Chord Function Identification with Modulation Detection Based on HMM

Yui Uehara[1], Eita Nakamura[2], and Satoshi Tojo[1] *

[1] Japan Advanced Institute of Science and Technology
[2] Kyoto University
{yuehara, tojo}@jaist.ac.jp, enakamura@sap.ist.i.kyoto-u.ac.jp

Abstract. This study aims at identifying the chord functions by statistical machine learning. Those functions found in the traditional harmony theory are not versatile for the various music styles, and we envisage that the statistical method would more faithfully reflect the music style we have targeted. In machine learning, we adopt hidden Markov models (HMMs); we evaluate the performance by perplexity and optimize the parameterization of HMM for each given number of hidden states. Thereafter, we apply the acquired parameters to the detection of modulation. We evaluate the plausibility of the partitioning by modulation by the likelihood value and, as our innovative method, the result is reduced back to the number of states conversely. As a result, we found that the six-state model outperformed the other models both for the major keys and for the minor keys although they assigned different functional roles to the two tonalities.

Keywords: chord function; hidden Markov model; modulation detection.

1 Introduction

The chord functions are one of the most fundamental bases of tonal music to identify the key. Although the traditional functional harmony theory well describes the general roles of chords, the functions should have been diversified in accordance with the target music.

Previously chord function identification has been carried out mainly by statistical clustering algorithms [4, 8]. Since these statistical methods learn from raw data instead of the textbook theory, they have the potential to reflect the difference of music styles. A recent study proposed a generative model [12], which is advantageous in its predictive power and in its applicability to practical problems such as melody harmonization [11]. However, this study focused on popular music and the key was assumed to be invariant within each piece. In our research, we consult J. S. Bach's music, thus the modulation detection would be inevitable. Thus far, modulation detection has been carried out either

* This research has been supported by JSPS KAHENHI Nos. 16H01744 and 19K20340.

by heuristics [8] or by a key-finding algorithm [4] though there have been still several difficult cases to determine the key [10].

We conceive that the local keys could be also determined by the functional progression of chords. Therefore, we propose a new dynamic modulation detection method, applying the statistically found chord functions. Here, the optimal number of functions would be also determined computationally so that it maximizes the likelihood of chord progressions in the entire corpus. In this research, we achieve the detection of the data-oriented chord functions, together with the detection of modulation. We envisage that we would obtain finer-grained chord functions which faithfully reflect the targeted music style. Our method is new in that we do not need to prefix the scope of modulation as opposed to the algorithm using the histogram of pitch classes [5, 10, 3, 13].

We begin this paper by reviewing related works, especially the key detection algorithms and the statistical learning methods of the chord functions in section 2. Then, we propose our method in section 3, and thereafter show the experimental results in section 4. We conclude in section 5.

2 Related Work

2.1 Key detection algorithms

Among the key detection algorithms based on the histogram of the pitch classes [10, 5, 3, 13], the most widely used one is the Krumhansl-Schmuckler algorithm that adopts the key-profile obtained by a psychological experiment [5]. More recently, the key-profile was obtained from music data by using a simple Bayesian probabilistic method [10] and the Latent Dirichlet Allocation (LDA) [3].

Sakamoto et al. [9] employed the distance between chords by using Tonal Pitch Space (TPS) [6] rather than the pitch classes. Given a sequence of Berklee chord names, the key is detected by the Viterbi algorithm, not requiring a fixed scope. A Berklee chord can be interpreted in multiple keys, for example, the chord **C** is **I** of C major key as well as **IV** of G major key. Therefore, the network of candidate nodes consists of keys with degree names. Since TPS does not have adjustable parameters, it cannot reflect the difference in music styles.

2.2 Statistical learning of the chord functions

Statistical learning of the chord functions has been studied by classifying the chords using clustering algorithms. Rohrmeier and Cross [8] used the hierarchical cluster analysis to find the statistical properties of the chords, where the most distinctive cluster of the pitch class sets reflected the dominant motion in both major and minor keys. They also found that the result for the minor key was significantly different from that for the major key. The clusters that represent the Tonic and Dominant of the relative major key were obtained.

