CHAPTER

13 Artificial intelligence methods for music generation: a review and future perspectives

Maximos Kaliakatsos-Papakostas[a], **Andreas Floros**[b], **Michael N. Vrahatis**[c]

[a]*Athena Research and Innovation Centre, Institute for Language and Speech Processing, Athens, Greece*

[b]*Ionian University, Department of Audio and Vidual Arts, Corfu, Greece*

[c]*University of Patras, Department of Mathematics, Patras, Greece*

CONTENTS

13.1 Introduction

Many diverse artificial intelligence (AI) methods have been proposed for music generation over many decades. From the rule-based and Markov approaches of the Illiac Suite (Hiller and Isaacson, 1979) to more recent deep learning approaches that allow interactive piano performance tools (Donahue et al., 2018) and score filling (Huang et al., 2019b,a), researchers find it intriguing to test AI methodologies for music generation. Among the many reasons that the application of AI for generating music is interesting and important, we find the fact that music is organized on many levels of abstraction, where even complex rules may not be enough to capture deeper structures. Even in the case of the Bach chorales, which is a style of music that is highly organized music with apparently strict rules, attempts to develop generative models for the Bach chorales with rule-based approaches are efficient up to a certain level.

Nature-Inspired Computation and Swarm Intelligence. https://doi.org/10.1016/B978-0-12-819714-1.00024-5

On the other hand, there is strong research interest about if or to what extent AI methods are creative. According to Boden (2004), there are three types of creativity: (i) explorational, (ii) transformational, and (iii) combinational. Certain AI methods potentially allow the exploration of musical styles, the transformation of rules for achieving novel musical results, and the combination of conceptual spaces for forming altogether new ones. There is, however, still debate on whether the creativity of such systems, some of which can arguably be categorized to one of the aforementioned creativity categories, is a reflection of the human agent's creativity. In other words, are the methods themselves "creative" or is the engineering of generative algorithms an essential creativity component that is a prerequisite for achieving computational creativity? Evaluation methods for answering such questions have been developed, e.g., with the FACE/IDEA models (Colton et al., 2011), where not only the creative output (e.g., generated music) is examined, but also the processes (e.g., the level of intervention of the agent constructing the algorithm) are examined to determine how creative a system is. Such evaluation models are still theoretical and they would potentially have very diverse implementations in different settings, e.g., for systems that generate music from scratch, assist composers, interact with musicians, etc.

This chapter presents AI methods that have been proposed for music generation over a wide variety of algorithmic approaches, attempting to predict which of the research directions in AI methods will be more promising for music generation. The concept of abstraction is highlighted and the chapter begins in Section 13.2 with presenting an information-based approach to how musical abstraction can provide deep musical meaning through simple geometric/computational modeling. Nonadaptive methods are presented in Section 13.3, which rely on human modeling for achieving interesting results. Section 13.4 presents methods that are adaptive and learn from data, and these methods are categorized as learning explicit or implicit features. Evolutionary approaches are presented in Section 13.5, which allow for intuitive interaction with users, highlighting the importance of transparent feature modeling. Finally, Section 13.6 gathers all the positive aspects of the aforementioned method, in an attempt to present what recent advancements are more promising towards developing systems that allow intuitive user involvement in generating novel music that interpolates learned styles or even extrapolates from them.

13.2 Information, abstraction, and music cognition

Music is a stream of information that can be comprehended by humans as having structure in the form of parts with beginnings and endings, conveys feelings, and presents meaning on different levels of abstraction, while the mechanisms that elicit emotions and make music interesting to humans are related to expectation and its fulfillment or violation (Huron, 2006). Several factors come into play when it comes to how humans understand, process, and value musical elements, ranging from low-level perceptual characteristics of the human anatomy (e.g., perception of harmonics due to the cochlear shape) to higher-level cognition that relates to memory capacity and statistical learning (Huron, 2006).

