MINIMAX VITERBI ALGORITHM FOR HMM-BASED GUITAR FINGERING DECISION

Gen Hori
Asia University / RIKEN
hori@brain.riken.jp

Shigeki Sagayama
Meiji University
sagayama@meiji.ac.jp

ABSTRACT

Previous works on automatic fingering decision for string instruments have been mainly based on path optimization by minimizing the difficulty of a whole phrase that is typically defined as the sum of the difficulties of moves required for playing the phrase. However, from a practical viewpoint of beginner players, it is more important to minimize the maximum difficulty of a move required for playing the phrase, that is, to make the most difficult move easier. To this end, we introduce a variant of the Viterbi algorithm (termed the "minimax Viterbi algorithm") that finds the path of the hidden states that maximizes the minimum transition probability (not the product of the transition probabilities) and apply it to HMM-based guitar fingering decision. We compare the resulting fingerings by the conventional Viterbi algorithm and our proposed minimax Viterbi algorithm to show the appropriateness of our new method.

1. INTRODUCTION

Most string instruments have overlaps in pitch ranges of their strings. As a consequence, such string instruments have more than one way to play even a single note (except the highest and the lowest notes that are covered only by a single string) and thus numerous ways to play a whole song. That is why the fingering decision for a given song is not always an easy task for string players and therefore automatic fingering decision has been attempted by many researchers. Previous works on automatic fingering decision have been mainly based on path optimization by minimizing the difficulty level of a whole phrase that is typically defined as the sum or the product of the difficulty levels defined for each move. (The product of difficulty levels easily reduces to the sum of the logarithm of the difficulty levels and therefore the sum and the product do not make any essential difference.) However, whether a string player can play a passage using a specific fingering depends almost only on whether the most difficult move included in the fingering is playable. Especially, from a practical viewpoint of beginner players, it is most important to minimize the maximum difficulty level of a move included in a fingering, that is, to make the most difficult move easier.

The purpose of this paper is to introduce a variant of the Viterbi algorithm [12] termed the "minimax Viterbi algorithm" that finds the sequence of the hidden states that maximizes the minimum transition probability on the sequence (not the product of all the transition probabilities on the sequence) and apply it to HMM-based guitar fingering decision. We employ a hidden Markov model (HMM) whose hidden states are left hand forms of guitarists and output symbols are musical notes, and perform fingering decision by solving a decoding problem of HMM using our proposed minimax Viterbi algorithm for finding the sequence of hidden states with the maximum minimum transition probability. Because the transition probabilities are set to large for easy moves and small for difficult ones, resulting fingerings "make the most difficult move easier" as previously discussed in this section. To distinguish the original Viterbi algorithm and our variant, we refer to the former as the "conventional Viterbi algorithm" and to the latter as the "minimax Viterbi algorithm" throughout the paper.

As for automatic fingering decision, several attempts have been made in the last two decades. Sayegh [10] first formulated fingering decision of string instruments as a problem of path optimization. Radicioni et al. [8] extended Sayegh [10]'s approach by introducing segmentation of musical phrase. Radisavljevic and Driessen [9] introduced a gradient descent search for the coefficients of the cost function for path optimization. Tuohy and Potter [11] first applied the genetic algorithm (GA) to guitar fingering decision and arrangements. As for applications of HMM to fingering decision, Hori et al. [4] applied input-output HMM [2] to guitar fingering decision and arrangement, Nagata et al. [5] applied HMM to violin fingering decision, and Nakamura et al. [6] applied merged-output HMM to piano fingering decision. Comparing to those previous works, the present work is new in that it introduces "minimax paradigm" to automatic fingering decision.

The rest of the paper is organized as follows. Section 2 recalls the conventional Viterbi algorithm and introduces our proposed minimax Viterbi algorithm. Section 3 introduces a framework of HMM-based fingering decision for monophonic guitar phrases. Section 4 applies the minimax Viterbi algorithm to fingering decision for monophonic guitar phrases and evaluates the results. Section 5 concludes the paper and discusses related future works.

2. MINIMAX VITERBI ALGORITHM

We start by introducing our newly proposed "minimax Viterbi algorithm" on which we build our fingering decision method in the following section. First of all, we recall the definition of HMM [1] and the procedure of the conventional Viterbi algorithm for finding the sequence of hidden states that gives the maximum likelihood. Next, we modify the algorithm to our new one for finding the sequence of hidden states that gives the maximum minimum transition probability.

