UNSUPERVISED MELODY STYLE CONVERSION

Eita Nakamura[1], *Kentaro Shibata*[1], *Ryo Nishikimi*[1], *Kazuyoshi Yoshii*[1,2]

[1]Kyoto University, Kyoto, Japan, [2]RIKEN AIP, Tokyo, Japan

ABSTRACT

We study a method for converting the music style of a given melody to a target style (e.g. from classical music style to pop music style) based on unsupervised statistical learning. Following the analogy with machine translation, we propose a statistical formulation of style conversion based on integration of a music language model of the target style and an edit model representing the similarity between the original and arranged melodies. In supervised-learning approaches for constructing style-specific language models, it has been crucial to use data that properly specify a music style. To reduce reliance on manual data selection and annotation, we propose a novel statistical model that can spontaneously discover styles in pitch and rhythm organization. We also point out the importance of an edit model that incorporates syntactic functions of notes such as tonic and build a model that can infer such functions unsupervisedly. We confirm that the proposed method improves the quality of arrangement by examining the results and by subjective evaluation.

***Index Terms*—** Symbolic music processing; music arrangement; style conversion; statistical music language models; unsupervised grammar induction.

1. INTRODUCTION

Computational understanding of music creation has been a challenge in artificial intelligence [1–4] and recently it has gathered much attention for use in applications such as automatic music generation [5–17]. Music arrangement for converting the music style, e.g. from classical music style to pop music style, is an important creative process to increase the variety of music [18–22]. A few speculations about this process lead to interesting questions such as 'How can we computationally define and model music styles?' and 'What musical characteristics are kept invariant for style conversion to maintain similarity between the original and arranged music?'. Here, we address these questions from the viewpoint of statistical learning and develop a method for melody style conversion that uses minimal prior musical knowledge and annotated data.

Style conversion (and other types of music arrangement [9, 14]) can be considered in analogy with machine translation: one needs to generate a melody [sentence] that fits into the target music style [language grammar] and retains the characteristics [meaning] of the original melody [sentence]. To achieve this, we follow the formalism of statistical machine translation [23] and propose a framework for style conversion based on integration of a style-specific language model representing the target style and an edit model representing the similarity between original and arranged melodies.

In most work on music generation [8, 11, 12, 17, 20, 22], style-specific models have been built with training data of a specific 'music category' (genre, composer, etc.). However, music categories do not often correspond to well-defined music styles [24–27]. For example, in the music of 'Mozart' or 'The Beatles', different styles of pitch organization (e.g. major and minor modes) and of rhythms (e.g. quarter-note rhythm, 8th-note rhythm, and dotted rhythm) coexist. Training data of mixed styles may obscure and even harm the learned style, leading to unsuccessful music generation [15–17]. Successful results have been obtained with well-selected and annotated training data (e.g. with key information) [11, 12, 28], but it requires much cost to do this for all possible target styles. To reduce the cost of manual selection and annotation of data, we need a method that can spontaneously discover styles from data [25, 29].

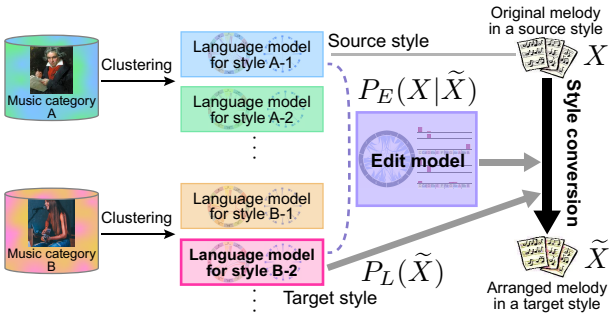

Fig. 1. Framework for style conversion. Style-specific language models and edit models are obtained by unsupervised learning.

For machine translation, a corpus of bilingual parallel text is typically used for building an edit model. For music style conversion, the cost of preparing such parallel data is very high, which calls for an unsupervised method for constructing edit models. Studies have revealed that tonal functions (e.g. tonic and leading tone) are important factors for music similarity, in addition to the geometric distance of notes [30, 31]. In the general situation that source and target styles have unknown and different musical systems, the challenge is to infer such syntactic functions of notes and the relations between the functions in the two styles without using annotated data.