Jacoby et al. [4] also carried out the clustering of the chords in J. S. Bach's chorales and some other datasets. They proposed the evaluation method using two criteria, accuracy and complexity, inspired by the information theory. They introduced the optimal complexity-accuracy curve, which is formed by the maximal accuracy for each complexity. When using diatonic scale degrees as the surface tokens, the functional harmony theory that uses Tonic, Dominant, Sub-dominant clustering was plotted on the optimal curve, while the Major, Minor, Diminished clustering was far less accurate. This means that the functional harmony theory is more favorable than Major, Minor, Diminished clustering when using the diatonic scale degrees as the surface tokens. In addition, they employed the analysis with automatically labelled data. They adopted the key-detection algorithm of White and Quinn [13] that used the Krumhansl-Shmuckler algorithm [5] on windows of eight slices, and picked up the most common 22 pitch classes (with the bass notes) as the surface tokens. They reported that the obtained clusters were quite close to the Tonic, Dominant, Sub-dominant classification when the number of the categories was 3.

On the other hand, Tsushima et al. [12] found the chord functions in datasets of popular music pieces, using generative models rather than clustering: HMM and Probabilistic Context Free Grammar (PCFG). They reported that when the number of states was 4, the output probability of HMM trained with a popular music dataset could be interpreted as the chord functions: Tonic, Dominant, Sub-dominant, and Others [12], though the model achieved less perplexity with more states. Although PCFG is more advantageous since it can represent more external structures such as long-range dependency of cadence, the reported performance did not exceed that of the HMM. Using a trained HMM as the initial value of PCFG was also found to be clearly effective. However, for the melody harmonization task, PCFG was reported more effective than HMM [11]. For training the HMM, they tested the expectation-maximization (EM) algorithm and Gibbs Sampling (GS) since GS showed significantly higher accuracy than the EM algorithm in the part-of-speech tagging task [2]. They reported that the GS algorithm may perform better especially for a large number of hidden states since it can avoid being trapped in bad local optima.

3 Chord function identification with HMM

Following the previous works, we used a statistical approach to identify chord functions. We chose the HMM for our model because its structure agrees well with that of the functional harmony theory. We expect that the states of the HMM represent chord functions, instead of another possible approach that assumes chord symbols as the hidden states and surface notes as the output tokens.

We obtained the chord functions with the plausible number of states that was fed back by the modulation detection in the following steps.

1. Train the HMM in the range of 2–12 states and choose the best parameterization for each number of states in terms of perplexity.
2. Calculate the likelihood of the chord progression of every candidate partition of key blocks by using the obtained HMM, and determine the intervals of

modulation that maximize the sum of the likelihoods of key blocks by using the set partitioning model.[3]
3. Obtain the best number of states that scores the highest sum of likelihoods on an entire corpus.

3.1 Dataset

We used J. S. Bach's four-part choral pieces BWV253-438 from the Music21 Corpus [1] as our dataset. Several pieces in the chorales have complicated modulations which are not compatible with the modern tonalities. We should also consider that the key signatures of several pieces are different from the modern tonal system. We excluded 24 pieces which obviously differed from the major and the minor key: 22 dorian, 1 mixolydian, and 1 phrygian, and targeted the remaining 94 major pieces and 68 minor pieces. However, there were still pieces that retained the feature of the church modes, especially in minor mode pieces.

To learn the chord functions, we used only the first and the last phrases[4] that were identified by the fermata[5] notation in each piece because we supposed to be able to identify the key of these phrases from the key signature. Those pieces whose first and last chords were different from the tonic of the key signature were excluded.

3.2 Functional chord progression model based on HMM

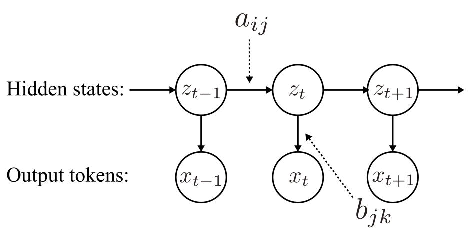

Fig. 1. Graphical representation of the hidden Markov model (HMM).

