

Violin Fingering Estimation According to the Performer's Skill Level Based on Conditional Random Field

Shinji Sako^(✉), Wakana Nagata, and Tadashi Kitamura

Nagoya Institute of Technology, Gokiso-cho, Nagoya, Showa-ku 466-8555, Japan
{s.sako,kitamura}@nitech.ac.jp
nagata@mmsp.nitech.ac.jp

Abstract. In this paper, we propose a method that estimates appropriate violin fingering according to the performer's skill level based on a conditional random field (CRF). A violin is an instrument that can produce the same pitch for different fingering patterns, and these patterns depend on skill level. We previously proposed a statistical method for violin fingering estimation, but that method required a certain amount of training data in the form of fingering annotation corresponding to each note in the music score. This was a major issue of our previous method, because it takes time and effort to produce the annotations. To solve this problem, we proposed a method to automatically generate training data for a fingering model using existing violin textbooks. Our experimental results confirmed the effectiveness of the proposed method.


1 Introduction

With a violin, the same pitch can be produced by several fingering patterns, and players decide which fingering to use. In general, the optimum fingering differs according to a player's skill level. For low-skill players, fingering that is easily played is optimum, whereas for high-skill players, fingering that allows the best performance expression is optimum. From this point of view, it is important to consider the skill level of the player in the field of automatic fingering estimation techniques.

Some studies have focused on fingering estimation for a plucked or bowed string instrument [1–5] or for the piano [6–8]. The methods proposed in these studies estimate the easiest fingering and can not recommend suitable fingering for various skill levels; however, one promising approach is a method to describe the relationship between fingering and music score by using a stochastic model such as [5, 7]. We have been working on the research of violin fingering estimation based on such a stochastic model approach. Our goal is to estimate the natural violin fingering according to the player's skill level for any musical compositions. It is considered to be a human-centered design in assistive technology for musical instrument training. In this paper, we propose the technique for violin fingering estimation according to the performer's skill level based on a conditional random field (CRF).

Fingering for beginners

Finger No.	2	1	0	3	2	1	0	2	1	0
String No.	E	E	E	A	A	A	E	A	A	A



Finger No.	4	3	2	1	2	1	4	2	1	0
String No.	A	A	A	A	A	A	A	A	A	A

Fingering for Intermediates

Fig. 1. Examples of violin fingering for different skill level

2 Methodology

2.1 Violin Fingering and Our Previous Study

Figure 1 shows examples of violin fingering for beginners and intermediates players. The violin player decides whether playing needs to be easy or whether performance expression is appropriate. We also realize that this priority is influenced by the note length. If the note is short, ease of play becomes a higher priority because playing a succession of short notes is more difficult. When the note is long, expression has a higher priority because playing longer notes is easier. Expression also has a higher priority when the skill level is high.

From this point of view, we previously proposed a fingering estimation method based on a hidden Markov model (HMM) [9]. In our previous study, we regarded fingering as the hidden state and the notes in the musical composition as the observation. We defined the priority of performance expression based on note length and skill level, and this priority was used to determine the output probability. Because note length also influences ease of transition from one fingering pattern to another, we defined the degree of change between fingering patterns based on note length, and this degree of change was related to transition probability.

Model parameters were estimated from textbook fingering patterns; however, that method requires fully annotated fingering for training HMM, making it difficult to prepare the training data. In general, partial fingering patterns are described in violin textbooks. We had to create a large amount of complementary fingering data manually in order to train fingering estimation models, which required a lot of time and the skill and knowledge of violin playing.

2.2 Outline of Our Method

Figure 2 shows an overview of our complementary training method using partial fingering data. The initial model is trained by using the completed fingering

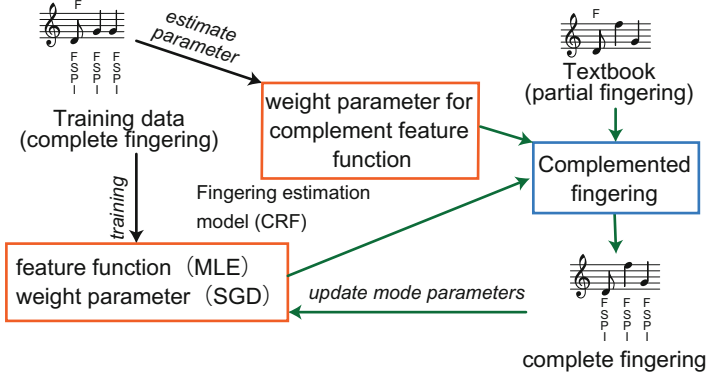


Fig. 2. Overview of our complementary training method using CRF

data that has been fully complemented manually in the same way as in our previous study. Parameters of complementary feature function are estimated from complete fingering data. Then, partial fingering data is complemented by using the CRF, and the model is updated using the complemented fingering data. In this study, we adopted well-known stochastic gradient descent (SGD) method to train CRF. This cycle is repeated under the convergence condition.