In this paper, we propose a framework for unsupervised music style conversion, focusing on the domain of melodies (Fig. 1). First, we present a statistical formulation of style conversion based on integration of language and edit models. Second, we develop a method for discovering styles from data by clustering characteristics in pitch organization and rhythms, based on unsupervised learning of a mixture of probabilistic sequential models. For this, we construct a novel Markov model with an architecture to embody metrical structure, transposition-symmetric structure of pitches such as musical scales, and interdependence of pitches and rhythms. Third, we build an edit model that incorporates both the geometric distance of notes and their syntactic functions. This is realized by means of a hidden Markov model (HMM) that spontaneously learns common syntax underlying two style-specific music language models. We examine the effect of the proposed method by a subjective evaluation experiment and by inspecting arranged melodies.

The work was supported by JST ACCEL No. JPMJAC1602, JSPS KAKENHI Nos. 16H01744, 16H02917, 16K00501, and 16J05486, Kayamori Foundation, and the Kyoto University Foundation.

The main results of this study are

- A rigid formulation of music style conversion based on the combination of a style-specific language model and an edit model. Previously, an edit model has not been formulated [18–21].
- Music language models capturing meaningful styles such as musical scales and typical rhythms can be obtained unsupervisedly.
- Unsupervised construction of edit models incorporating syntactic relations of notes without parallel data or annotated data.
- Both the refinements of the music language model and the edit model improve the quality of melody style conversion.

2. PROPOSED METHOD

2.1. Statistical Formulation of Style Conversion

Let us first formalize the problem of melody style conversion. A melody is described as a sequence $((p_{mn}, s_{mn})_{n=1}^{N_m})_{m=1}^{M}$, where p_{mn} is the pitch of the nth note in the mth bar and s_{mn} is the onset score time of that note, both of which take integer values (M is the number of bars and N_m is the number of notes in the mth bar). We here focus on pieces in 4/4 time for simplicity and represent score times s_{mn} in units of $1/3$ s of a 16th note. We assume that the number of notes in each bar and the octave ranges of notes are kept invariant under style conversion. With this assumption, we can represent pitches p_{mn} by their relative values (called *pitch classes*) $q_{mn} \in \{0, \dots, 11\}$ in each octave range ($q_{mn} \equiv p_{mn} \bmod 12$) and score times s_{mn} by their relative values (called *beat positions*) $b_{mn} \in \{0, \cdots, 47\}$ in each bar ($b_{mn} \equiv s_{mn} \bmod 48$). Thus, we can represent a melody X as $X = ((q_{mn}, b_{mn})_{n=1}^{N_m})_{m=1}^{M}$. A method for melody style conversion is defined as an algorithm that maps an original melody X belonging to a source music style to an arranged melody $\widetilde{X} = ((\tilde{q}_{mn}, \tilde{b}_{mn})_{n=1}^{N_m})_{m=1}^{M}$ belonging to a target music style. As necessary conditions for successful style conversion, we require that an arranged melody matches a target style and that humans can feel the original melody in the arranged melody.

In the statistical formulation, we model the probability $P(\widetilde{X}|X)$ of the arranged melody $\widetilde{X}$ given the original melody X. Similarly as for statistical machine translation [23], we decompose this probability as $P(\widetilde{X}|X) \propto P_L(\widetilde{X})P_E(X|\widetilde{X})$, where P_L represents a *target language model* and P_E represents an *edit model*. The purpose of the target language model is to describe the characteristics of the target music style and that of the edit model is to embody the (content) similarity between melodies X and $\widetilde{X}$.

2.2. Music Language Model

A minimal model for the sequence of pitch classes (pcs) is the pitch-class Markov model (PcMM) defined with initial probabilities $P(q_{11} = q)$ and transition probabilities as $P(q_{mn} = q \,|\, q'_{mn} = q')$. Hereafter, we write q'_{mn} (and similarly s'_{mn} etc.) for the note that comes just before the q_{mn}. Similarly, a minimal model for beat positions is the metrical Markov model (MetMM) [32, 33] defined with probabilities $P(b_{11} = b)$ and $P(b_{mn} = b \,|\, b'_{mn} = b')$.