Model We regarded chord degrees as output tokens for the HMM in Fig. 1, and states as chord functions. Here, z_t denotes the hidden state and x_t the output token at each time step. The state-transition probability is denoted by a_{ij} and the output probability b_{jk}. The number of distinct states is denoted by N_s, and that of output tokens N_v. When we need to specify a state, we use $(z_t =)s_i, i \in \{1, \ldots, N_s\}$, and for output tokens we use $(x_t =)v_k, k \in \{1, \ldots, N_v\}$.

[3] The set partitioning model is a sort of the linear programing.
[4] In this paper, a phrase means a section divided by fermatas.
[5] Fermata is a notation which usually represents a grand pause. However, in the chorale pieces, it represents the end of a lyric paragraph.

Surface tokens We modelled the chord functions of the major key and the minor key by the HMM, and investigated the number of states in the range from 2 to 12. To train the models, we transposed all the major keys to C major and all the minor keys to A minor.

Basically, we used chord degrees on the diatonic-scale as the surface tokens because we trained the models only for C major and A minor, and used them for other keys by transposing the surface tokens. We needed to use more tokens for the minor key considering the all possible chords that were created by introducing the leading-tone in addition to the natural **VII**. The surface tokens of the major and minor keys are listed in Table 1.

Major		Minor	
Chord name	Proportion	Chord name	Proportion
C major(**I**)	30.50%	A minor(**i**)	28.59%
G major(**V**)	19.56%	E major(**V**)	14.94%
D minor(**ii**)	12.35%	C major(**III**)	7.91%
A minor(**vi**)	10.76%	B diminished(**ii**°)	7.22%
F major(**IV**)	9.57%	D minor(**IV**)	6.23%
B diminished(**vii**°)	5.44%	G major(**VII**)	6.13%
E minor(**iii**)	4.37%	G♯ diminished(**vii**°)	5.24%
Others	7.45%	F major(**VI**)	5.14%
		E minor(**v**)	3.46%
		C augmented(**III**$^+$)	2.08%
		Others	13.06%

Table 1. Surface tokens.

Here, we simply removed chords that were not classified to major, minor, diminished, and augmented by using a function to classify the qualities of chords in the Music21 library [1]. We treated the remaining chords that were not in the diatonic scale as 'Others'. In addition, we treated a succession of the same chord as a single surface token.

Optimization method We used the simple EM-based approach known as the Baum-Welch algorithm for learning the HMM parameters from data. While the GS would be effective to avoid bad local optima [12, 2], we rather employed the optimization from a large number of initial values to study the variance of locally optimal parameterizations. For each number of states, we used 1000 different initial values to learn the parameters. We randomly initialized the state-transition probability matrix, while the output probability matrix was initialized uniformly. For each initial value setup, the training data consisting of randomly connected pieces, where we shuffled the opus numbers of the pieces and put them into one sequence.

Evaluation measures We evaluated the parameterizations of the HMM obtained from 1000 different initial values on each number of states (among 2 – 12) to find the optimal one. For each number of states, we selected the optimal parameterization which scored the lowest perplexity defined by following equation:

$$\mathcal{P} = \exp\left(-\frac{1}{|\boldsymbol{x}|}\ln P(\boldsymbol{x}|\boldsymbol{\theta})\right). \tag{1}$$

We also calculated the variance of the 1000 optimal parameterizations for each number of states by employing the K-means clustering around the best optimal parameterization. A large variance indicates larger difficulty to consistently obtain the optimal parameterization.

3.3 Modulation detection as the set partitioning problem

The remaining problem is to select the best number of hidden states. We obtain it by using the modulation detection described below. We select a key that maximizes the likelihood, calculated by the obtained HMM. If we simply apply the HMM, we can only obtain one optimal key for a target piece. By the set partitioning algorithm to detect modulations, we can assign the optimal key blocks to the target piece. The chord functions are expected to work well for detecting a key, especially when there are modulations in the target pieces.