The importance of statistical learning in music is supported by studies where perception of music structures in Western listeners correlates with statistical findings in corpora, e.g., tonal center and mode (Krumhansl, 2001; Temperley, 2004). Listeners who are exposed to different musical environments have different norms of expectation to such an important extent as to allow scientists to support the *cultural distance* hypothesis: "the degree to which the musics of any two cultures differ in the statistical patterns of pitch and rhythm will predict how well a person from one of the cultures can process the music of the other" (Demorest and Morrison, 2016). Statistical learning, however, is evident on higher levels of information where musical information is abstracted from the "musical surface," i.e., the layer of discrete notes, and their vertical organization in chords and melodies.

In cognitive science, research has shown that humans employ some common basic mechanisms on an ultimately abstract level for understanding and categorizing concepts in their environment; those mechanisms have been called "schemata" (Gick and Holyoak, 1983; Hedblom et al., 2016). An example of a schema is the concept of the "container," where an object acts as a container to other objects, regardless of what those objects are. In music, the idea of schemata is mainly associated with tools that create abstractions from the musical surface and facilitate the acquisition of a mental knowledge structure (Leman, 2012). Examples of such abstractions that humans unconsciously extract when exposed to musical stimuli as studied, among other works, in Leman (2012), are the concepts of tonal center and mode (Krumhansl, 2001). Those abstractions allow listeners to relate and compare musical excerpts on more abstract levels, e.g., two pieces might be similar in that they sound "happy" because they both utilize elements of a major scale similarly.

From an information perspective, it is important to note that on higher levels of abstraction, the cognitive mechanisms function under geometric principles. For instance, the tonality of a tonal piece can be accurately predicted through the correlation of its pitch class profile with the pitch class profile templates extracted from the empirical experiment conducted by Krumhansl (2001). This also means that on abstract levels of information, the perception of similarity and therefore the notion of categories in music can be approached accurately by geometric relations.

As an example that shows the powerful interpretations that are offered by geometrical information reduction techniques, a set of 35 Bach chorales is considered, obtained from the COINVENT harmonic training dataset.[1] In this datasets two steps of abstraction are performed:

1. Pitch class abstraction: Each pitch in the Bach chorales is represented by its pitch class, i.e., its value modulo 12.
2. Tonality abstraction: Each phrase in each Bach chorale, being annotated according to its tonality, is shifted to a neutral tonality, making each pitch class from the

[1] https://github.com/maximoskp/COINVENT_HTD.git.

above mentioned step a *relative* pitch class rather than an *absolute* pitch class. For instance, for a piece in C major, the note G corresponds to relative pitch class 7, while for a piece in D major, the G pitch corresponds to relative pitch class 5.

The result of this process leads to the *relative pitch class* matrix representation, denoted as R, of a musical piece; an example where only note onsets (i.e., beginning times) are considered is illustrated in Fig. 13.1.

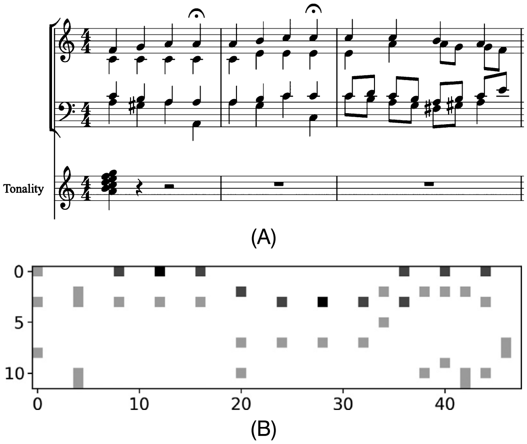

FIGURE 13.1

(A) A musical segment from a Bach chorale and its tonality annotation. (B) Illustration of its abstract relative pitch class representation in the form of a matrix, R, considering only note onsets and disregarding duration information. Colors in (B) correspond to occurrences, where darker colors represent higher values.

