Towards balanced tunes: A review of symbolic music representations and their hierarchical modeling

Nádia Carvalho
University of Porto, Faculty of Engineering
Porto, Portugal
ei12047@fe.up.pt

Gilberto Bernardes
INESC TEC
University of Porto, Faculty of Engineering
Porto, Portugal
gba@fe.up.pt

Abstract

Since the heydays of music informatics, around the 1950s, the modeling and prediction of musical structures manifested as symbolic representations have been continuously pursued. The operational property of such methods is to provide the conditional distribution over an alphabet – i.e., the entire collection of unique musical events in a composition or corpus – given a context – i.e., a preceding sequence. This distribution unpacks temporal morphologies that support multiple applications for predictive and assisted creative tasks, such as the generation of new musical sequences that retain a structural resemblance to a modeled source. Despite their longstanding tradition, state-of-the-art methodologies for symbolic music modeling are yet to reach the music community. Naive models such as Markov chains, which are known to neglect the fundamental hierarchical nature of musical structure, remain common practice. In this paper, we extensively review existing methodologies for symbolic music representation and modeling, as the first steps towards a study on the resulting balance across familiarity and novelty in generative music applications.

Introduction

Historically, music informatics has been exploring the algorithmic modeling and prediction of musical structure. Existing applications stem from information theory principles and the postulate of music as a low entropy phenomenon (Conklin and Witten, 1995). Given the temporal and hierarchical nature of the musical structure, algorithmic methods are typically informed by a sequence of past events, i.e., a context, to both model existing structures and predict or generate new structures (Conklin and Anagnostopoulou, 2001). These models aim to capture different degrees of inter-dependency across the component elements of the musical structure. Prior to the modeling of musical structure, a discrete and finite alphabet including all unique symbolic *representations* for a given structure has to be created. Depending on the adopted intra- and inter-opus musical material, algorithmic models capture different musical traits ranging from recurrent patterns in a composition to stylistic idiosyncrasies of a composer or even tonal music principles.

The balance between familiarity to known compositional traits, captured by these algorithmic methods and novelty introduced by unfamiliar and unpredictable structures is of utter importance in the design of generative systems (Bevington and Knox, 2014). The Wundt curve, a hedonic function that relates the levels of novelty and expectation to the 'pleasantness' of creative works (Berlyne, 1970), captures the notion of balance as mentioned above.

In this paper, we argue that the interaction between discrete and finite alphabets of music and their temporal modeling is instrumental in controlling the resulting balanced of generative music models across the novelty-familiarity range. To this end, we extensive review musical representations and modeling methods adopted in the context of generative music, as the first steps towards a larger study on their (balanced) interaction thereof.

The remainder of this paper is structured as follows. Section "Symbolic Representation of Musical Structures" provides a literature review on the topic. Section "Modeling Temporal Musical Structures" presents modeling methods that capture the morphology of musical structures. Section "Applications" reviews representative generative music applications, which combine the two above components and have a broader adoption by the music community. A twofold categorization of computer-aided algorithmic composition and machine improvisation applications is adopted. Finally, Section "Summary and Future Challenges" presents the conclusions and discusses future challenges.

Symbolic Representation of Musical Structures

In this section, we review the following four symbolic music representations adopted in the computational modeling of musical structure: formal strings, graphs, formal grammars, and geometrical representations. These representations were selected based on their focus, relevancy and impact in improving the models for musical structure modeling and prediction across related literature.

Formal Strings

Formal strings are one of the earliest and most frequently adopted computational representations of symbolic music manifestations. It encodes musical structure as sequences of symbols driven from a finite and discrete alphabet, Σ. To encode duple pitch-duration information – two primary elements in Western music (Wishart and Emmerson, 1996)