2.3 Basic Idea of Fingering Model Using CRF

In this paper, we propose a framework for training the fingering estimation model using a partial fingering data from violin textbooks. To accomplish that, we model violin fingering using the concept underlying the CRF as shown in Fig. 3. This model is an extended version of our HMM-based fingering model. State sequences \mathbf{s} is the left-hand state sequence, and output \mathbf{o} is the note and rest sequence in the score. We assume that the state changes for every note and that the state sequence is a Markovian process. To simplify the problem, the model has the following restrictions: the score is monophonic, and only the factors pitch, note length, and rest length are considered by this model.

Each note information of \mathbf{o} can be represented as set of pitch information p , expressiveness e and changeableness c . Expressiveness e depends on the parameter w^l representing the skill level.

The four elements are represented by the following variables: finger number FN, string SP, hand position HP and finger interval FI.

$$\mathbf{s}_n = \{x_n^{\text{FN}}, x_n^{\text{SP}}, x_n^{\text{HP}}, x_n^{\text{FI}}\} \quad (1)$$

The objective function for the fingering estimation is defined as Eq. (2) by using potential function $\Phi(\mathbf{o}, \mathbf{s})$.

$$\hat{\mathbf{s}} = \arg \max_{\mathbf{s}} P(\mathbf{s}|\mathbf{o}) = \arg \max_{\mathbf{s}} \frac{\exp(\Phi(\mathbf{o}, \mathbf{s}))}{z(\mathbf{o})} \quad (2)$$

Here, $z(\mathbf{o})$ is a normalization term.

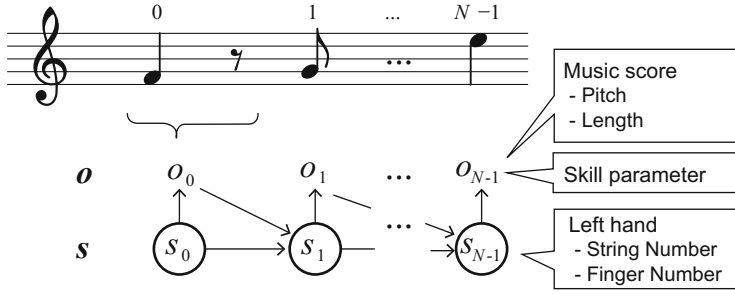


Fig. 3. Outline of our fingering model

2.4 Potential Representation

CRF potential function Φ is represented as a linear combination of feature functions as follows:

$$\Phi(\mathbf{o}, \mathbf{s}) = \sum_n \sum_{i,j} w_{i,j} f_{i,j}(o_{n-1}, o_n, s_{n-1}, s_n) \quad (3)$$

In this study, feature functions of CRF are represented by a probability density function because the degree of expressiveness e and changeableness c are continuous values. Both e and c have been introduced to accommodate skill levels in our previous study (please refer to [9] for more details). There are four features: state feature, transition feature, expression feature, and pitch feature. Due to the number of states being too large, each feature is defined individually for each element except for the pitch feature.

2.5 State Feature

This feature represents the appropriateness of the hands of the state. To simplify the model, we assumed that feature is independent of each element. The state feature function is defined as Eq. (4) as the logarithm of the probability of occurrence of each element.

$$f_{1,j}(s_n) = \log P(x_n^j) \quad (4)$$

It can be considered as $j \in \{\text{FN}, \text{HP}, \text{FI}\}$, because this feature does not depend on the violin string.

2.6 Transition Feature

This feature represents the appropriateness of the transition of the state, and its feature function is defined as the logarithm of the Laplace distribution or the exponential distribution, where the variance of distribution depends on the degree of changeableness c . It is noted that this feature also depends on the

finger numbers before the transition. The details of transition the probability are described in [9], and a simplified example is represented by Eq. (5).

$$f_{2,j}(o_{n-1}, s_{n-1}, s_n) = \log f_{\text{Lap}}(x_n^j; x_{n-1}^j, k_j c_n) \quad (5)$$

Here, $f_{\text{Lap}}(x; \mu, \sigma^2)$ is the Laplace distribution with mean μ and variance σ^2 . It can be considered as $j \in \{\text{SP}, \text{HP}, \text{FI}\}$, because the transition of finger numbers does not depend on the appropriateness of fingering.

