $X = \{(x_1, r_1), \cdots\}$ representing the main melody (+ root notes of the chord progression), the system finds the optimal sequence of fingering states, $Q = \{\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_N\}$ ($\boldsymbol{q}_n \in V$) by minimizing the cost (degree of non-optimality) for Q. The cost is defined as a combination of the following three.

- Initial cost $C(\boldsymbol{q}_1)$: gives a slightly larger cost to the fret furthest from the neck, based on the idea that playing in a position closer to the neck is more common.
- Transition cost $C(\boldsymbol{q}_{n+1}|\boldsymbol{q}_n)$: Based on the idea that it is easier to play when the movement of the fingering position is less, a larger cost is given when the movement of the fingering position is larger.
- Emission cost $C((x_n, r_n)|\boldsymbol{q}_n)$: gives a sufficiently large cost if somebody cannot play the correct note at the given fingering position.

The total cost $C(Q)$ is defined by the following equation:

$$C(Q) = C(\boldsymbol{q}_1) + \left\{ \sum_{n=1}^{N-1} (C((x_n|r_n)|\boldsymbol{q}_n) + C(\boldsymbol{q}_{n+1}|\boldsymbol{q}_n)) \right\} + C((x_N|r_N)|\boldsymbol{q}_N)$$

and the Viterbi algorithm finds the minimum Q.

The initial cost, transition cost, and emission cost are defined as follows:

Initial cost On acoustic guitars, the closer-to-the-neck position is more common than the closer-to-the-body position. Therefore, we give the closer-to-the-body position a slightly higher cost than the closer-to-the-neck position. That is, the initial cost $C(\boldsymbol{q}_1)$ is defined as follows:

$$C(\boldsymbol{q}_1) = \begin{cases} 2.5 & (\max_m(f_m(\boldsymbol{q}_1) \leq 5) \\ 5.0 & (\text{otherwise}) \end{cases}$$

Transiton cost By giving higher costs to large movements in fingering-positions, we reduce the difficulty of playing the strings. In addition, for the same reason as above, we prioritize the position closest to the neck. Therefore, we divide the transition cost $C(\boldsymbol{q}_{n+1}|\boldsymbol{q}_n)$ into the cost of the move itself $C_1(\boldsymbol{q}_{n+1}|\boldsymbol{q}_n)$ and the cost to prioritize the neck side $C_2(\boldsymbol{q}_{n+1})$:

$$C(\boldsymbol{q}_{n+1}|\boldsymbol{q}_n) = C_1(\boldsymbol{q}_{n+1}|\boldsymbol{q}_n) + C_2(\boldsymbol{q}_{n+1})$$

$$C_1(\boldsymbol{q}_{n+1}|\boldsymbol{q}_n) = \begin{cases} 0.0 & (\text{dist}(\boldsymbol{q}_n, \boldsymbol{q}_{n+1}) \leq 3) \\ 5.0 & (\text{dist}(\boldsymbol{q}_n, \boldsymbol{q}_{n+1}) \leq 4) \\ 30.0 & (\text{otherwise}) \end{cases}$$

$$C_2(\boldsymbol{q}_{n+1}) = \begin{cases} 5.0 & (\max_m(\boldsymbol{q}_{n+1})) = 0) \\ 10.0 & (\max_m(\boldsymbol{q}_{n+1})) \leq 4) \\ 20.0 & (\text{otherwise}) \end{cases}$$

where $\text{dist}(\boldsymbol{v}_1, \boldsymbol{v}_2)$ $(\boldsymbol{v}_1, \boldsymbol{v}_2 \in V)$ is defined as follows:

$$\text{dist}(\boldsymbol{v}_1, \boldsymbol{v}_2) = |\max_m(f_m(\boldsymbol{v}_1)) - \max_m(f_m(\boldsymbol{v}_2))|$$

Emission cost The emission cost indicates whether the fingering position can produce the given note. Let $\text{note}(f_m(\boldsymbol{v}))$ be the pitch (MIDI note number) played at the fingering position $f_m(\boldsymbol{v})$ on string m. The fingering state $\boldsymbol{q}_n$ produces (x_n, r_n) when each of $m = 1, \cdots, 6$ satisfies one of the following:

- $f_m(\boldsymbol{q}_n) = -1$
- $\text{note}(f_m(\boldsymbol{q}_n)) = x_n$
- $\text{note}(f_m(\boldsymbol{q}_n)) = r_n + 12o$ (in the case of $r_n \neq \epsilon$)

o is an integer to change octaves in the range satisfying $r_n < x_n$. The emission cost $C((x_n, r_n)|\boldsymbol{q}_n)$ is represented as follows.

$$C((x_n, r_n)|\boldsymbol{q}_n) = \begin{cases} 0.0 & \text{(satisfying the above conditions)} \\ 10.0 & \text{(otherwise)} \end{cases}$$

2.3 Exporting Tablature

The notes and fingering positions obtained by the method described above are exported in the MusicXML format.

3 Preliminary Results

We tried to generate a tablature for a lead sheet shown in Figure 1. The tablature generated by the proposed system is shown in Figure 2. The melody in the lead sheet in Figure 1 was retained, and the root note of each chord was output. Because the initial and transition costs for open strings are relatively low, the generated tablature used open strings as much as possible.

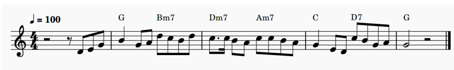

Fig. 1. Input lead sheet

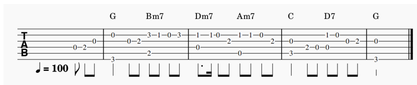

Fig. 2. Generated tablature through our system

4 Conclusion

In this paper, we proposed a system that generates a tablature containing a melody and bass line from a lead sheet. Our preliminary experiment shows that the system generated a tablature with which the player can play a melody and a bass line simulteneously. Future work will include the generation of more complex accompaniments such as arpeggio.

References

1. A. Wiggins and Y. Kim: Guitar Tablature Estimation with A Convolutional Neural Network. In Proceedings of ISMIR (2019).
2. K, Yazawa, K, Itoyama, and H. G. Okuno: Automatic Transcription of Guitar Tablature from Audio Signals in Accordance with Player's Prificiency. In Proceedings of ICASSP, IEEE (2014).
3. G, Hori and S, Sagayama: HMM-based Automatic Arrangement for Guitars with Transposition and its Implementation. In Proceedings of ICMC-SMC (2014).