Pitch/Duration Encoding	Encoded Sequence
Common music notation (p_{cmn})	(G,$\natural$,3) (G,$\natural$,3) (D,$\natural$,3) (G,$\natural$,2) (B,$\natural$,3) (C,$\natural$,4) (D,$\natural$,4) (D,$\natural$,4) (B,$\natural$,3) (A,$\natural$,3)
Absolute Pitch (MIDI values)	55 55 50 43 59 60 62 60 59 57
Base-12 (p_{12})	8 8 3 8 12 1 3 1 12 10
Base-21 (p_{21})	13 13 4 13 19 1 4 1 19 16
Base-40 (p_{40})	26 26 9 26 38 3 9 3 38 32
Interval (p_{itv})	0 0 -5 -7 16 1 2 -2 -1 -2
Interval from tonic (p_{ift})	0 0 7 0 4 5 7 5 4 2
Contour (p_c)	0 0 -1 -1 1 1 1 -1 -1 -1
HD-Contour (p_{hdc})	0 0 -3 -3 4 1 1 -1 -1 -1
Absolute time (r_{tabs})	0 1/2 1 3/2 2 9/4 5/2 11/4 3 13/4
Absolute duration (r_{dabs})	1/2 1/2 1/2 1/4 1/4 1/4 1/4 1/4 1/4 1/2
Contour (r_c)	0 0 0 0 -1 0 0 0 0 0
HD-Contour (r_{hdc})	In this case, the resulting string is the same as r_c because there is only changes between close rhythm durations.

Table 1: Multiple encoding of pitch and duration using formal string representations for the musical excerpt shown in Figure 1.

– decoupled symbols from (Σ_p) and (Σ_r) alphabets can be adopted. Table 1 summarizes typical formal string representations for encoding the pitch and duration of the musical excerpt shown in Figure 1.

Figure 1: The first bar of J. S. Bach's Courante of Suite No. 1 in G major, BWV 1007.

The multiple viewpoint systems (Conklin and Witten, 1995) emerged as an extension of the duple pitch-duration formal string representations. It expands the former formal string representations by including secondary structural information, such as metrical position and interval. These systems use domain knowledge to derive new representations for encoding temporal events from the musical structure by abstracting properties types, τ, as summarized in Table 1. To compute each type a function Ψ_τ is adopted. A viewpoint comprises one such function and the set of strings that can be computed. A multiple viewpoint system comprises a collection of different viewpoints, some of which can be *derived* from *basic* viewpoints. Furthermore, as the viewpoints can have correlations, a new type was introduced: the product type ($\tau_x \otimes \tau_y$), whose elements are the cross product of their constituents. Table 2 shows a set of multiple viewpoints of the musical excerpt in Figure 1.

Type	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10
start offset	10	12	14	16	18	19	20	21	22	23
pitch (absolute)	55	55	50	43	59	60	62	60	59	57
duration	2	2	2	2	1	1	1	1	1	1
key signature	1	1	1	1	1	1	1	1	1	1
time signature	12	12	12	12	12	12	12	12	12	12
deltast (is rest?)	F	F	F	F	F	F	F	F	F	F
posinbar (position in bar)	10	0	2	4	6	7	8	9	10	11
fib (is first in bar?)	F	T	F	F	F	F	F	F	F	F
seqint (sequential interval from last note)	$\perp$	0	-5	-7	16	1	2	-2	-1	-2
contour	$\perp$	0	-1	-1	1	1	1	-1	-1	-1
hdcontour	$\perp$	0	-3	-3	4	1	1	-1	-1	-1
referent	7	7	7	7	7	7	7	7	7	7
thrbar (seqint at bars)	$\perp$	0	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
thrqu (seqint at quarters)	$\perp$	0	$\perp$	-12	$\perp$	$\perp$	19	$\perp$	$\perp$	$\perp$

Table 2: Some basic and derived viewpoints for the events of Figure 1.

Polyphonic (i.e., multi-layer) formal string representations can be split into the following three categories: *non-interleaved* (Lemström and Tarhio, 2003), *interleaved* (Pienimäki, 2002) and *onset-based* (Lemström and Tarhio, 2003). The first category encodes polyphonic music textures as independent monophonic layers sequentially. The second category encodes all layers linearly, ordering pitch values sequentially by their onset times. It provides greater flexibility for handling the complex multidimensional nature of polyphonic music; however, it does not highlight the vertical, namely homophonic, nature of the textures. The third category underlines homophonic textures by discriminating all overlapping notes as vertical aggregates. However, the duration information is lost and it can lead to a combinatorial explosion. Hanna et al. (2008) minimize the lack of duration information by fragmenting long notes into notes of fixed duration connected by ties.

The task of capturing all co-dependencies across vertical and horizontal textures is still challenging and has not yet been fully achieved.

Graphs

E-Graph is a representation of monophonic sequences proposed by Marsden (2001). It typically adopts a minimum of two *places* (i.e., nodes) for each symbol, namely time and pitch.