To incorporate interdependence of pitches and rhythms, we combine the PcMM and MetMM by considering the product space of pc and beat position. The transition probabilities are written as $P(q_{mn}, b_{mn} \,|\, q'_{mn}, b'_{mn}) = \Psi(q'_{mn}, b'_{mn}; q_{mn}, b_{mn})$. Hereafter, to save space, initial probabilities are not written down explicitly but readers should understand that they are similarly defined. We call this model a *torus Markov model* (TMM) since the state space can be identified as a grid on a torus $S^1 \times S^1$ with one circle corresponding to the pc space and the other one corresponding to the beat-position space. Note that even with the first-order restriction, the TMM can induce a range of dependence up to a bar length since pcs on different beat positions are treated as independent states.

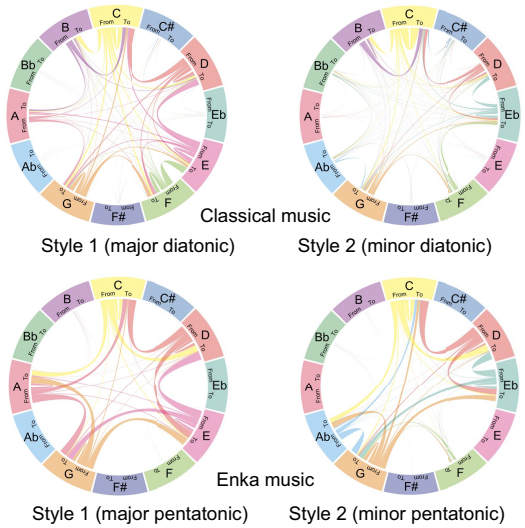

Fig. 2. Learned parameters of the TSTMMixMs. Marginalized pitch-class transition probabilities are shown as bands.

To describe both transpositions (global pitch shifts of entire musical pieces) and modulations (local pitch shifts of phrases or sections of a musical piece), we introduce (local) key variables $k_m \in \{0, \dots, 11\}$ defined for each bar m. For example, when $k_m = 0$ indicates C major key, D major key is indicated by $k_m = 2$. We suppose that key variables are generated by a Markov model $P(k_m|k_{m-1})$ and extend the TMM as

$$P(k_m = k \,|\, k_{m-1} = k') = \pi_{k'k}, \tag{1}$$

$$P(q_{mn}, b_{mn} \,|\, q'_{mn}, b'_{mn}, k_m) = \Psi^{(k_m)}(q'_{mn}, b'_{mn}; q_{mn}, b_{mn}). \tag{2}$$

To relate parameters for different keys, we impose transposition symmetry for the model parameters (similarly as in [34]):

$$\pi_{k'k} = \pi_{(k'+\ell)(k+\ell)}, \tag{3}$$

$$\Psi^{(k)}(q', b'; q, b) = \Psi^{(k+\ell)}(q' + \ell, b'; q + \ell, b), \tag{4}$$

for any $\ell \in \{0, \dots, 11\}$ (additions for pcs and keys are defined in modulo 12). We call this model a transposition-symmetric TMM (TSTMM). A transposition-symmetric PcMM is defined similarly.

As discussed in Sec. 1, there are commonly several modes of pitch and rhythm organizations in a music category. With the expectation that these modes can be represented by different parameter values of TSTMMs, we construct a mixture model by introducing mode variables $\rho_m \in \{1, \dots, N_{\mathrm{M}}\}$ defined for each bar. For each mode we consider 12 transpositions indexed by k_m. The generative process is now described as

$$P(\rho_m = \rho, k_m = k \,|\, \rho_{m-1} = \rho', k_{m-1} = k') = \pi_{\rho' k', \rho k}, \tag{5}$$

$$P(q_{mn}, b_{mn} \,|\, q'_{mn}, b'_{mn}, \rho_m = \rho, k_m = k) = \Psi^{(\rho,k)}(q'_{mn}, b'_{mn}; q_{mn}, b_{mn}). \tag{6}$$

We again impose transposition symmetry as in Eqs. (3) and (4). This model is called a *transposition-symmetric torus Markov mixture model* (TSTMMixM). A component TSTMM defines $P_L(\widetilde{X})$.