2.1 Hidden Markov model (HMM)

Suppose that we have two finite sets of hidden states Q and output symbols O,

$$Q = \{q_1, q_2, \dots, q_N\},$$
$$O = \{o_1, o_2, \dots, o_K\},$$

and two sequences of random variables $\boldsymbol{X}$ of hidden states and $\boldsymbol{Y}$ of output symbols,

$$\boldsymbol{X} = (X_1, X_2, \dots, X_T),$$
$$\boldsymbol{Y} = (Y_1, Y_2, \dots, Y_T),$$

then a hidden Markov model M is defined by a triplet

$$M = (A, B, \pi)$$

where A is an $N \times N$ matrix of the transition probabilities,

$$A = (a_{ij}),\ a_{ij} \equiv a(q_i, q_j) \equiv P(X_t = q_j | X_{t-1} = q_i),$$

B an $N \times K$ matrix of the output probabilities,

$$B = (b_{ik}),\ b_{ik} \equiv b(q_i, o_k) \equiv P(Y_t = o_k | X_t = q_i),$$

and Π an N-dimensional vector of the initial distribution of hidden states,

$$\Pi = (\pi_i),\ \pi_i \equiv \pi(q_i) \equiv P(X_1 = q_i).$$

2.2 Conventional Viterbi algorithm

When we observe a sequence of output symbols [2]

$$\boldsymbol{y} = (y_1, y_2, \dots, y_T)$$

from a hidden Markov model M, we are interested in the sequence of hidden states

$$\boldsymbol{x} = (x_1, x_2, \dots, x_T)$$

that generates the observed sequence of output symbols $\boldsymbol{y}$ with the maximum likelihood,

$$\begin{aligned}\hat{\boldsymbol{x}}_{ML} &= \arg\max_{\boldsymbol{x}} P(\boldsymbol{y}, \boldsymbol{x} | M) \\ &= \arg\max_{\boldsymbol{x}} P(\boldsymbol{x} | M) P(\boldsymbol{y} | \boldsymbol{x}, M) \\ &= \arg\max_{\boldsymbol{x}} \prod_{t=1}^{T} (a(x_{t-1}, x_t) b(x_t, y_t)) \qquad (1)\end{aligned}$$

where we write $a(x_0, x_1) = \pi(x_1)$ for convenience. The problem of finding the maximum likelihood sequence $\hat{\boldsymbol{x}}_{ML}$ is called "decoding problem." Although an exhaustive search requires iterations over the N^T possible sequences, we can solve the problem efficiently using the Viterbi algorithm [12] based on dynamic programming (DP), which uses two $N \times T$ tables $\Delta = (\delta_{it})$ of maximum likelihood and $\Psi = (\psi_{it})$ of back pointers and the following four steps.

Initialization initializes the first columns of the two tables Δ and Ψ using the following formulae for $i = 1, 2, \dots, N$,

$$\delta_{i1} = \pi_i\, b(q_i, y_1),$$
$$\psi_{i1} = 0.$$

Recursion fills out the rest columns of Δ and Ψ using the following recursive formulae for $j = 1, 2, \dots, N$ and $t = 1, 2, \dots, T–1$,

$$\delta_{j,t+1} = \max_i(\delta_{it} a_{ij})\, b(q_j, y_{t+1}),$$
$$\psi_{j,t+1} = \arg\max_i(\delta_{it} a_{ij}).$$

Termination finds the index of the last hidden state of the maximum likelihood sequence $\hat{\boldsymbol{x}}_{ML}$ using the last column of the table Δ,

$$i_T = \arg\max_i \delta_{iT}.$$

Backtracking tracks the indices of the hidden states of the maximum likelihood sequence $\hat{\boldsymbol{x}}_{ML}$ from the last to the first using the back pointers of Ψ for $t = T, T–1, \dots, 2$,

$$i_{t-1} = \psi_{i_t, t}$$

from which $\hat{\boldsymbol{x}}_{ML}$ is obtained as

$$x_t = q_{i_t} \quad (t = 1, 2, \dots, T).$$

2.3 Modification for minimax Viterbi algorithm

Next, we consider the problem of finding the sequence of hidden states $\boldsymbol{x}$ with the maximum minimum transition probability [3],

$$\hat{\boldsymbol{x}}_{MM} = \arg\max_{\boldsymbol{x}} \min_{1 \le t \le T} (a(x_{t-1}, x_t) b(x_t, y_t)), \qquad (2)$$

which we call "minimax decoding problem [4]." A naive approach to the problem is an exhaustive search, that is, to enumerate all the sequences of the N hidden states and the length T, calculate the minimum transition probability for all the sequences, and find the one with the maximum value, which involves iterations over N^T sequences and is not for an actual implementation. Instead, we introduce a

[1] See [7] for more tutorial on HMM and its applications.