Places can be connected by *elaborations* (i.e., edges), which typically include metrical and pitch information. The latter generates new intermediate places without crossing links, making the representation interpretable as an acyclic graph, hence easily represented as a tree. Elaborations can either be *simple* or *accented*. Simple elaborations refer to insertions between two-note events, such as rests, repetitions, anticipations, passing notes, and octave jumps. Accented elaborations refer to delays, suspensions, and accented passing notes.

E-Graphs have shown great potential in capturing musical patterns in multiple stylistic contexts. However, it was gradually abandoned due to its: i) excessive complexity, ii) ambiguous and multiple representations derived from the same musical sequence, and iii) restriction to melodic layer parsing.

Tagliolato (2008) presented a time-independent graph-based signature of melodic layers as an alternative to E-Graphs. It adopts a reduced 12-tone pitch alphabet, which captures invariant-pitch structures under inversion and retrogradation. Basic rhythmic features and events' order can be encoded in the graph's edges.

Formal Grammars

Formal music grammars, or simply grammars, represent the "intuitions of a listener who is experienced in a musical idiom" (Lerdahl and Jackendoff, 1983, p. 3) using formalized methods (i.e., higher-level abstractions). It can be split into four hierarchies: i) *grouping* into motives, phrases, periods or sections, ii) *metric* alternation of strong and weak beats, iii) *time-span reduction* from metrical and grouping structures to higher-level hierarchies, and iv) *prolongational reductions* describing tension and relaxation phenomena across time (Lerdahl and Jackendoff, 1983). Despite being well-suited for representing structural dependencies in musical structure, formal context-free[1] music grammars are challenging to compute. Their strict hierarchy is difficult to reconcile with the inherent ambiguity of musical structure (Rohrmeier and Pearce, 2018).

Rizo (2010) proposed a formal grammar representation for computing the similarity of monophonic and polyphonic music sequences as tree structures. The tree leaves represent note events and pitch class sets from monophonic and polyphonic layers, respectively. Employing multi-sets with duration information and number of notes overlapping solves the problem of encoding polyphonic information. Rizo showed that tree structures are versatile in encoding information other then pitch or rhythm, such as harmonic structure and form. However, its dependency on *a priori* knowledge of the metric structure and its high complexity in representing ties, dots, and syncopations are prominent drawbacks.

Geometric Representations

Maidín (1998) proposes the use of geometric representations to encode music as 2-dimensional pitch-duration contours across time. From the resulting geometrical representation, various metrics were proposed, namely for similarity computation across musical sequences.

A particular case of these geometric representations is the *multidimensional point sets*, proposed by Meredith (2006), that adopt the Euclidean space to represent musical events as a tuple of 5 elements: onset time of the note; chromatic (absolute) pitch; diatonic pitch, defined by an integer that indicates the position of the head of the note on the staff; duration; and the voice (i.e., layer) number within the polyphonic texture. The 2- or 3-dimensional projection of the point sets allows the efficient search for similar patterns, including small variations, using their spatial configuration. Figure 2 shows a projection of onset time and chromatic pitch for the musical excerpt in Figure 1.

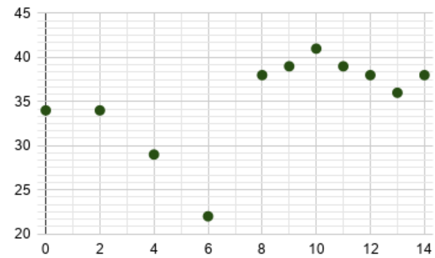

Figure 2: A Multidimensional Point Sets projection of onset time and chromatic pitch for the example in Figure 1.

Modeling Temporal Musical Structures

In this section, we review the following two musical structure modeling techniques, which were selected based on the visibility and attention they attract from the music community: statistical sequence modeling and compression algorithms. Despite the possibility to manually draw these models from scratch, they are typically driven from existing musical structures (e.g., individual pieces or a corpus). Despite falling into the general category of musical structure modeling, we do not include evolutionary computation algorithms in this review, as they do not address domain-specific knowledge (Nierhaus, 2009).

Statistical Sequence Modeling

***N*-grams** are specific types of Markov models, which capture dependencies across discrete and finite symbols from an alphabet, given a context (Downie, 1999). N corresponds to the total number of contiguous symbols under consideration in the model, i.e., the context. Despite its popularity, N-grams have been criticized due to their limitation in capturing long-term structural dependencies (Sears et al., 2017). When encoding musical structures in multiple directions or hierarchies, the number of associations between events can explode in combinatorial as N and the length of the original sequence increase (Sears et al., 2017).