TSTMMixMs can be learned unsupervisedly by the EM algorithm [35]. Although one can use random initialization in princi-

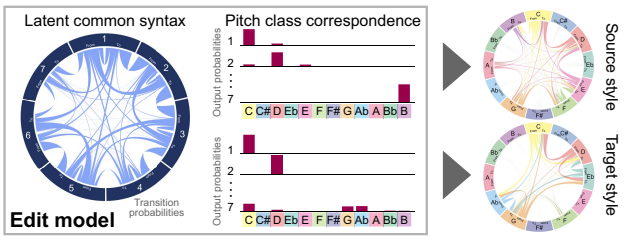

Fig. 3. Syntactic functions of notes described in the edit model.

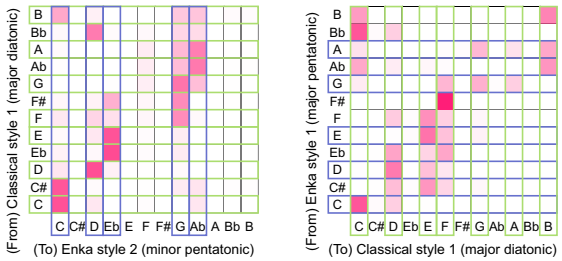

Fig. 4. Learned edit probabilities $P(q|\tilde{q})$ for two sets of source and target music styles. Thick rectangles indicate principal scale notes.

ple, we empirically found that this often leads to unwanted local optima. To solve this problem, we can learn a mixture of transposition-symmetric PcMMs and that of MetMMs separately and then use them to initialize the learning of TSTMMixMs.

The two TSTMMixMs learned from the classical music dataset and the Enka dataset (see Sec. 3.1 for details) are illustrated in Fig. 2. Here, the PcMMs obtained by marginalizing the component TMMs are visualized, and the pcs are transposed to make interpretation easier. For the classical music data the two models represent the major and minor (diatonic) scales, and for the Enka data the two models represent the major and minor pentatonic scales. Major and minor scales are similarly learned from the J-pop data. While these scales are well-known and expected to be extracted from data, it is worth emphasizing that they are here inferred unsupervisedly without any annotation on the tonic and mode. Similar visualizations for other styles are accessible from the accompanying web page [36].

We can also see that different rhythmic styles are learned from different music categories (see visualizations in [36]). Most notably, onsets on stronger beats are more frequent in the styles of the classical music, whereas this is not true in the styles of J-pop music reflecting the presence of frequent syncopations.

2.3. Edit Model

If note $x{=}(q, b)$ of the original melody corresponds to note $\tilde{x}{=}(\tilde{q}, \tilde{b})$ of the arranged melody, a *simple edit model* can be defined by

$$P(x|\tilde{x}) \propto \exp\left(-\frac{(q-\tilde{q})^2}{2\sigma_p^2}\right)\exp\left(-\frac{(b-\tilde{b})^2}{2\sigma_r^2}\right), \qquad (7)$$

where the squared distances are defined in the spaces of pc and beat position, and σ_p and σ_r are scale parameters.

The simple edit model has an essential problem, especially in the unsupervised setup. For example, if one converts a C-major melody into the minor mode, normally one chooses C minor key for the arranged melody and retains the tonic note (C), which is often used as a closing note. However, with the simple edit model, A minor key (or another minor key) is often selected as the one that minimizes the geometric distances of notes and then the structure of syntactic functions of notes (such as tonic) is not retained.