2.7 Expressiveness Feature

This feature represents the appropriateness of the expression. The frequency of expression can be approximated by log-normal distribution. The feature function is represented as Eq. (6). It can be considered as $j \in \{\text{FN}, \text{SP}\}$, because expressiveness depends on both finger number and string.

$$f_{3,j}(o_n, s_n) = \log f_{\text{LN}}(e_n; \mu_{j,x_n^j}, \sigma_{j,x_n^j}^2) \quad (6)$$

2.8 Pitch Feature

The relationship between state and pitch is represented as Eq. (7). The state corresponds to the pitch to set the probability to zero, otherwise set the probability to ∞ .

$$f_4(s_n, o_n) = \begin{cases} 0 & \text{state } s_n \text{ correspond to pitch } p_n \\ -\infty & \text{state } s_n \text{ does not correspond to pitch } p_n \end{cases} \quad (7)$$

3 Automatic Fingering Completion

In general, fingering is not described for every note in commercial fingering textbooks. One reason is that an easily guessed part is often omitted. In addition, fingering information is represented by finger number only, with string information provided only as required. To use such partial fingering data as training data, it is necessary to create a full fingering data by completion.

3.1 Outline of Completion Method

In this study, we introduce a new method for a semi-automatic completion method by using a fingering estimation framework based on CRF. At first, we focus on the difference between described and non-described fingering in the textbook. As for non-described parts, only a small change in the state of the hand would be expected; on the other hand, a large change in the state of the hand is expected at locations where the fingering is described.

These relationships can be represented by changing the weights of the feature by the described or non-described fingering in the textbook. Finally, a potential function for the automatic complement is represented as Eq. (8).

$$\Phi_{\text{cmp}}(\mathbf{o}, \mathbf{s}) = \sum_n \sum_{i,j} \alpha_{i,j}(n) w_{i,j} f_{i,j}(o_{n-1}, o_n, s_{n-1}, s_n) \quad (8)$$

Here, $\alpha_{i,j}(n)$ is the function that changes the weight by the described or non-described fingering:

$$\alpha_{i,j}(n) = \begin{cases} \alpha_{i,j}^0 & \text{fingering is not described at } n^{\text{th}} \text{ note} \\ \alpha_{i,j}^1 & \text{fingering is described at } n^{\text{th}} \text{ note} \end{cases} \quad (9)$$

Here, $\alpha_{i,j}^0$ means fingering was not described at the n^{th} note, while $\alpha_{i,j}^1$ means fingering was described at the n^{th} note.

An optimal complemented fingering can be searched from the state sequence through the textbook fingering by using the potential function. Complement state sequence can be represented as Eq. (10) for the set of state sequences \mathbf{S}_{text} that satisfy the textbook fingering.

$$\hat{\mathbf{s}}_{\text{cmp}} = \arg \max_{\mathbf{s} \in \mathbf{S}_{\text{text}}} \frac{\exp(\Phi_{\text{cmp}}(\mathbf{o}, \mathbf{s}))}{z(\mathbf{o})} \quad (10)$$

4 Training Method Using Textbooks

Figure 4 shows an outline of our training method. Our goal is to train the fingering estimation model from textbooks. In order to obtain automatic complemented fingering data from textbooks, an initial fingering estimation model is required. In this study, we use an initial data set with a small amount of manually complemented fingering data. Estimation of auto-complemented fingering data and updating of model parameters is repeated. The termination condition for this process is the case where a concordance rate of $(t_1 - 1)^{\text{th}}$ and t_1^{th} complemented fingering is equal to or more than $\alpha\%$.

4.1 Parameter Estimation for θ

The parameter set of the feature function consists of an occurrence probability of each element of 57 dimensions in total. These parameters are estimated in the same manner as [9].

4.2 Parameter Estimation for ω

It is not necessary to consider the weight, since the pitch feature can not be defined only as zero or one. We have to estimate an 8-dimensional feature weight in total.