Each column in R represents a time instance (corresponding to a 16th of a measure). Even from the small example in Fig. 13.1, it is obvious that information can be compressed extensively and describe matrix R with its most often used components. By defining R as a matrix in $\mathbb{N}^{12\times t}$, where t is the total number of 16th time steps in all Bach chorales, we can apply nonnegative matrix factorization (NMF) (Lee and Seung, 1999) and obtain a compressed representation of R, namely, $\hat{R}$, defined as

$$\hat{R} = W\ H,\ \text{where } W \in \mathbb{R}^{12\times 3} \text{ and } H \in \mathbb{R}^{3\times t},$$

effectively reducing the dimension of R through a product of a basis vector (W) and the activation of each basis vector in time (H). The patterns that come out as basis vectors (columns of W) are shown in Fig. 13.2A, their activations (H) in (B), and the achieved reconstruction of R ($\hat{R}$) in (C).

Fig. 13.2A shows that when a geometry-based method (NMF) is used for compressing the information in three bases for the abstract representations of a set of 35 Bach chorales, each base has important musical meaning. The first column of W has large values in the locations $\{0, 3, 4, 7\}$, as indicated by the darker colors, the second in $\{0, 2, 5, 9\}$, and the third in $\{2, 7, 11\}$. The musical explanation of each column is:

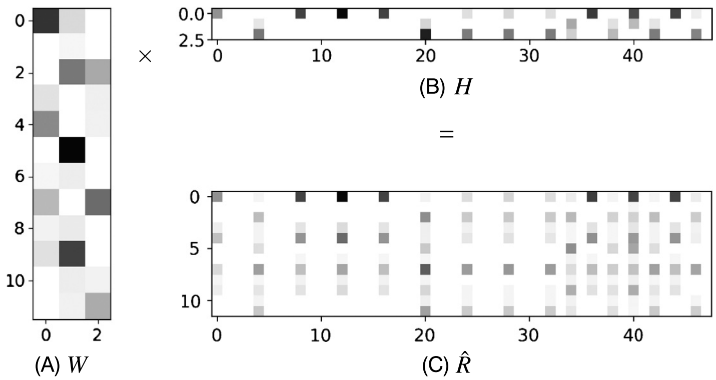

FIGURE 13.2

(A) A musical segment from a Bach chorale. (B) Illustration of its abstract relative pitch class representation in the form of a matrix, R, considering only note onsets and disregarding duration information. Colors in (A), (B) and (C) correspond to occurrences, where darker colors represent higher values.

1. $\{0, 3, 4, 7\}$ is the tonic scale degree, corresponding to either major ($\{0, 4, 7\}$) or minor ($\{0, 3, 7\}$);
2. $\{0, 2, 5, 9\}$ corresponds to the subdominant scale degree, either through IIm ($\{2, 5, 9\}$) or IV ($\{5, 9, 0\}$);
3. $\{2, 7, 11\}$ corresponds to the dominant ($\{7, 11, 2\}$).

Therefore, by simply representing the data after two steps of abstraction, a geometric method can clearly infer the "Schenkerian" basis of tonal music (W) and perform Schenkerian analysis (Cook, 1994) (H) on a set of Bach chorales.

This simple example shows the immense impact of proper abstractions from the musical surface; the NMF-inferred Schenkerian analysis simply demonstrated that there is a geometric/mathematical basis of music when considering proper abstractions. For generating music, however, abstractions on many levels and under many perspectives need to be taken into account. How can rhythmic abstractions be made? How about harmony/chords, melodies, or textural patterns (i.e., concerning voicing layouts)? The following sections, and especially Sections 13.4 and 13.5, focus on how methods decode, encode, and leverage abstractions obtained from data and/or inherited to generated musical surfaces.