To solve this problem, we construct a refined edit model that takes into account syntactic functions of notes. To infer syntactic functions of notes from data unsupervisedly, we apply the technique developed in [37]. This method uses HMMs with latent states representing functions of symbols, which can be trained using sequential contexts, i.e. what comes before and after a certain symbol. Writing $z_{mn} \in \{1, \ldots, N_{\mathrm{F}}\}$ for a latent state (N_{F} is a predefined number of syntactic functions), the HMM is defined wth transition probabilities $P(z_{mn}|z'_{mn})$ and output probabilities $P(q_{mn}|z_{mn})$, which are to approximate the probability $P(q|q')$ of a TSTMMixM.

To construct an edit model connecting two styles, we extend this model with two output probabilities, one for the source style $P(q_{mn}|z_{mn})$ and the other for the target style $P(\tilde{q}_{mn}|z_{mn})$ (Fig. 3). The latent states and transition probabilities $P(z_{mn}|z'_{mn})$ are shared in the two styles to induce the model to represent common syntactic structure. These probabilities are to approximate both $P(q|q')$ and $P(\tilde{q}|\tilde{q}')$. This model can induce the following edit probability:

$$P_F(X|\widetilde{X}) = \sum_{\boldsymbol{z}} \left[\prod_{m,n} P(q_{mn}|z_{mn})\right] P(\boldsymbol{z}|\tilde{\boldsymbol{q}}), \qquad (8)$$

where $\boldsymbol{z} = (z_{mn})$ and $\tilde{\boldsymbol{q}} = (\tilde{q}_{mn})$. The second factor in the left-hand side can be computed by the forward-backward algorithm.

Combining both the distance-based model and the function-based model, the *refined edit model* is defined as

$$P_E(X|\widetilde{X}) \propto P_F(X|\widetilde{X})^{\alpha_1} \prod_{m,n} P_D(q_{mn}|\tilde{q}_{mn})^{\alpha_2} P_D(b_{mn}|\tilde{b}_{mn})^{\alpha_3},$$

where $P_D(q|\tilde{q})$ and $P_D(b|\tilde{b})$ denote the two factors in Eq. (7) and we have introduced weights α_1, α_2, and α_3 for the component models.

Examples of learned parameters of the edit model are shown in Fig. 4, where the heat maps represent the probabilities $P(q|\tilde{q}) \propto \sum_z P(q|z)P(\tilde{q}|z)\pi_z^*$ (π_z^* is the stationary distribution of z). In the left figure, notes of the major diatonic scale are mapped to those of the minor pentatonic scale, and one can find the mediant and submediant (E and A) are mapped mainly to flat notes (E♭ and A♭), which agrees with the musical intuition. In the right figure, the major pentatonic scale is mapped to the major diatonic scale. Notably, the fourth and fifth notes (G and A) of the pentatonic scale correspond to multiple notes of the diatonic scale. The functions of these notes can change depending on the context; e.g. A before C in the pentatonic scale can correspond to the leading tone B in the diatonic scale. As in these examples, we found that the refined edit models obtained by unsupervised learning often matched the musical intuition.

2.4. Algorithm for Melody Style Conversion

An algorithm for melody style conversion can be derived based on statistical inference of the combination of the language model in Sec. 2.2 and the edit model in Sec. 2.3. We first learn one set of a TSTMMixM for each dataset of the source and target music categories and extract music styles indexed by the mode variable ρ_{source} and ρ_{target}. If an original melody X and the corresponding music style ρ_{source} are given, then the key information can be estimated by the Viterbi algorithm using the source language model.

The target music style is specified by one of the values of ρ_{target}. One can infer the target melody $\widetilde{X}$ and its key information with respect to the target language model jointly by maximizing the probability $P(\widetilde{X}|X) \propto P_E(X|\widetilde{X})P_L(\widetilde{X})$, which can be done with the Viterbi-like algorithm. For efficient computation, after the key information of the original melody is estimated we simply transfer this information to the target melody using the edit model since the accuracy of key estimation is high (100% accuracy for our test data).