This idea can be formulated as a special case of the set partitioning model, regarding that a music piece is composed of locally optimal key blocks. Here, we use the following notation.

$T = \{1, \cdots, N_t\}$	Serial number of chords in a target sequence
$C = \{1, \cdots, N_c\}$	Set of indices of candidate blocks
$j \in C$	Index of blocks
C_j	Set of chords in candidate block j
e_{ij}	$e_{ij} = 1$ if chord $i \in C_j$ and otherwise $e_{ij} = 0$
d_j $(j \in C)$	$d_j = 1$ if C_j is chosen in the partition and otherwise $d_j = 0$
r_j	Score (the likelihood and penalty) of candidate block C_j

Table 2. Notation in the set partitioning model.

The objective of this set partitioning model is to maximize $\sum_{j=1}^{N_c} r_j d_j$, which means that we select the set of blocks that gives the highest score. The imposed constraints are $\sum_{j=1}^{N_c} e_{ij} x_j = 1, i \in T, d_j \in \{0,1\}, j \in C$, which means that a surface token must be included in one and only one block.

Since we used only the chords on the diatonic scales, there were many tokens that were classified as 'Others' described in Table 1 when considering all the candidate keys. We imposed penalty on 'Others' tokens. The penalty value was empirically set to log(0.01).

4 Experimental results

4.1 Evaluation for each number of hidden states

Perplexity For each number of states, we assumed that a parameterization with a lower perplexity is better. With this criteria, we sorted the results by the perplexity and selected the best one in all the results from 1000 initial values. The best perplexity decreased as the number of states increased (Fig. 2). This result is consistent with the previous work that used a popular music dataset [12].

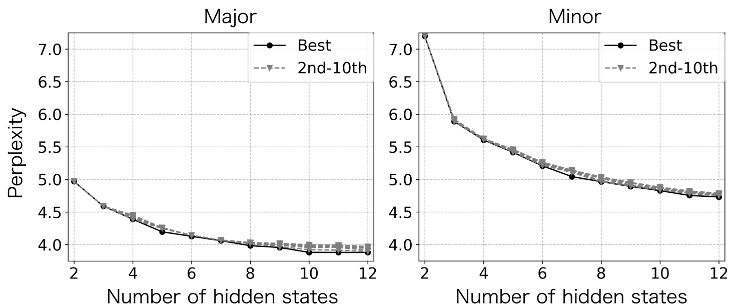

Fig. 2. Perplexities of 10-top parameterizations for each number of states.

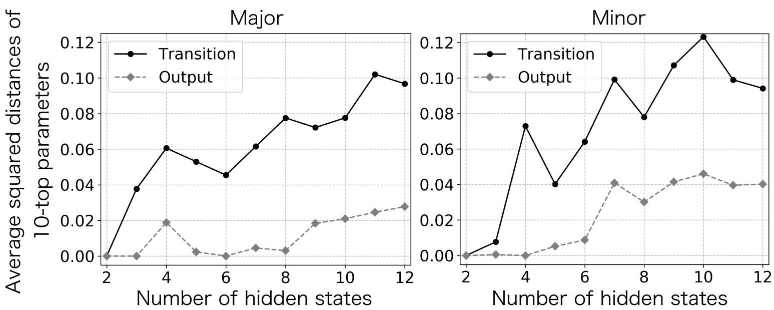

Fig. 3. Average squared distances of 10-top parameters with K-means clustering.

Variance Next, we studied the variance of the optimal parameterizations. For each number of states, we calculated the average squared distances of each of the output and transition probabilities among the top 10 optimal parameterizations. To eliminate the influence of the permutation ambiguity of the state labels, we adopted the K-means clustering method for calculating the squared distance between two parameterizations of output/transition probabilities. More specifically, we used the Scikit-learn library [7] and fixed the centroids of the clusters as the best optimal parameter values.