[2] According to the conventional notation of the probability theory, we denote random variables by uppercase letters and corresponding realizations by lowercase letters.

[3] Because the output probabilities are 0 or 1 in our application of HMM to guitar fingering decision, the sequence $\hat{\boldsymbol{x}}_{MM}$ eventually becomes the one with the maximum minimum transition probability, although its definition (2) depends on the output probabilities as well.

[4] Although the antonym "maximin" is appropriate for probability (which is the reciprocal of difficulty), we still use "minimax" for our proposed algorithm because it is appropriate for difficulty and conveys our concept of "make the most difficult move easier."

variant of the conventional Viterbi algorithm that can solve the problem efficiently. We modify the second step of the conventional Viterbi algorithm by replacing the term $\delta_{it}a_{ij}$ with $\min(\delta_{it}, a_{ij})$ where

$$\min(\delta_{it}, a_{ij}) = \begin{cases} \delta_{it} & (\delta_{it} \le a_{ij}) \\ a_{ij} & (a_{ij} < \delta_{it}) \end{cases}.$$

The modified second step is as follows.

Recursion for minimax Viterbi algorithm fills out the two tables Δ and Ψ using the following recursive formulae for $j = 1, 2, \dots, N$ and $t = 1, 2, \dots, T-1$,

$$\delta_{j,t+1} = \max_i(\min(\delta_{it}, a_{ij}))\, b_j(y_{t+1}),$$
$$\psi_{j,t+1} = \arg\max_i(\min(\delta_{it}, a_{ij})).$$

We modify only the second step and leave other steps unchanged. The modified second step works as the original one but now the element δ_{it} keeps the value of the maximum minimum transition probability of the subsequence of hidden states for the first t observations. The term $\min(\delta_{it}, a_{ij})$ updates the value of the minimum transition probability as the term $\delta_{it}a_{ij}$ in the conventional Viterbi algorithm does the likelihood [5].

3. FINGERING DECISION BASED ON HMM

We implement automatic fingering decision based on an HMM whose hidden states are left hand forms and output symbols are musical notes played by the left hand forms. In this formulation, fingering decision is cast as a decoding problem of HMM where a fingering is obtained as a sequence of hidden states. Because each hidden state has a unique output symbol, the output probability for the unique symbol is always 1. To compare the results of the conventional Viterbi algorithm and the minimax Viterbi algorithm clearly, we concentrate on fingering decisions for monophonic guitar phrases in the present study although HMM-based fingering decision is able to deal with polyphonic songs as well.

3.1 HMM for monophonic fingering decision

To play a single note with a guitar, a guitarist depresses a string on a fret with a finger of the left hand and picks the same string with the right hand. Therefore a form q_i for playing a single note can be expressed in a triplet

$$q_i = (s_i, f_i, h_i)$$

where $s_i = 1, \dots, 6$ is a string number (from the highest to the lowest), $f_i = 0, 1, \dots$ is a fret number, and $h_i = 1, \dots, 4$ is a finger number of the player's left hand (1,2,3 and 4 are the index, middle, ring and pinky fingers). The fret number $f_i = 0$ means an open string for which the finger number h_i does not make sense. For a classical guitar with six strings and 19 frets, the total number of forms is $6 \times (19 \times 4 + 1) = 462$ [6]. For the standard tuning (E_4-B_3-G_3-D_3-A_2-E_2), the MIDI note numbers of the open strings are

$$n_1 = 64, n_2 = 59, n_3 = 55, n_4 = 50, n_5 = 45, n_6 = 40$$

from which the MIDI note number of the note played by the form q_i is calculated as

$$note(q_i) = n_{s_i} + f_i.$$

3.2 Transition and output probabilities

In standard applications of HMM, model parameters such as the transition probabilities and the output probabilities are estimated from training data using the Baum-Welch algorithm [1]. However, for our application of fingering decision, it is difficult to prepare enough training data, that is, machine-readable guitar scores attached with tablatures. For this reason, we design those parameters as explained in the following instead of estimation from training data.