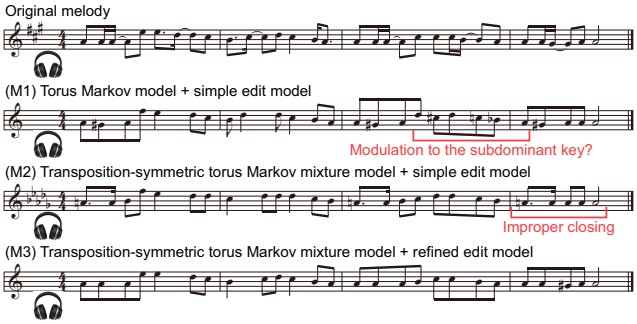

Fig. 5. Examples of melody style conversion. The original melody is a J-pop song. The target category is the classical music and the 2nd style (minor mode) is used as the target for methods M2 and M3.

3. RESULTS AND EVALUATION

3.1. Experimental Setup

For numerical experiments, we use three datasets of different music categories, (Western) classical music, J(apanese)-pop music, and Enka music (a genre of Japanese popular song), which were chosen for the ease of data preparation and subjective evaluation. The classical music data consist of 7133 bars of soprano melodies composed by Mozart, the J-pop data consist of 3878 bars of vocal melodies composed by a Japanese band 'Mr. Children', and the Enka data consist of 37032 bars of vocal melodies by various artists [38, 39].

The language model for each music category is trained as follows. First, N_{PM} mixtures of PcMMs and N_{RM} mixtures of MetMMs are learned by the EM algorithm. Then, all combinations of products of these models are used as initial values for learning the TSTMMixMs with $N_{\mathrm{M}} = N_{\mathrm{PM}}N_{\mathrm{RM}}$ mixtures. The numbers of mixtures for the test data are set to $(N_{\mathrm{PM}}, N_{\mathrm{RM}}) = (3, 1), (3, 2)$, and $(3, 3)$ for the classical music, J-pop, and Enka data, respectively. Out of the learned TSTMMs, we choose two that have large mixture weights, i.e. most frequently appearing one, and use them as representative styles of each music category. After a few trials, we set $\alpha_1 = 0.4$, $\alpha_2 = \alpha_3 = 0.8$, $\sigma_p = 0.7$, $\sigma_r = 3$, and $N_{\mathrm{F}} = 7$. For comparison, we implement and test the following three methods:

- (M1) Torus Markov model (TMM) + simple edit model
- (M2) TSTMMixM + simple edit model
- (M3) TSTMMixM + refined edit model

3.2. Example Results

Examples of melody style conversion are shown in Fig. 5, where a J-pop melody is converted to a style of the classical music category. In the rhythmic aspect, results for the three methods are all successful: tied notes that are typical for J-pop songs are replaced with modest rhythms typical for the classical music style. On the other hand, three arrangements have different pitch organizations. In the result for M1, accidentals in the third bar imply a modulation to the subdominant key, which is not present in the original melody, leading to an unstable closing. This can be explained by the fact that the key structure is not described in the TMM. In the results for M2 and M3, the key structure is consistent. However, the result for M2 has the key of B-flat minor and the closing notes are not properly converted. This can be explained by the fact that functions of notes such as tonic are not modelled in the simple edit model. One can find no such problems in the pitch organization in the result for M3.

Similar tendencies can be found in other examples (see [36]). Results for M1 often have unnatural key structure and sometimes do not raise sense of tonality. Results for M2 and M3 are often similar when the keys of the target melodies are same. Otherwise, results for M2 can have unnatural tonal structure, when functions of notes or key structure are improperly transferred to the arranged melody.

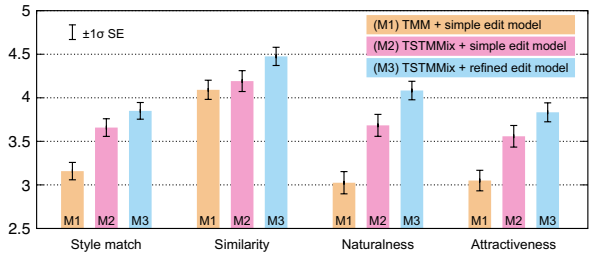

Fig. 6. Result of the subjective evaluation. Main bars indicate means and error bars indicate 1σ standard errors.