As shown in Fig. 3, the distances of the optimal parameterizations increase along with the number of hidden states. This suggests that when the number of hidden states is large there are many different optimal parameterizations and it is difficult to uniquely find the best parameterization solely based on the perplexity.

4.2 Selecting the number of hidden states

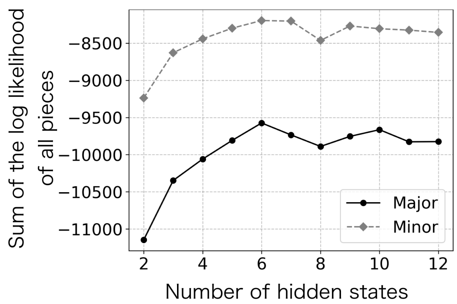

Fig. 4. Sum of the log likelihood of all pieces.

As explained in section 3.3, we obtained the the appropriate number of states by simultaneously employing the chord function identification and modulation detection. To reduce the computation time, we separated a piece into phrases by using the fermata notation, and calculated the likelihood on each phrase.

The 6-state model scored the highest sum of likelihoods both for the major keys and for minor keys (Fig. 4).

4.3 Chord function identification

Major key For the major key, the chords were classified into fine-grained functions, up to 6 states, as shown in Fig. 5. When the number of states is 3, in addition to the clear functions of Tonic $\{\mathbf{I}\}$ and Dominant $\{\mathbf{V}, \mathbf{vii}^\circ\}$, there is a mixed function of Tonic and Sub-dominant to which $\{\mathbf{ii}, \mathbf{iii}, \mathbf{IV}, \mathbf{vi}\}$ are assigned. This mixed function is separated into Tonic $\{\mathbf{iii}, \mathbf{vi}\}$ and Sub-dominant $\{\mathbf{ii}, \mathbf{IV}\}$ when the number of states is 4. And then, the state of Dominant is separated into $\{\mathbf{V}\}$ and $\{\mathbf{vii}^\circ\}$ with 5 states. Finally, when the number of states is 6, most chords are assigned to an almost unique state, except that $\{\mathbf{iii}, \mathbf{vi}\}$ form one state. Here, we see that $\{\mathbf{iii}\}$ is mainly assigned to Tonic, which recovers the result of Tsushima et al. for popular music datasets [12].

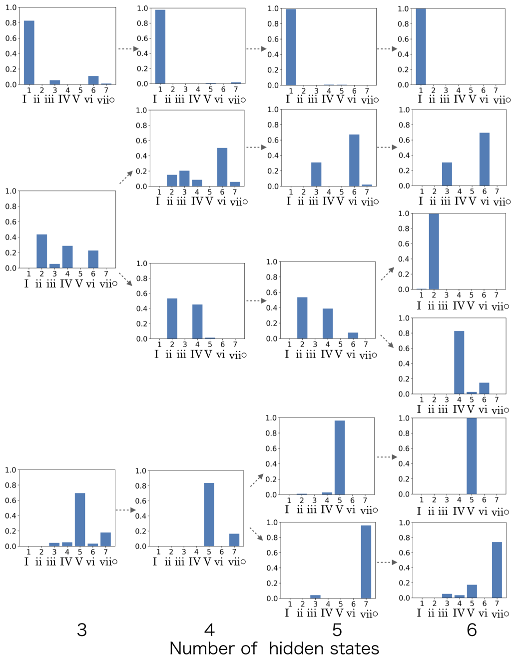

Fig. 5. Output probabilities of the best HMMs for the major key.

The fine-grained state-transition probability is also meaningful. As shown in Fig. 6, we can find detailed functions. For example,

1. The state s_2 for **V** and state s_6 for **vii**$^\circ$ both tend to proceed to state s_4 for **I**, while state s_6 less often proceeds to s_3 for $\{\mathbf{iii}, \mathbf{vi}\}$.
2. Although both states s_1 and s_5 have the function Sub-dominant, s_1 for **ii** more often proceeds to Dominant chords (state s_2 and state s_6) than state s_5 for **IV**.