13.3 Composing music with nonadaptive AI

In a "classical" AI approach, algorithms can be used that employ rules for defining the appropriateness of the output. As with Schenkerian analysis itself, musicologists have devised several abstract rules for defining musical style and for expressing what is "allowed" in specific types of compositions. These rules, however, are not sufficiently clear for producing new compositions. In fact, those rules describe very specific constraints that need to be satisfied while allowing absolute freedom on other aspects. For instance, harmonic motion (i.e., what chords are employed) and bass voice leading (i.e., how the bass voice is moving) are very strictly defined in cadences (i.e., phrase or piece endings), while they are more loosely defined in all other parts of a piece, allowing for many alternative approaches to be considered as valid for the Western tonal idiom. Examples of approaches to encoding musical knowledge of the Bach chorale style through expert-designed rules have been presented by Ebcioglu (1988) and Phon-Amnuaisuk et al. (2006), while the interested reader is referred to Pachet and Roy (2001) for a review of such methods.

Generative AI methods that are not adaptive incorporate proper mappings from the data they process to musical surfaces for music generation. Examples of such methods that are discussed in this section are *cellular automata* (CA), *L-systems*, and nonadaptive/autonomous *swarm intelligence*. The musical rules of expert systems mentioned in the previous paragraph are parts of the mapping, i.e., the numerical output of the generative algorithms was explicitly mapped to a musical entity (pitch, rhythm, intensity value, and/or information related with structure). The design and development of the mapping is what actually makes the musical output of such methods musically meaningful. In terms of the creativity reflected by such systems, the important task of the human designer/programmer in coming up with proper mappings plays an important role. Therefore, in a possible implementation of the FACE/IDEA (Colton et al., 2011) models, aiming to evaluate the creativity of such systems, the contribution of the human agent would be crucial to the creativity capabilities of the system.

Cellular automata

In CA, simple rules of interactions between neighboring units result in complex emergent behavior with structural characteristics that extend well beyond the radius of interaction between neighbors. CA can be implemented in any number of dimensions, but since their visual materialization is interesting and informative about the evolution of the CA society, one-, two-, and three-dimensional CA have been employed for music generation. The general setup of CA includes values for each unit, usually discrete or binary, that update iteratively in each "generation" according to rules that employ information about the values of neighboring units. The rules typically simulate physical phenomena as "extinction," "domination," or alternations between those extremes of units with specific values within the "universe" of the CA. The emergent behavior may have the following characteristics: (i) patterns disappear and units with a fixed value dominate; (ii) patterns repeat periodically, creating units that periodically change values; and (iii) patterns evolve chaotically/nonperiodically.

The diversity in the behavior of such systems has attracted the interest of many artists and researchers; detailed reviews of several approaches and mappings can be found in Burraston and Edmonds (2005), Burraston and Martin (2006), and Miranda and Al Biles (2007). Symbolic music mapping from CA to notes has been attempted in many contexts, with some diverse and notable examples including the piece *Horos* by Iannis Xanakis Solomos (2005), where areas of a CA universe were mapped to notes on a musical score; an application that employs similar principles for mapping to MIDI notes was presented in Millen (2005). CA have also been used for synthesizing audio through granular synthesis (Miranda, 2001), where the location of each active unit indicated which granule of sound would be active at any given time, or from given spectrograms (Serquera and Miranda, 2010) where each unit was acting as an amplitude envelope for the respective frequency area of an input spectrogram.

L-systems

L-systems generate fractal structures that resemble plants in their visual appearance (Prusinkiewicz and Lindenmayer, 2012). L-systems generate string sequences of symbols belonging to a given alphabet, based on substitution rules over an initial sequence. The typical and simplest form of L-systems belongs to the category of *deterministic context-free* grammars (DOL-systems), even though there are nondeterministic variants. As a form of formal presentation, L-systems incorporate an alphabet V of all possible symbols, a set of rules P that associate symbols in the alphabet with a string. Starting from an initial sequence of symbols $\omega \in V^+$,[2] denoted as x_0, the rules are applied for each symbol in x_0, resulting in a new sequence of symbols, denoted as x_1. Recursively, the sequence x_{n+1} is formed by applying the rules in P on each symbol in x_n. After a number of k steps, the sequence x_k will constitute a symbol sequence that potentially exhibits interesting structural characteristics on many levels, i.e., not only neighboring but also remote symbols.