3.3. Subjective Evaluation

We conducted a subjective evaluation test to measure the quality of style conversion. 10 evaluators who listen to music more than one hour a day participated the experiment. Two well-known melodies (8-bar length) are chosen in each of the three categories (classical, J-pop, and Enka) and are converted to one style of the two different music categories; in total we have 12 arranged melodies for each method (results are accessible in [36]). After evaluators listened to the arranged melodies, being informed the target styles but not the methods, they evaluated the following metrics in 6-level scores:

- *Style match:* Does the arranged melody match the target style?
- *Similarity:* Do you feel the original melody?
- *Naturalness:* Is the melody natural?
- *Attractiveness:* Is the melody attractive?

The results in Fig. 6 show that the mean scores of all the metrics are improved by refinements of the method. Particularly, the 'style match' score improved by 0.5 (p-value $< 10^{-5}$, t-test) with the refined language model (M1 vs M2), and the 'similarity' score improved by 0.28 (p-value $= 3.3 \times 10^{-3}$, t-test) with the refined edit model (M2 vs M3). The improvements in the 'naturalness' and 'attractiveness' scores are also statistically significant. These results clearly demonstrate the efficacy of the proposed method.

4. CONCLUSION

Back to our questions, the results of this study indicate that aspects of music styles such as musical scales and typical rhythms can be described as clusters defined by statistical generative models, which can be learned without much relying on expert musical knowledge. Unsupervised learning of music styles has been studied for use in genre classification [25, 29], and we showed that it is also useful for generating or arranging music. We also revealed the importance of syntactic functions of musical notes in describing music similarity for the arrangement task, and the effect is particularly enhanced when the key information must be estimated automatically. The generality of our framework and the promising results suggest that the proposed formulation of style conversion can also be useful for other forms of music such as chord sequences and polyphonic music.

The present framework can easily be applied to other music styles and we plan to examine the universality of the approach in a wide variety of music styles. We found that different music styles are obtained by different initial values from the same data and the results are not always interpretable. How to characterize musically meaningful clustering of styles in terms of information measures (e.g. likelihood) is therefore essential. Determining the optimal number of mixtures is also an important issue left for future work.

5. REFERENCES

[1] G. Papadopoulos and G. Wiggins, "AI methods for algorithmic composition: A survey, a critical view and future prospects," in *Proc. AISB Symposium on Musical Creativity*, 1999, vol. 124, pp. 110–117.

[2] G. Nierhaus, *Algorithmic Composition*, Springer, 2009.

[3] J. D. Fernández and F. Vico, "AI methods in algorithmic composition: A comprehensive survey," *J. Artificial Intelligence Res.*, vol. 48, pp. 513–582, 2013.

[4] J.-P. Briot, G. Hadjeres, and F. Pachet, "Deep learning techniques for music generation—A survey," *arXiv preprint arXiv:1709.01620*, 2017.

[5] F. Pachet, "The continuator: Musical interaction with style," *J. New Music Res.*, vol. 32, no. 3, pp. 333–341, 2003.

[6] H. Maekawa et al., "On machine arrangement for smaller wind-orchestras based on scores for standard wind-orchestras," in *Proc. ICMPC*, 2006, pp. 268–273.

[7] S. Fukayama et al., "Automatic song composition from the lyrics exploiting prosody of the japanese language," in *Proc. SMC*, 2010, pp. 299–302.

[8] F. Pachet and P. Roy, "Markov constraints: Steerable generation of Markov sequences," *Constraints*, vol. 16, no. 2, pp. 148–172, 2011.

[9] G. Hori, H. Kameoka, and S. Sagayama, "Input-output HMM applied to automatic arrangement for guitars," *J. Info. Processing Soc. Japan*, vol. 21, no. 3, pp. 264–271, 2013.

[10] M. McVicar, S. Fukayama, and M. Goto, "AutoLeadGuitar: Automatic generation of guitar solo phrases in the tablature space," in *Proc. ICSP*, 2014, pp. 599–604.

[11] B. L. Sturm et al., "Music transcription modelling and composition using deep learning," in *Proc. CSMC*, 2016, pp. 1–16.