Minor key The results for the minor key were significantly different from those for the major key, where states corresponding to Tonic and Dominant of the relative major key were obtained when the number of states was larger than 4. With 6 hidden states, in addition to Tonic, Dominant and Sub-dominant, the Tonic of the relative major and that of the Dominant of the relative major were obtained. This result reflects the feature of the choral, whose melodies were composed in the medieval ages in the church modes instead of modern tonalities, prior to the harmonization by J. S. Bach, because the relative keys share the common pitch classes like the church modes.

Rohrmeier et al. also pointed out that the groups of chords corresponding to the relative major key were existing in the minor key clusters [8]. In addition

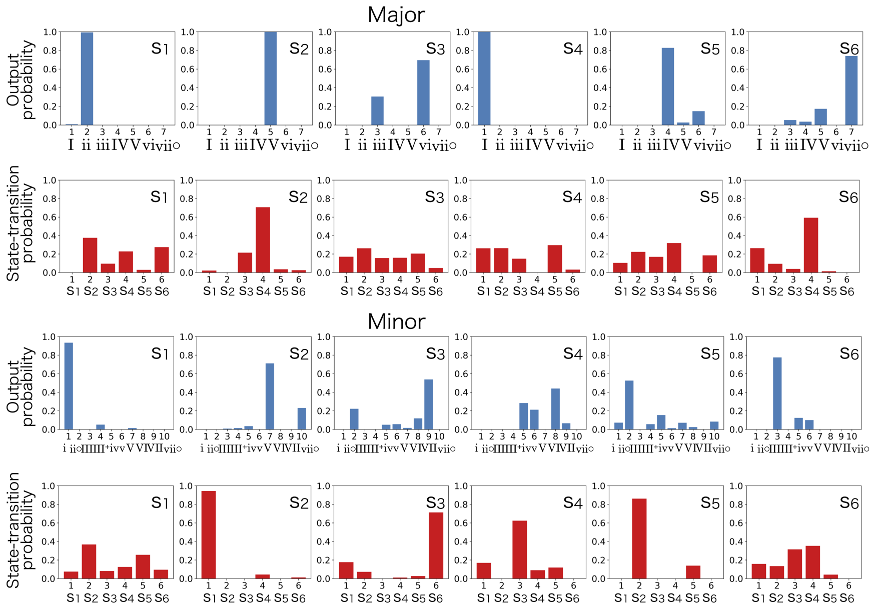

Fig. 6. Output and transition probabilities of 6 hidden states.

to this finding, we found how the same chord could have different functions by observing the value of state-transition probability. As shown in Fig. 6, **ii**$^{\circ}$ appears in both hidden states s_3 and s_5. Here, state s_5 is Sub-dominant since it tends to proceed to state s_2 which is clearly Dominant. On the other hand, **ii**$^{\circ}$ in state s_3 can be interpreted as the Dominant of the relative major key since it mainly proceeds to state s_6, which represents **III** corresponding to **I** of relative major key.

4.4 Example of the modulation detection

Although we calculated the sum of the likelihood on separated phrases to reduce the computation time as mentioned in section 4.2, we can detect the modulation on the entire piece. Since pieces of classical music often have a number of modulations and their phrase boundaries are usually not explicitly indicated, this fully dynamic modulation detection is practically useful.

For example, Fig. 7 shows the modulation detection for the piece BWV271. The initial key of this piece is D major, while the key at the end is B minor with a half cadence. This piece has key blocks in D major, B minor, E minor, and A major. The proposed method captured the modulations for the most part.

Fig. 7. Modulation detection for the piece BWV271. The 'No.' denotes serial numbers, 'Chord' denotes chord names, 'Key' denotes keys and block numbers obtained by the proposed method, and 'State' denotes HMM state labels.

5 Conclusion

We have employed the Hidden Markov Model (HMM) to identify the chord functions, regarding the surface chord degrees as observed outputs. First, we have looked for the best parameterization for each number of hidden states by perplexity, and then, we evaluated the best likelihood of partitioning by modulation. We found that the most adequate number of hidden states was six, which is not large, and thus we could give the fine-grained interpretations for chord functions; *e.g.*, the Dominant **V** and $\mathbf{vii}^\circ$ had different tendency towards $\{\mathbf{iii}, \mathbf{vi}\}$, or the subdominant **IV** and **ii** behaved differently toward the Dominant.