In light of this limitation, *skip*-grams have been proposed to parse non-contiguous elements from the musical structure. The maximum length of these skips can be defined by a threshold in the *fixed-skip* model. Symbols are only considered if within a fixed range of skips from the event processed. Alternatively, it can follow a *variable-skip* approach that parses all events satisfying a particular condition (Herremans and Chuan, 2017). The latter approach is typically adopted for modeling temporal-dependent sequences.

Sears et al. (2017) has shown that *skip*-grams significantly outperform contiguous *n*-grams in discovering cadences. Markov chains embed *n*-gram and *skip*-grams models to generate musical sequences that are statistically similar to modeled sources.

[1] A context-free formal grammar is a set of production rules that describe all possible strings in a given formal language by allowing the application of those rules regardless of the context of the *nonterminal* symbols.

Factor Oracle (FO) was introduced by Allauzen, Crochemore, and Raffinot (1999) as acyclic automata that recognizes at least the factors of a word. FO is a time- and memory-efficient string-matching algorithm and has been recently proposed for modeling musical structures.

FO is learned online in an incremental fashion. Repeated patterns in FO are denoted by two types of links between states: factor links and suffix links. Factor links indicate paths across states that produce similar patterns by continuing forward. Suffix links indicate paths across states that share the largest similar subsequence from the input sequence. FO is particularly useful in satisfying the incremental and fast online learning, time-bounded generation of musical sequences, and implementation of multi-attribute models to deal with the multi-dimensionality of music (Tatar and Pasquier, 2019).

Toro (2016) and Déguernel, Vincent, and Assayag (2018) extend FOs towards the introduction of link probabilities to maximize novelty when adopting the model for generative purposes. Furthermore, they promote the application of FO to multidimensional domains such as polyphonic music or improvisations with multiple musicians.

The Variable Markov Oracle (VMO) was proposed by Wang and Dubnov (2014a) for clustering multivariate time series without *a priori* assumptions on the number of clusters. VMO algorithm is based on FO (Allauzen, Crochemore, and Raffinot, 1999) and Audio Oracle (Dubnov, Assayag, and Cont, 2007). It allows the construction of the oracle without an initial alphabet. To this end, it introduces a threshold variable for computing the degree of similarity across states. The threshold value in VMO typically adopts an entropy metric to capture the information rate across events (Wang and Dubnov, 2014b).

Wang, Hsu, and Dubnov (2016) made a first attempt at establishing a statistical model for VMO by making an analogy to the HMMs based on the inference of emission probabilities, without introducing probabilities to the transitions themselves. Transition probabilities in the VMO were later proposed by Wang and Dubnov (2017), using the lengths of longest repeated suffixes, which provide variable-length Markov transition information. This new model has shown to provide a more compact and abstract representation of the oracle structure while keeping its variable-length Markov properties. Furthermore, it allows the processing of multiple works (i.e., a corpus) in a single VMO.

Hidden Markov Models (HMMs) capture relations between states that are partially hidden, i.e., unknown. Probability distributions per state define a particular alphabet symbol emission. These models are defined by a tuple of five elements that correspond to: i) the finite alphabet of visible symbols, ii) the finite set of states, iii) the mappings defining the probability of transitions between hidden states, iv) the emission probability of each visible symbol at a given hidden state, and v) the initial probabilities of the hidden states. HMMs can process complex structures of sequential data but require a considerable understanding of the problem domain and a large number of training examples (Schulze and van der Merwe, 2011).

Frankel-Goldwater (2007) implements HMM using the forward-backward, Viterbi, and Baum-Welch algorithms for modeling pitch, duration and dynamics from musical structures. Schulze and van der Merwe (2011) computes the parameters for HMMs of variable order by means of empirical counts to capture monophonic arc structures and accompaniment chord progressions.

Compression Algorithms

General-purpose lossless compression algorithms use the redundancy of input sequences to decrease storage memory size while maintaining the information in full. When applied to musical structures, these algorithms find relevant patterns and efficiently model musical structures as the "shortest descriptions of any musical object, [...] that describe the best possible explanations for the structure of that object" (Louboutin and Meredith, 2016, p. 2). In this context, we will review the following compression algorithms adopted in music structure modeling: *LZ77* and *LZ78*, the *Burrows-Wheeler Transform* and the variants in the *Structure Induction Algorithm (SIA)* family.