As with CA, the visual interpretation of L-systems makes it evident that the higher-level structures that emerge can potentially provide a sense of structural hierarchy when properly mapped to sound/music. The first study of transforming L-systems to music via direct interpretation of generated symbols to notes was presented by Prusinkiewicz (1986); later, McCormack (1996) presented probabilistic L-systems that include probabilities about several possible rules associated with a symbol, mapping the resulting symbols to melodic notes. Further exploration of mappings between various forms of L-systems and musical notes can be found in Worth and Stepney (2005), while approaches to generating sound with the output of L-systems can be found in Manousakis (2006). A variation of the L-systems, namely, the *finite L-systems* (FL-systems), has been proposed by Kaliakatsos-Papakostas et al. (2012b), where the produced strings at each next step were truncated to a fixed length, producing strings that had quasiperiodic characteristics at different levels. Methodologies that evolve the rules of musical L-systems (de la Puente et al., 2002a; Lourenc and Brand, 2009) and FL-systems (Kaliakatsos-Papakostas et al., 2012d) through grammatical evolution have been also explored, which offer gradual alterations of the output, and adaptive behavior.

[2] V^+ is the set of nonempty words in V, i.e., nonempty symbol sequences comprising symbols of V.

Swarm intelligence

Swarm intelligence leads to the emergence of collective spatial behavior through the individual readjustment of the location of unique individuals, based on the application of simple rules that update the velocity of each agent according to the location and velocity of other neighboring agents. Variations of such emerging behavior have proved successful in optimization, through particle swarm optimization (Kennedy, 2010). Artistic expression has been examined in a specific setting of swarm intelligence that simulates the flocking behavior of real-world animals. The algorithm introduced by Reynolds (1987) has gained attention in computer and game graphics for simulating the motion of swarms, herds, and flocks (e.g., in the Batman Returns movie from 1992). This algorithm defines the motion of agent based on three components: *shoaling*, where each agent moves towards the center of mass of its neighboring agents, *collision avoidance*, where each agent moves away from the agents that are too close, and *schooling*, where the velocity of each agent gets aligned with the mean velocity of the neighboring agents.

Several parts of the aforementioned social characteristics have been embodied to interactive agents, leading to music and sound output. A swarm that was able to improvise with symbolic music output with the guidance of a human singing voice was presented by Blackwell and Bentley (2002b). The latter work was also enhanced with the addition of collision avoidance skills to the agents (Blackwell and Bentley, 2002a). Symbolic music has also been composed by agents that were specialized in certain musical tasks (Blackwell, 2003). A thorough review of these systems can be found in Blackwell (2007). Such intelligent societies have also been used for additive synthesis (Apergis et al., 2018) and granular synthesis (Blackwell and Young, 2004; Blackwell, 2008) as well as granular synthesis with spatial characteristics (Wilson, 2008). Finally, an interactive system has been proposed that receives feedback from the user to create audio and visual material using swarm intelligence and genetic algorithms (Jones, 2008). The sonification of the swarm intelligence agents behavior has been integrated into the "Swarmlake" (Kaliakatsos-Papakostas et al., 2014a) game, which expanded the social behavior with user-controlled commands and attributed different agents with different sound properties, according to specific conditions of the game.

Section summary

This section presented AI methods for music generation that are not adaptive and thus rely on human expertise in generating structured output. The creativity of such methods relies heavily on the creativity of the developer and thus the generative strengths of such models rely heavily on musical abstractions in the human agent's mind and how well the human agent can communicate those abstractions to method-related variables.

13.4 Learning from data

Trying to generate music in a specific style with nonadaptive systems is a difficult task. This is evident even for the structured style of Western tonal music, not to mention the most structured subset of this style, the Bach chorales. Regarding rule-based modeling of the tonal harmony, leaving aside voicing layout and rhythm, grammatical structures have been proposed by Rohrmeier (2011) and Koops et al. (2013), while attempts have been made to expand tonal grammars to the jazz style by Granroth-Wilding and Steedman (2014). This modeling is, however, incomplete, since it disregards all other aspects except harmony, which is by itself already complex to describe formally. In contrast to nonadaptive AI methods, methods that statistically adapt to given data have been proposed. Such methods either capture the statistical behavior of "explicitly" defined features (e.g., chord transition probabilities) or learn "implicit" representations from the musical surface into latent feature spaces. This section presents work on explicit and implicit AI modeling for music generation.