[12] G. Hadjeres and F. Pachet, "DeepBach: A steerable model for Bach chorales generation," *arXiv preprint arXiv:1612.01010*, 2016.

[13] L. Crestel and P. Esling, "Live orchestral piano, a system for real-time orchestral music generation," in *Proc. SMC*, 2017, pp. 434–442.

[14] E. Nakamura and K. Yoshii, "Statistical piano reduction controlling performance difficulty," *APSIPA Trans. on Signal and Information Processing*, 2018, to appear.

[15] H.-W. Dong et al., "MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proc. AAAI*, 2018.

[16] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," *arXiv preprint arXiv:1703.10847*, 2017.

[17] H. H. Mao, T. Shin, and G. Cottrell, "DeepJ: Style-specific music generation," in *Proc. IEEE ICSC*, 2018, pp. 377–382.

[18] D. Tzimeas and E. Mangina, "Jazz Sebastian Bach: A GA system for music style modification," in *Proc. IEEE ICSNC*, 2006, pp. 36–42.

[19] F. Zalkow, S. Brand, and B. Graf, "Musical style modification as an optimization problem," in *Proc. ICMC*, 2016, pp. 206–2011.

[20] W-T. Lu and L. Su, "Transferring the style of homophonic music using recurrent neural networks and autoregressive models," in *Proc. ISMIR*, 2018, pp. 740–746.

[21] G. Brunner et al., "Symbolic music genre transfer with CycleGAN," in *Proc. IEEE ICTAI*, 2018, pp. 786–793.

[22] N. Mor et al., "A universal music translation network," *arXiv preprint arXiv:1805.07848*, 2018.

[23] P. F. Brown et al., "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[24] R. L. Crocker, *A History of Musical Style*, McGraw-Hill, 1966.

[25] J.-J. Aucouturier and F. Pachet, "Representing musical genre: A state of the art," *J. New Music Res.*, vol. 32, no. 1, pp. 83–93, 2003.

[26] N. Scaringella, G. Zoia, and D. Mlynek, "Automatic genre classification of music content: A survey," *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 133–141, 2006.

[27] B. L. Sturm, "Classification accuracy is not enough," *J. Intelligent Information Systems*, vol. 41, no. 3, pp. 371–406, 2013.

[28] J. Sakellariou et al., "Maximum entropy models capture melodic styles," *Sci. Rep.*, vol. 7, no. 9172, pp. 1–9, 2017.

[29] X. Shao, C. Xu, and M. S. Kankanhalli, "Unsupervised classification of music genre using hidden Markov model," in *Proc. ICME*, 2004, vol. 4, pp. 2023–2026.

[30] C. L. Krumhansl, "The psychological representation of musical pitch in a tonal context," *Cognitive Psychology*, vol. 11, no. 3, pp. 346–374, 1979.

[31] P. Hanna, P. Ferraro, and M. Robine, "On optimizing the editing algorithms for evaluating similarity between monophonic musical sequences," *J. New Music Res.*, vol. 36, no. 4, pp. 267–279, 2007.

[32] C. Raphael, "A hybrid graphical model for rhythmic parsing," *Artificial Intelligence*, vol. 137, pp. 217–238, 2002.

[33] M. Hamanaka et al., "A learning-based quantization: Unsupervised estimation of the model parameters," in *Proc. ICMC*, 2003, pp. 369–372.

[34] D. Hu and L. K. Saul, "A probabilistic topic model for unsupervised learning of musical key-profiles," in *Proc. ISMIR*, 2009, pp. 441–446.

[35] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

[36] Supplemental material (available online), http://melodyarrangement.github.io/demo.html.

[37] H. Tsushima et al., "Generative statistical models with self-emergent grammar of chord sequences," *J. New Music Res.*, vol. 47, no. 3, pp. 226–248, 2018.

[38] Y. Goto (ed.), *Grand Collection of Enka Songs by Male Singers 5th Ed. (in Japanese)*, Zen-on Music Co., 2016.

[39] Y. Goto (ed.), *Grand Collection of Enka Songs by Female Singers 5th Ed. (in Japanese)*, Zen-on Music Co., 2016.