The difficulty levels of moves are implemented in the transition probabilities between hidden states; a small value of the transition probability means the corresponding move is difficult and a large value easy. As for the movement of the left hand along the neck, the transition probability should be monotone decreasing with respect to the movement distance with the transition. Furthermore, the distribution of the movement distance is sparse and concentrates on the center because the left hand of a guitarist usually stays at a fixed position for several notes and then leaps a few frets to a new position. To approximate such a sparse distribution concentrated on the center, we employ the Laplace distribution (Figure 1),

$$f(x) = \frac{1}{2\phi} \exp\left(-\frac{|x-\mu|}{\phi}\right). \quad (3)$$

It is known that a one dimensional Markov process with increments according to the Laplace distribution is approximated by a piecewise constant function [3] that is similar to the movement of the left hand along the neck. The mean and the variance of the Laplace distribution (3) are μ and $2\phi^2$ respectively. We set μ to zero and ϕ to the time interval between the onsets of the two notes at both ends of the transition so that a long interval makes the transition probability larger, which reflects that a long interval makes the move easier. For simplicity, we assume that the four fingers of the left hand (the index, middle, ring and pinky fingers) are always put on consecutive frets. This lets us calculate the *index finger position* (the fret number the index finger is put on) of form q_i as follows,

$$ifp(q_i) = f_i - h_i + 1.$$

[5] Note that $\min(\delta_{it}, a_{ij})$ does not compare the probability of some subsequence and some transition probability but it does two transition probabilities here.

[6] The actual number of forms is less than this because the 19th fret is most often split by the sound hole and not usable for third and fourth strings, the players hardly place their index fingers on the 19th fret or pinky fingers on the first fret, and so on.

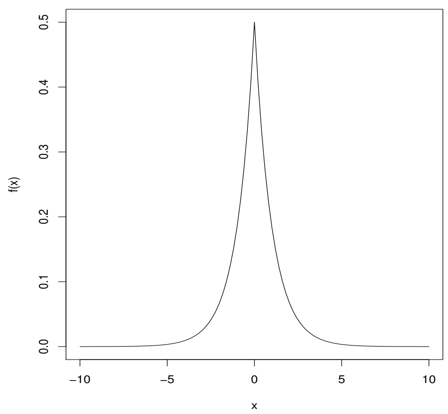

Figure 1. The probability density function of the Laplace distribution for $\mu = \phi = 1$, which is sparse and concentrates on the center.

Using the index finger position, we set the transition probability as

$$a_{ij}(d_t) = P(X_t = q_j | X_{t-1} = q_i, d_t) \sim \frac{1}{2d_t} \exp(-\frac{|ifp(q_i) - ifp(q_j)|}{d_t}) \times \frac{1}{1 + |s_i - s_j|} \times p_H(h_j) \quad (4)$$

where d_t in the first term is set to the time interval between the onsets of the $(t-1)$-th note and the t-th note. The second term corresponds to the difficulty of changing between strings where we employ a function $1/(1 + |x|)$ which is less sparse than the Laplace distribution (3). The third term $p_H(h_j)$ corresponds to the difficulty level of the destination form defined by the finger number h_j. In the simulation in the following section, we set $p_H(1) = 0.35$, $p_H(2) = 0.3$, $p_H(3) = 0.25$ and $p_H(4) = 0.1$ which means the form using the index finger is easiest and the pinky finger the most difficult. The difficulty levels of the forms are expressed in the transition probabilities (not in the output probabilities) in such a way that the transition probability is small when the destination form of the transition is difficult.

As for the output probability, because all the hidden states have unique output symbols in our HMM for fingering decision, it is 1 if the given output symbol o_k is the one that the hidden state q_i outputs and 0 if o_k is not,

$$b_{ik} = P(Y_t = o_k | X_t = q_i) \sim \begin{cases} 1 & (\text{if } o_k = note(q_i)) \\ 0 & (\text{if } o_k \neq note(q_i)) \end{cases} .$$

4. EVALUATION

To evaluate our proposed method, we compared the results of fingering decision using the conventional Viterbi algorithm and the minimax Viterbi algorithm. Figures2-4 show the results for three example monophonic phrases. In each figure, the top and the middle tablatures show the two fingerings obtained by the conventional Viterbi algorithm and the minimax Viterbi algorithm. The numbers on the tablatures show the fret numbers and the numbers in parenthesis below the tablatures show the finger numbers where 1,2,3 and 4 are the index, middle, ring and pinky fingers. The bottom line chart shows the time evolution of the transition probability of the conventional Viterbi algorithm (gray line) and the minimax Viterbi algorithm (black line). The two tablatures and the line chart share a common horizontal time axis, that is, a point on the line chart between two notes in the tablature indicates the transition probability between the two notes.