13.4.1 Explicit modeling

Probabilistic generative models can capture probabilities of occurrence of specific elements in a dataset. Regarding music, probabilities of explicitly defined features from a musical score can be captured, e.g., note or chord occurrences, note or chord transitions, and conditional probabilities of chords over given notes. Capturing such statistical information allows the development of trained models that reflect specific characteristics of the musical style in the training data. In contrast to rule-based, nonadaptive modeling, sampling from such models can potentially generate new music that reflects the characteristics of a given style.

Capturing combined and conditional probabilities of elements on a musical surface allows for the generation of new music under different settings. For methods that learn explicit representations, the review will focus on two specific test cases common to Western compositional practice, namely, *four-part harmonization* and *melodic harmonization*. In four-part harmonization, the goal is to compose a piece with a soprano, alto, tenor, and bass voice layout, where those voices are combined properly to form both concise harmonic, i.e., proper vertical positioning, and melodic streams, i.e., each voice should be a well-formed melodic part. Melodic harmonization is the composition of concise harmony over a given melody, without necessarily implementing compositions with specific voicing layout, i.e., some studies go as far as to simply assign proper chord symbols without any voicing information.

The most popular probabilistic AI techniques employed for solving such problems include hidden Markov models (HMMs) and more generalized Bayesian networks (BNs). Such models are suitable for modeling and generating music that incorporates relations between various elements, since these models allow the formulation of conditional probabilities across various aspects. For instance, the typical probability conditions for HMMs that model and generate melodic harmonizations learn two aspects of the musical surface from data: (i) chord transition (hidden state) relations through probabilities of the form $P(c_t|c_{t-1} = C)$, i.e., the probability density function of the current chord/state given the previous, and (ii) $P(c_t|m_t = \vec{M})$, i.e., the probability of the current chord given the current observed note sequence.

Regarding four-part harmonization, an approach that employed a dual HMM was proposed by Allan and Williams (2004). The role of the first HMM was to define a coarse harmonic layout given a melodic (soprano) line, modeling chords as unified symbols rather than independent voices. The second HMM was generating ornamentations, given the selected chords from the first HMM. A BN has been proposed in Suzuki et al. (2013) for four-part harmonization, which incorporated different nodes for each voice. Specifically, each voice was hierarchically conditioned on its higher voice, i.e., alto was conditioned on the soprano voice, tenor on alto, etc., while for each voice, current notes were conditioned on their previous ones. This method was able to generate the ATB voices given a melody/soprano voice. The employment of an additional node for conditioning chord symbols on the ATB voices was also examined, and the results were compared, showing the importance of the chord symbol for generating harmonically concise four-part harmonizations.

HMMs have been extensively studied for melodic harmonization, where the hidden states are chord symbols and observations are melodic notes. Microsoft has presented the then-called MySong (Simon et al., 2008) application, which allowed users to sing melodies, and after the melodic note fundamentals were extracted with digital signal processing, an HMM composed chord sequences on the given melody. The system was trained in two harmonic styles, classical and jazz. The study by Raczyński et al. (2013) examined the idea of incorporating additional information as the local tonality of the piece, therefore conditioning chord selection on tonality as well.