We have applied those chord functions to the partitioning by modulation. The interval of modulation was determined dynamically without fixing the scope beforehand, however, the resultant score of partitioning was also fed back to the number of hidden states. Thus, this process is a tandem model, which is one of the most important features of our work.

Another important feature is the characterization of music styles by parameters. In our example, the set of parameters reflects the specific feature of Bach's chorales, where the basic melodies are of church modes while the harmonization is in the Baroque style. In general, other sets of parameters may have a potential to characterize different music styles such as post-romanticism.

Since our main objective was the key identification, we excluded those borrowed chords and assigned an artificial penalty value to them. Thus, to investigate the key recognition with extraneous chords is our immediate future work. And also, the evaluation with human annotations is our another important future work, even though the human recognition of modulations could admit multiple interpretations. In addition, although we have realized an efficient modulation detection, our method included such errors to regard groups of chords as modulation. To solve this issue, we plan to introduce the notion of dependency in chords, that is to assess the prolongation of the influence of preceding chords.

References

1. Cuthbert, M. S., Ariza, C.: music21: A toolkit for computer-aided musicology and symbolic music data. In: 11th International Society for Music Information Retrieval Conference, pp. 637–642 (2010)
2. Goldwater, S., Griffiths, T. L.: A fully Bayesian approach to unsupervised part-of-speech tagging. In: 45th Annual Meeting of the Association of Computational Linguistics, pp. 744–751 (2007)
3. Hu, D. J., Saul, L. K.: A Probabilistic Topic Model for Unsupervised Learning of Musical Key-Profiles. In: 10th International Society for Music Information Retrieval Conference, pp. 441–446 (2009)
4. Jacoby, N., Tishby, N., Tymoczko, D.: An Information Theoretic Approach to Chord Categorization and Functional Harmony. J. New Music Res., vol. 44(3), pp.219–244 (2015)
5. Krumhansl, Carol L. and Kessler, E. J.: Tracing the dynamic changes in perceived tonal organisation in a spatial representation of musical keys Key-Finding with Interval Profiles. Psychological Review, vol. 89(2), pp.334–368 (1982)
6. Lerdahl, F.: Tonal pitch space. Oxford University Press (2004)
7. Pedregosa, F., et al.: Scikit-learn: Machine learning in Python. J. Machine Learning Res., vol. 12, pp.2825–2830 (2011)
8. Rohrmeier, M., Cross, I.: Statistical Properties of Tonal Harmony in Bach's Chorales. In: 10th International Conference on Music Perception and Cognition, pp. 619–627 (2008)
9. Sakamoto, S., Arn, S., Matsubara, M., Tojo, S.: Harmonic analysis based on Tonal Pitch Space. In: 8th International Conference on Knowledge and Systems Engineering, pp. 230–233 (2016)
10. Temperley, D.: The Tonal Properties of Pitch-Class Sets: Tonal Implication, Tonal Ambiguity, and Tonalness. Computing in Musicology, vol. 15, pp.24–38. Center for Computer Assisted Research in the Humanities at Stanford University (2007)
11. Tsushima, H., Nakamura, E., Itoyama, K., Yoshii, K.: Function- and Rhythm-Aware Melody Harmonization Based on Tree-Structured Parsing and Split-Merge Sampling of Chord Sequences. In: 18th International Society for Music Information Retrieval Conference, pp. 502–508 (2017).
12. Tsushima, H., Nakamura, E., Itoyama, K., Yoshii, K.: Generative statistical models with self-emergent grammar of chord sequences. J. New Music Res., vol. 47(3), pp.226–248 (2018)
13. White, C. W., Quinn, I.: The Yale-Classical Archives Corpus. Empirical Musicology Review, vol. 11(1), pp.50-58 (2016)