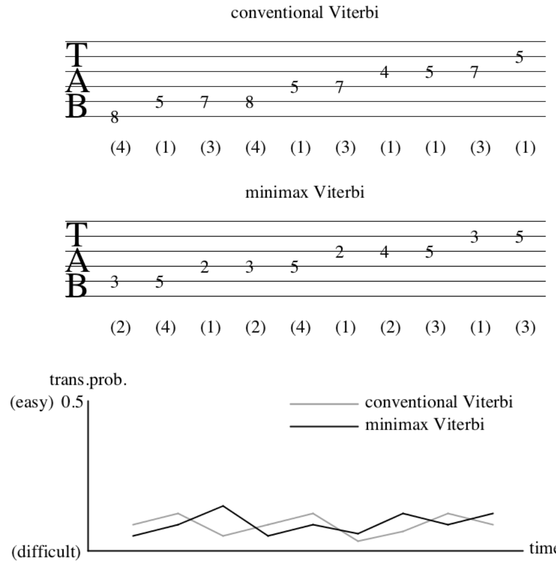

Figure 2. The results of fingering decision for the C major scale starting from C3. Comparing the two tablatures, the one obtained by the minimax Viterbi algorithm is more natural and one that actual guitarists would choose. As for the minimum transition probability, the line chart shows that the minimax Viterbi algorithm gives a larger one.

Figure 2 shows the results for the C major scale starting from C3. From the line chart of the transition probability, we see that the minimum value of the gray line (the conventional Viterbi algorithm) at the sixth transition is smaller than any value of the black line (the minimax Viterbi algorithm), that is, the minimax Viterbi algorithm gives a larger minimum transition probability. As for the tablatures, the one obtained by the minimax Viterbi algorithm is more natural and one that actual guitarists would choose.

Figure 3 shows the results for the opening part of "Romance Anonimo." From the line chart of the transition probability, we see that the gray line (the conventional Viterbi algorithm) keeps higher values at the cost of two very small values while the black line (the minimax Viterbi algorithm) avoids such very small values although it keeps relatively lower values. From the line charts of Figures

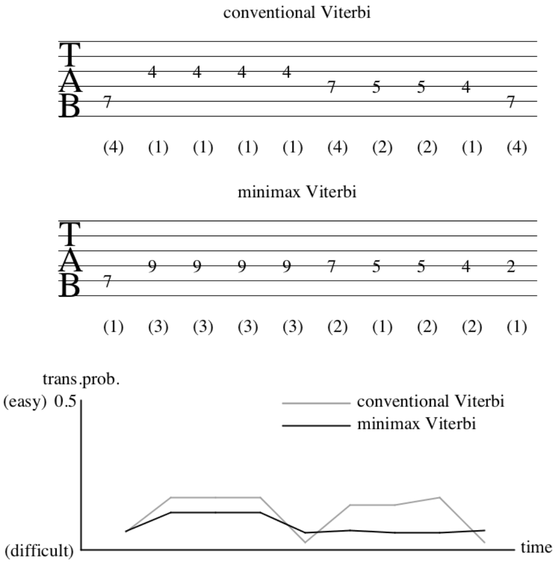

Figure 3. The results of fingering decision for the opening part of "Romance Anonimo" (only top notes). Comparing the two tablatures, the one obtained by the minimax Viterbi algorithm avoids using the pinky finger and suppresses changing between strings. We see from the line chart that the conventional Viterbi algorithm keeps higher values at the cost of two very small values while the minimax Viterbi algorithm avoids very small values although it keeps relatively lower values.