Among the main weaknesses of Markov-based models is their inability to capture structures on a larger timescale, since they are able to capture statistical information only to the extent that their order allows. Human-composed music incorporates meaning on many structural levels, with intermediate phrases and repetitions of large harmonic segments that cannot be captured by low-order HMMs. On the other hand, using high-order Markov models for modeling harmony leads to extremely specialized models that cannot capture style, but rather capture unaltered segments of pieces in the training dataset. Hierarchical Markov models have been proposed for capturing long-term structure (Thornton, 2009), preserving the generalization capabilities of lower-order Markov models. Those models rely on modeling repeating parts of hidden states in new hidden states, building hierarchically Markov models on top of each other for consecutively capturing patterns on different levels of time granularity. Graphical models for modeling chord progressions (Paiement et al., 2005) and melodic harmonization (Paiement et al., 2006) have been proposed, which are capable of capturing long-term relationships between chords through tree-like nodes that model conditional probabilities from top to bottom. Such methods, however, model only a fixed number of chords in a sequence – 16 chords in both aforementioned examples.

Higher-level structure in human composition is to a great extent evident in the presence *cadences*, both intermediate and final, which are parts of a piece that convey a feeling of closure, i.e., that a section closes or ends and a new section begins – in the case of a final cadence, that the entire piece ends. This fact has led many researchers to develop variations of Markov-based models that focus on generating proper cadences. For instance, Borrel-Jensen and Hjortgaard Danielsen (2010) and Yi and Goldsmith (2007) evaluated entire melodic harmonizations based on a cadence score that rated higher more typical cadential schemes of Western harmony. Other methods implemented a backwards propagation of the harmonization process (Allan and Williams, 2004; Hanlon and Ledlie, 2002), beginning from the end (cadence) to ensure that the ending part will be as concise as possible. Given the importance of cadences, a study presented by Yogev and Lerch (2008) studied the identification of possible intermediate cadence locations. If the positions of cadences are known, then Markov models with constraints (Pachet et al., 2011) could be employed, forcing the harmonization system to apply intermediate cadences at proper locations, therefore reflecting longer time structures. A first trivial approach towards this direction was presented by Kaliakatsos-Papakostas and Cambouropoulos (2014), where constraints were merely straightforwardly added in the trellis diagram, for selecting chord progressions that belong to learned cadential schemes. This method was incorporated in the CHAMELEON[3] melodic harmonization assistant (Kaliakatsos-Papakostas et al., 2017), but this method requires the user to annotate the location of intermediate cadences.

13.4.2 Implicit learning

The features to be captured by the methods in the previous paragraphs are defined "explicitly," meaning that their definition is transparent and they encompass concrete meaning; e.g., "the probability of appearance of a chord over a set of given melodic notes." Features can also be extracted "implicitly." A popular example for implicit feature computation is artificial neural networks (ANNs), which, in the case of music, process information from the musical surface and produce more abstract representations in each layer. Those representations, however, are not transparent, in the sense that there is no distinction on which aspects of the musical surface are represented at each computational unit of the ANN. Recently, deep learning has increased the attention of the research community on methods that incorporate vast amounts of computational units (i.e., neurons) organized in multiple layers, which learn implicitly from large amounts of data. Implicit learning with deep ANNs offers important possibilities for categorization and prediction, without still giving clear information about what aspects of the data are more important for taking decisions. There is significant research on alleviating this "trade-off" (lack of transparency in what the latent abstract features represent) with which those powerful methods come, towards mak-

3 http://ccm.web.auth.gr/chameleonmain.html.

ing ANNs that are "explainable," "intuitive," or "interpretable"; Section 13.6 refers to such studies.

Early approaches on using ANNs for composing music employed basic recurrent neural network (RNN) architectures, where a hidden layer was used in between input and output layer. The hidden layer, also called "states" of the network, included recurrent connections from each unit to all other units of this layer, which reinjected a weighted sum of the states values in the previous step to each unit in the states layer. The recurrent connections allow such networks to learn local dynamics of data, developing a local memory. Even from the first implementations of such networks (Todd, 1989), research was focused not only on making RNNs that reproduce music in a specific style, but also allow the network to switch styles according to an input "plan," which was actually a separate input vector to the system with the binary code that corresponded to the piece name (plan) that was currently incorporated in training. This allowed the experimentation on interpolating and extrapolating from learned melodies, by properly manipulating the "plan" part of the input.