Ziv and Lempel's LZ77 (1977) and LZ78 (1978) are two of the most popular lossless data-compression algorithms. Both algorithms adopt a dictionary-form of the alphabet from the original musical structure to be compressed. LZ77 replaces portions of the input data with symbols representing the longest found match, in run-length encoding format, using a sliding window. The larger the window, the highest amount of recent data is acquired. The encoder can too search farthest back for creating references. LZ78 improves the performance of LZ77 by using an ordered dictionary of reusable data and its indexes, instead of the actual stream data.

The Transform of Burrows and Wheeler (1994) uses a suffix array to permutate the input data structure so that identical elements are brought closer together. It increases the probability of finding an event from an alphabet if there are near occurrences of the same event. Along with *move-to-front* coding, it builds the alphabet from the events in the structure using left to right parsing and constructs a vector of the alphabet indexes, which promotes enhanced compression factors.

The family of Structure Induction Algorithms (SIA) aims at discovering maximal repeated patterns from n-dimensional sets of points in Cartesian spaces, namely those representing musical structures (Meredith, Wiggins, and Lemström, 2002). SIA discovers all maximal subsets from a n-dimensional point set with an ordering metric and removes repetition under symmetry. SIATEC (Meredith, Wiggins, and Lemström, 2002) extends SIA by finding all occurrences of maximal repeated patterns, including those related by translational equivalence as a Translational Equivalent Class (TEC). COSIATEC ("COmpression with SIATEC") extracts the TECs resulting from SIATEC and selects those that provide "best" compression factor without overlap. RECURSIA-RRT stands for recursive translatable point-set pattern discovery with the removal of redundant translators. It optimizes the previous algorithms by increasing the compression factor.

Louboutin and Meredith (2016) compares all these algorithms on classifying folk song melodies using a multiple viewpoints system representation. LZ77 and COSIATEC were shown to achieve the best results.

Applications

In this section, we review applications that make use of the representations and modeling techniques detailed in Sections "Symbolic Representation of Musical Structures" and "Modeling Temporal Musical Structures". The non-comprehensive, yet representative, selection was based on the visibility and attention the applications attract from the music community. We adopt the following twofold categorization of the applications: computer-aided algorithmic composition (CAAC) and machine improvisation.

Computer-aided algorithmic composition

CAAC systems refer to computer applications that promote the generation of musical structures by means other than the direct manipulation the musical surface elements (Ariza, 2005a). These computational systems expand compositional design strategies towards the adoption of (semi-)automatic algorithmic techniques at different levels of the composition. A process referred to as meta-composition (Ariza, 2005b).

One of the earliest algorithmic composition examples is the *Illiac Suite* by Hiller and Isaacson (1957). It adopts rule-based systems and Markov chains to compose formal music structures. Inspired by Hiller and Isaacson's work, Baker proposed in 1963 the library *MUSICOMP*, which implements its various algorithmic composition methods (Ames, 1987).

In the early-1960s, Xenakis, renowned for his stochastic processes, uses computers to automate his composition methods (Ames, 1987). Koenig, another pioneer, implements some techniques, such as Markov chains, to automate the generation of music structures (Ames, 1987). Berg (1995) lately compiled these techniques in a collection of tools, the *AC Toolbox*, to promote various methods for algorithmic composition.

In 1981, Cope presented *EMI* (Experiments in Musical Intelligence). The system learns stylistic traits from a music corpus, manifested in the MIDI standard, to imprint them into generated musical structures (Cope, 1989). The musical information at the note level is encoded in a multidimensional point set, although the term had not yet been coined. The relation of these musical units is encoded in a formal grammar.

CACIE (Computer Aided Composition using Interactive Evolution) is an application by Daichi Ando and Hitoshi Iba (2007) that aims to assist composers in creating atonal music. It uses formal grammars to represent musical phrases from music manifested in the MIDI standard and an evolutionary (genetic) system to generate new musical structures.

FlowComposer is an interactive music composition environment developed by Papadopoulos, Roy, and Pachet (2016). It uses Markov chains to automatically compose lead sheets, which are further harmonized by style-specific traits encoded in formal string representations driven from MIDI musical corpora.