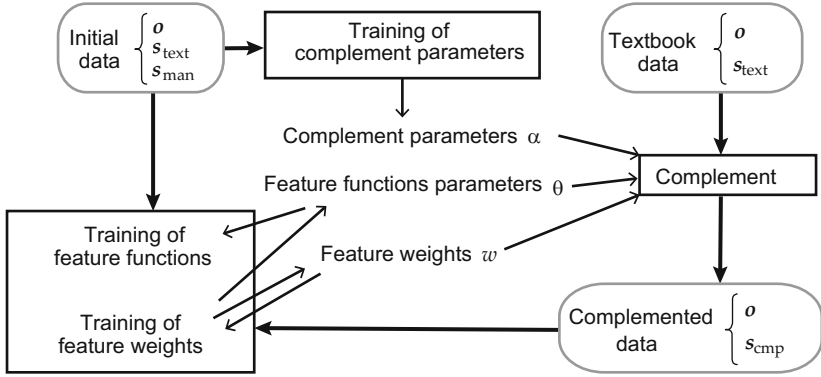


Fig. 4. Outline of the training method

In general, the weight parameter of the CRF is estimated by using the limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method in cases of batch training. On the other hand, a stochastic gradient descent (SDG) method is used in cases of online training. In the batch training, parameters are updated by using all training data. An advantage of the batch method is the stability of convergence, but the computational time is a problem. In the online method, training is performed using the training data sequentially. An advantage of the online method is that the calculation time is small, but the problem is instability of convergence. In this study, one data set corresponds to the whole music score, which contains several hundreds notes.

The gradient vector of SGD is defined as Eq. (11), and the weight update equation is defined as Eq. (12).

$$\mathbf{g}(\mathbf{o}^{(d)}, \mathbf{s}^{(d)}) = \mathbf{f}(\mathbf{o}^{(d)}, \mathbf{s}^{(d)}) - \sum_{\mathbf{s}} \mathbf{f}(\mathbf{o}^{(d)}, \mathbf{s}) P(\mathbf{s} | \mathbf{o}^{(d)}) \quad (11)$$

$$\mathbf{w}^{t_2+1} \leftarrow \mathbf{w}^{t_2} + \eta_{t_2} \mathbf{g}(\mathbf{o}^{(d_{t_2})}, \mathbf{s}^{(d_{t_2})}) \quad (12)$$

Here, t_2 , d and η_{t_2} represent the number of updates of the weight, index of training data and training coefficients, respectively. In the online training, the update process is performed in units of one music score. d is defined as the remainder obtained by dividing the t_2 and D . $P(\mathbf{s} | \mathbf{o}^{(d)})$ can be calculated efficiently using a forward-backward algorithm.

In the SGD method, the training result is often strongly affected by the data that is used in the most recent update cycle. For this reason, training coefficient η_{t_2} is defined to be reduced by the weight parameter update cycle t_2 . Determination condition of convergence L_2 -norm, which is a change in ratio of weight, is used. A summary of these ideas can be written as Eq. (13).

$$\Delta_{t_2} = \sqrt{\sum_{i,j} (n(w_{i,j}^{t_2}) - n(w_{i,j}^{t_2-1}))^2} \quad (13)$$

Here, $n(\cdot)$ is a function to normalize such that the sum of the weight is equal to one. Δ_{t_2} is divided by the number of occurrences $N^{(d)}$ because it is strongly affected by the high frequency of occurrence data. Because it is considered to be dependent on the data (music), we use the average of the last D times. It is considered to have converged if the average is less than the threshold value b . The update performed at least D times.

$$\sum_{i=t_2-D+1}^{t_2} \frac{\Delta_i}{N^{(d_i)}} < b \quad (14)$$

4.3 Complement Parameter α

Dimensionality of the complementary parameters is twice the number of dimensions of feature weights. Our complementary parameters are represented by the ratio of the weight of fingering the described part and the non-described part.

Complementary parameters are defined as Eq. (15). Here, w_{all} , w_0 , and w_1 are weight parameters corresponding to training from all notes, fingering described notes and fingering non-described notes, respectively.

$$\alpha_{i,j}^k = n(w_{i,j}^k) / n(w_{i,j}^{\text{all}}) \quad k = 0, 1 \quad (15)$$

5 Experiment and Results

5.1 Settings

We performed a fingering estimation experiment to confirm the differences between our complementary training method using partial data and our previous training method using complete data. We used sixteen musical pieces (total 4,594 notes) from some textbooks for intermediate violin students. We also used one musical pieces (total 101 notes) for the initial model. Sixteen musical pieces were used to generate the complemented fingering data that was used for training the CRF model. The intermediate test data set comprised fourteen musical pieces (total 2,265 notes) that did not overlap with the training data.