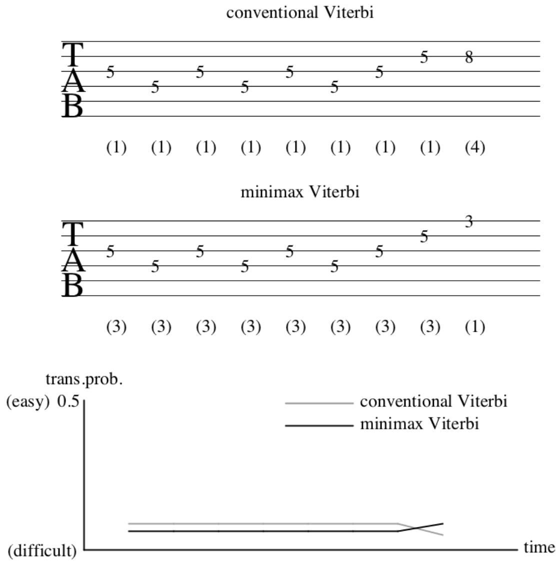

Figure 4. The results of fingering decision for the opening part of "Eine Kleine Nachtmusik" (only top notes). Comparing the two tablatures, the one obtained by the minimax Viterbi algorithm uses the ring finger (instead of the index finger) for the first eight notes to avoid using the pinky finger for the ninth note. The slight difference in the transition probability for the first eight notes comes from the difference in the difficulty of using the index finger and the pinky finger.

2 and 3, we see that the minimax Viterbi algorithm actually minimizes the maximum difficulty for playing a given phrase and makes the most difficult move easier, which can not be done by the conventional Viterbi algorithm. As for the resulting tablatures, while the one obtained by the conventional Viterbi algorithm uses the pinky finger twice and changes between strings three times, the one obtained by the minimax Viterbi algorithm does not use the pinky finger and changes between strings only once.

Figure 4 shows the results for the opening part of "Eine Kleine Nachtmusik." In both fingerings, the first eight notes are played with a single finger that presses down multiple strings across a single fret. The top tablature obtained by the conventional Viterbi algorithm uses the index finger for the first eight notes and the pinky finger for the ninth note while the middle one obtained by the minimax Viterbi algorithm prefers the ring finger for the first eight notes to avoid using the pinky finger for the ninth note. The slight difference in the transition probability for the first eight notes comes from the difference in the difficulty of the form $p_H(h_j)$ in (4) defined by the finger number h_j.

5. CONCLUSION

We have introduced a variant of the Viterbi algorithm termed the minimax Viterbi algorithm that finds the sequence of the hidden states that maximizes the minimum transition probability, and demonstrated the performance of the algorithm with guitar fingering decision based on a synthetic HMM. Fingering decision using our proposed variant has turned out to be able to minimize the maximum difficulty of the move required for playing a given phrase. We have compared the resulting fingerings by the conventional Viterbi algorithm and the minimax Viterbi algorithm to see that our proposed variant is capable of making the most difficult move easier that can not be done by the conventional one. Those observations give rise to interests in the interpolation between the conventional Viterbi algorithm and the minimax Viterbi algorithm. We consider that such an interpolation can be implemented using the L^p-norm of a real vector, which is the absolute sum of the vector elements for $p=1$ and the maximum absolute value for $p=\infty$, and is one of our future study plans. We hope that the present work draws the researcher's attention to the new "minimax paradigm" in automatic fingering decision.

6. ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number 26240025.

7. REFERENCES

[1] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

[2] Yoshua Bengio and Paolo Frasconi. An input output HMM architecture. *Advances in neural information processing systems*, 7:427–434, 1995.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[4] Gen Hori, Hirokazu Kameoka, and Shigeki Sagayama. Input-output HMM applied to automatic arrangement for guitars. *Journal of Information Processing*, 21(2):264–271, 2013.

[5] Wakana Nagata, Shinji Sako, and Tadashi Kitamura. Violin fingering estimation according to skill level based on hidden Markov model. In *Proceedings of International Computer Music Conference and Sound and Music Computing Conference (ICMC/SMC2014)*, pages 1233–1238, Athens, Greece, 2014.

[6] Eita Nakamura, Nobutaka Ono, and Shigeki Sagayama. Merged-output HMM for piano fingering of both hands. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR2014)*, pages 531–536, Taipei, Taiwan, 2014.

[7] Lawrence R Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[8] Daniele P Radicioni, Luca Anselma, and Vincenzo Lombardo. A segmentation-based prototype to compute string instruments fingering. In *Proceedings of Conference on Interdisciplinary Musicology (CIM04)*, volume 17, pages 97–104, Graz, Austria, 2004.

[9] Aleksander Radisavljevic and Peter Driessen. Path difference learning for guitar fingering problem. In *Proceedings of International Computer Music Conference*, volume 28, Miami, USA, 2004.

[10] Samir I Sayegh. Fingering for string instruments with the optimum path paradigm. *Computer Music Journal*, 13(3):76–84, 1989.

[11] Daniel R Tuohy and Walter D Potter. A genetic algorithm for the automatic generation of playable guitar tablature. In *Proceedings of International Computer Music Conference*, pages 499–502, 2005.

[12] Andrew J Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.