MorpheuS uses the COSIATEC pattern recognition technique to find repeated sequences in a musical piece, in MIDI format, combined with a three-dimensional geometric model (the Spiral Array) with tonal tension information from MusicXML music. Found sequences are then used to constraint generative polyphonic music processes based on evolutionary computation (Herremans and Chew, 2016).

Machine improvisation

Machine improvisation refers to musical collaborations between humans and machines in an improvisation setting. Lewis' *Voyager* (1988) is a pioneer work that expands upon the concept of "virtual improvising orchestra." It produces variations from live performers' MIDI input data and generates responses accordingly, using a rule-based, formal-grammar approach (Lewis, 2000).

The *Continuator* was proposed by Pachet (2003) as a system that "bridges the gap between interactive musical systems, limited in their ability to generate stylistically consistent material, and music imitation systems, which are fundamentally not interactive." It adopts variable-length Markov chains to model MIDI input data from a live musician encoded as a tree structure. A weighted fitness function controls the level of 'sensitivity' to the musical context.

OMax is a real-time system presented in 1998 and under active development. The system learns the style of live musicians and actively participates in an ongoing performance as a co-improviser machine. OMax uses the FO, and lately the VMO, to learn stylistic traits from a performer's MIDI stream in the form of formal string encodings.

FILTER (Freely Improvising, Learning and Transforming Evolutionary Recombination) by Nort, Oliveros, and Braasch (2013), combines FO and HMM in a context of free improvisation to learn temporal structures from the input. Moreover, it adopts a fitness function for controlling the level of imitation vs. novelty of the responses.

Summary and Future Challenges

This paper reviewed symbolic music representations using finite and discrete alphabets and temporal modeling techniques that capture musical structure hierarchies.

In Section "Symbolic Representation of Musical Structures", we detailed multiple strategies to represent musical structures, namely formal strings, formal grammars, graphs, and geometrical representations. These representations can encode complex and hierarchical music structures, yet, only a few can adequately parse inter-part (polyphonic) dependencies, such as multiple viewpoint systems. Encoding linear (i.e., part) and vertical (i.e., inter-part) dependencies should be further explored. Moreover, as noted in Section "Applications", despite the considerable number of existing representations for musical structure, their adoption by the music community focus almost exclusively on formal string representations. We believe that domain-specific representations, such as the multiple viewpoint systems, would allow for enhanced control over which structural elements are further modeled.

Methodologies for modeling the temporal structure of music at multiple hierarchies typically draw on pattern recognition methods, such as the LZ77 and LZ78 compression algorithms, and statistical modeling that capture repeating sequences in the longer-term musical structure. The compression algorithms explore exact matches, while statistical modeling, namely the Variable Order Markov Oracles, also contemplate variations (e.g., note insertions, passing notes). Despite the advances on the state-of-the-art, existing modeling techniques inadequately account for implicit, yet important, elements of musical structure, such as phrase boundaries.

In Section "Applications", we explore the combination of representations and temporal modeling in the scope of CAAC and machine improvisation. These applications are ideal test-beds for exploring the balance across the novelty-familiarity range in generated musical structures. CAAC systems typically require a wider exploration of this range, while machine improvisation tend to rather focus on balanced outputs with greater tendency towards familiarity. Yet, both use predominantly Markov chains, which neither optimize this balance nor provide fine degrees of control across the familiarity-novelty range. VMO shows finer and more flexible degrees of control over the representation and pattern recognition. However, its full capacity within generative music contexts is yet to be explored, namely the non-linear relations between the representation and modeling.

Currently, the state-of-the-art in temporal music modeling is at a crossroads. The rise of deep learning techniques – which can be explained by the increasing amount of available data, and efficient and affordable computing power – can render traditional modeling and prediction obsolete. Representative models are BachBot (Liang et al., 2017) and Music Transformer (Huang et al., 2018). However, a relevant problem of these models is that they often fail to capture the intrinsic non-linear relationships of creative tasks (Briot, Hadjeres, and Pachet, 2019). Deep learning architectures rely on multiple layers to directly extract relevant features from the sources before modeling. Their hyper-specialization towards a particular objective or a specific training corpus and their lack of explainability pose a critical problem to the creative balance. A greater understanding of latent spaces[2] is instrumental in promoting balanced outputs that match user preferences across novelty and familiarity.