By the preliminary experiments, some parameters of the training were set as follows: $a = 99.5\%$, $b = 3.0 \times 10^{-5}$, $\eta_{t_2} = 10^{-3} \times 0.7e^{t_2/D}$, $w^l = 1.0$.

5.2 Results

Figure 5 shows the average concordance rate obtained from the experiment. The horizontal axis represents the amount of data used for training the initial model. The proposed method obtained equivalent performance to the previous method that was trained using the complete data set. In addition, the proposed method was superior to the previous method even when the training data of the initial model was small. The results show that it is possible to greatly reduce the manually complemented fingering data. In particular, cost of preparing training data has been reduced to about 1/45 by using proposed method.

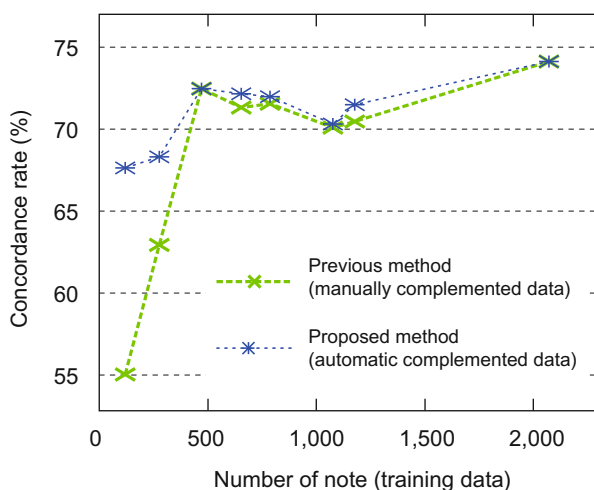


Fig. 5. Result of fingering estimation experiment using proposed training method and previous training method

6 Conclusion

In this paper, we proposed a CRF-based violin fingering estimation model and complementary training method by extending our HMM-based violin fingering estimation method according to skill level. Our complementary training method, using partial fingering data, showed the same performance as the previous training method using the complete fingering data. In other word, high-precision fingering estimation model can be obtained from small amount of manually complemented data. Our study also makes simplifies and reduce the time cost in the training data creation task.

There are still some issues, however, about the naturalness of the estimated fingering, especially in the performance expression, which depends on slur, volume, and other factors. In our future work, we will consider such another information obtained from the music score other than note information.

Acknowledgement. This research was supported in part by Japan Society for the Promotion of Science (JSPS) KAKENHI (Grant-in-Aid for Scientific Research) Grant Number 26730182, and The Telecommunications Advancement Foundation.

References

1. Radisavljevic, A., Driessen, P.: Path difference learning for guitar fingering problem. In: Proceedings of the International Computer Music Conference, pp. 456–461 (2004)
2. Miura, M., Hirota, I., Hama, N., Yanagida, M.: Constructing a system for finger-position determination and tablature generation for playing melodies on guitars. *Syst. Comput. Jpn* **35**(6), 10–19 (2004)

3. Tuohy, D.R., Potter, W.D.: A genetic algorithm for the automatic generation of playable guitar tablature. In: Proc. the International Computer Music Conference, pp. 499–502 (2005)
4. Radicioni, D., Scienza, C.D., Lombardo, V.: Guitar fingering for music performance. In: Proc. the International Computer Music Conference. (2005) 527–530
5. Hori, G., Kameoka, H., Sagayama, S.: Input-output hmm applied to automatic arrangement for guitars. *Journal of information processing* **21**(2), 264–271 (2013)
6. Hart, M., Bosch, R., Tsia, E.: Finding optimal piano fingerings. *The UMAP Journal* **21**(2), 167–177 (2000)
7. Yonebayashi, Y., Kameoka, H., Sagayama, S.: Automatic decision of piano fingering based on hidden markov models. In: Proc. the 20th International Joint Conference on Artificial Intelligence. (2007) 2915–2921
8. Kasimi, A.A., Nichols, E., Raphael, C.: A simple algorithm for automatic generation of polyphonic piano fingerings. In: Proc. the 8th International Conference on Music Information Retrieval. (2007) 355–356
9. Nagata, W., Sako, S., Kitamura, T.: Violin fingering estimation according to skill level based on hidden markov model. In: International Computer Music Conference (ICMC) and Sound and Music Computing conference (SMC) 2014. (2014) 1233–1238