[2]Latent spaces comprises the representation of compressed data in deep learning techniques.

References

Allauzen, C.; Crochemore, M.; and Raffinot, M. 1999. Factor Oracle: A New Structure for Pattern Matching. In Pavelka, J.; Tel, G.; and Bartošek, M., eds., *Proceedings of the 26th Conference on Current Trends in Theory and Practice of Informatics on Theory and Practice of Informatics*, volume 1725 of *SOFSEM '99*, 295–310. Berlin, Heidelberg: Springer-Verlag.

Ames, C. 1987. Automated Composition in Retrospect: 1956-1986. *Leonardo* 20(2):169–186.

Ariza, C. 2005a. Navigating the landscape of computer aided algorithmic composition systems: A definition, seven descriptors, and a lexicon of systems and research. In *Proceedings of International Computer Music Conference*.

Ariza, C. 2005b. *An Open Design for Computer-Aided Algorithmic Music Composition: Athenacl*. Ph.D. Dissertation, New York University, USA.

Berg, P. 1995. AC Toolbox. https://www.actoolbox.net/, Last accessed on 2020-07-11.

Berlyne, D. E. 1970. Novelty, complexity, and hedonic value. *Perception & Psychophysics* 8(5):279–286.

Bevington, J., and Knox, D. 2014. Cognitive Factors in Generative Music Systems. In *Proceedings of the 9th Audio Mostly: A Conference on Interaction With Sound*, AM '14. New York, NY, USA: Association for Computing Machinery.

Briot, J.-P.; Hadjeres, G.; and Pachet, F.-D. 2019. *Deep Learning Techniques for Music Generation*. Computational Synthesis and Creative Systems Series. Springer.

Burrows, M., and Wheeler, D. J. 1994. A block-sorting lossless data compression algorithm. Technical report.

Conklin, D., and Anagnostopoulou, C. 2001. Representation and discovery of multiple viewpoint patterns. In *International Computer Music Conference*, 479–485.

Conklin, D., and Witten, I. H. 1995. Multiple Viewpoint Systems for Music Prediction. *Journal of New Music Research* 24(1):51–73.

Cope, D. 1989. Experiments in musical intelligence (EMI): Non-linear linguistic-based composition. *Interface* 18(1-2):117–139.

Daichi Ando, and Hitoshi Iba. 2007. Interactive composition aid system by means of tree representation of musical phrase. In *2007 IEEE Congress on Evolutionary Computation*.

Déguernel, K.; Vincent, E.; and Assayag, G. 2018. Probabilistic Factor Oracles for Multidimensional Machine Improvisation. *Computer Music Journal* 42(2):52–66.

Downie, J. S. 1999. *Evaluating a simple approach to music information retrieval: conceiving melodic n-grams as text.* Ph.D. Dissertation, University of Western Ontario, Canada.

Dubnov, S.; Assayag, G.; and Cont, A. 2007. Audio oracle: A new algorithm for fast learning of audio structures. In *International Computer Music Conference, ICMC 2007*, 224–227.

Frankel-Goldwater, L. 2007. Computers Composing Music: An Artistic Utilization of Hidden Markov Models for Music Composition. *Journal of Undergraduate Research* 5(1 and 2):17–20.

Hanna, P.; Robine, M.; Ferraro, P.; and Allali, J. 2008. Improvements of alignment algorithms for polyphonic music retrieval. 244–251.

Herremans, D., and Chew, E. 2016. Morpheus: automatic music generation with recurrent pattern constraints and tension profiles. In *2016 IEEE Region 10 Conference (TENCON)*, 282–285.

Herremans, D., and Chuan, C.-H. 2017. Modeling musical context with word2vec. *ArXiv* abs/1706.09088.

Hiller, L., and Isaacson, L. 1957. Illiac Suite. Score, Theodore Presser Co, New York, USA.

Huang, C.-Z. A.; Vaswani, A.; Uszkoreit, J.; Shazeer, N.; Hawthorne, C.; Dai, A. M.; Hoffman, M. D.; and Eck, D. 2018. Music Transformer: Generating Music with Long-Term Structure. *arXiv preprint arXiv:1809.04281*.

Lemström, K., and Tarhio, J. 2003. Transposition invariant pattern matching for multi-track strings. *Nordic Journal of Computing* 10(3):185–205.

Lerdahl, F., and Jackendoff, R. 1983. *A generative theory of tonal music*. The MIT Press.

Lewis, G. 2000. Too many notes: Computers, complexity and culture in voyager. *Leonardo Music Journal* 10:33–39.

Liang, F. T.; Gotham, M.; Johnson, M.; and Shotton, J. 2017. Automatic stylistic composition of bach chorales with Deep LSTM. In *Proceedings of the 18th ISMIR Conference*.

Louboutin, C., and Meredith, D. 2016. Using general-purpose compression algorithms for music analysis. *Journal of New Music Research* 45:1–16.

Maidín, D. O. 1998. A geometrical algorithm for melodic difference. In *Computing in Musicology*, number 11, 65–72.

Marsden, A. 2001. Representing melodic patterns as networks of elaborations. *Language Resources and Evaluation* 35(1):37–54.

Meredith, D.; Wiggins, G.; and Lemström, K. 2002. Pattern induction and matching in polyphonic music and other multidimensional datasets. *Proceedings of the 5th World Multi-Conference on Systemics, Cybernetics and Informatics* 10.

Meredith, D. 2006. Point-set algorithms for pattern discovery and pattern matching in music. In Crawford, T., and Veltkamp, R. C., eds., *Content-Based Retrieval*, number 06171 in Dagstuhl Seminar Proceedings. Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

Nierhaus, G. 2009. *Algorithmic Composition: Paradigms of Automated Music Generation*. Springer Publishing Company, Incorporated, 1st edition.

Nort, D. V.; Oliveros, P.; and Braasch, J. 2013. Electro/acoustic improvisation and deeply listening machines. *Journal of New Music Research* 42(4):303–324.

Pachet, F. 2003. The continuator: Musical interaction with style. *Journal of New Music Research* 32:333–341.

Papadopoulos, A.; Roy, P.; and Pachet, F. 2016. Assisted lead sheet composition using flowcomposer. In Rueher, M., ed., *Principles and Practice of Constraint Programming*, 769–785. Cham: Springer International Publishing.

Pienimäki, A. 2002. Indexing Music Databases Using Automatic Extraction of Frequent Phrases. *3rd Int. Conf. on Music Information Retrieval* 25–30.

Rizo, D. 2010. *Symbolic Music Comparison with Tree Data Structures.* Ph.D. Dissertation, Universidad de Alicante.

Rohrmeier, M., and Pearce, M. 2018. Musical Syntax I: Theoretical Perspectives. 473–486.

Schulze, W., and van der Merwe, B. 2011. Music generation with markov models. *IEEE MultiMedia* 18(03):78–85.

Sears, D. R. W.; Arzt, A.; Frostel, H.; Sonnleitner, R.; and Widmer, G. 2017. Modeling harmony with skip-grams. *ArXiv* abs/1707.04457.

Tagliolato, P. 2008. A Generalized Graph-Spectral Approach to Melodic Modeling and Retrieval Categories and Subject Descriptors. 89–96.

Tatar, K., and Pasquier, P. 2019. Musical agents: A typology and state of the art towards musical metacreation. *Journal of New Music Research* 48(1):56–105.

Toro, M. 2016. Probabilistic extension to the concurrent constraint factor oracle model for music improvisation. *Inteligencia Artificial* 19(57):37–73.

Wang, C., and Dubnov, S. 2014a. Variable markov oracle: A novel sequential data points clustering algorithm with application to 3d gesture query-matching. In *2014 IEEE International Symposium on Multimedia*, 215–222.

Wang, C. I., and Dubnov, S. 2014b. Guided music synthesis with variable Markov Oracle. In *AAAI Workshop - Technical Report*, volume WS-14-18, 55–62.

Wang, C.-i., and Dubnov, S. 2017. Context-Aware Hidden Markov Models of Jazz Music with Variable Markov Oracle. In *Eighth International Conference on Computational Creativity (ICCC 2017)*, number October.

Wang, C. I.; Hsu, J.; and Dubnov, S. 2016. Machine improvisation with Variable Markov Oracle: Toward guided and structured improvisation. *Computers in Entertainment* 14(3):1–18.

Wishart, T., and Emmerson, S. 1996. *On Sonic Art*. Contemporary music studies. Harwood Academic Publishers.

Ziv, J., and Lempel, A. 1977. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory* 23(3):337–343.

Ziv, J., and Lempel, A. 1978. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory* 24(5):530–536.