During the early days of studying RNNs for melodic generation, the effect of psychoacoustical modeling of the inputs was examined by Mozer and Soukup (1991). According to this approach, the representation of inputs and outputs was not a simple one-hot encoding of the note currently played, but a vector of coordinates that combined pitch height (one dimension), the (x, y) coordinates on the chromatic circle, and the (x, y) coordinates on the circle of fifths. The aforementioned method exhibited the ability to learn scales, the form of interspersed random walks, and to generate melodies in the style of Bach chorales. The authors have validated, however, that such architectures were poor in capturing long-term structure of the learned melodies. To this end, the system was improved by incorporating a blurred "bird's eye" view as an additional input, for getting information from further back in the past (Mozer, 1994). Even though the results were better, significant improvements on adaptation to long-term structure was exhibited by using the long short-term memory (LSTM) networks proposed by Eck and Schmidhuber (2002). Those networks include trainable gates that selectively forget information from the past or recall information from arbitrarily back in time. The first study on how those networks learned on musical data (Eck and Schmidhuber, 2002) showed that they are capable of learning long time dependencies that allowed them to learn and generate structures belonging to the style of 12-bar blues.

A more recent approach to using LSTM RNNs for generating melodies was presented by Sturm et al. (2015, 2016), where folk tunes (monophonic melodies) were modeled in the ABC format, a text and character-based representation that includes metadata, overview of musical setup (e.g., tempo and time signature), metric information (i.e., measure boundaries), and the music surface. In contrast to the directly numerical format of musical data representation, elements of characters and strings corresponding to elements of a melody were extracted into a one-hot dictionary representation (binary array with a single unit).

Many alternatives have been proposed in representing polyphonic music for efficient processing by ANNs. Many studies have examined the efficiency of proposed representations and how ANNs process them by learning and generating pieces in the style of Bach chorales, since this style of music has a very strictly defined form. A "quasimonophonic" approach to modeling polyphonic data was presented by Liang et al. (2017), where note symbols, bar limits, and fermata symbols are learned and generated sequentially, from top to bottom and from left to right. A similar representation approach was followed by Colombo et al. (2018) with more refined information about the duration and offset time of notes. In these approaches, the ANN was fed with a sequence of single notes, where simultaneous notes simply had the same onset (beginning) time. Both aforementioned studies incorporated learning and generating polyphonic music in the style of Bach chorales; in addition to the partly different representations they used, another difference was that Liang et al. (2017) used LSTM units while Colombo et al. (2018) used gated recurrent units (GRUs) for polyphonic symbolic music generation. GRUs, like LSTMs, include a gating mechanism, but only for selectively resetting and updating the content of information in the recursive connections. The GRU architecture is simpler than the LSMT architecture, thus GRUs are less computationally expensive, while at the same time being approximately equally efficient to the LSTMs (Chung et al., 2014).

An important concern in generative RNNs is not only to enable them to capture longer-time dependencies, which LSTMs and GRUs achieve quite efficiently, but also enable them to capture different modes of structures for long-time relations, e.g., to compose a piece in 4/4 or 7/8 time signature. To this end, constraints have been introduced in Hadjeres et al. (2017) which allow the networks representing each voice to have a more robust understanding about the overall metric structure and the activity in each voice. Typical bidirectional LSTM layers in the architecture were responsible for learning the motion in single voices, while other parts of the network were imposing constraints for the metric structure, allowing the network to learn to generate four-part harmonizations in specific time signatures defined in the input. Additionally, the network was generating music through sampling and therefore any voice could have any set of notes fixed as a priori constraints, allowing the network to fill in the remaining notes for completing a composition. A similar approach to imposing constraints was presented for drum rhythm generation by Makris et al. (2017, 2019). In the latter studies, indications were given that proper representation of the metric constraints could allow the network to compose rhythms in time signatures that were not encountered during training. For instance, the network was trained in pieces in 4/4 and 7/8 time signatures of a given style and could compose consistent rhythms in 5/4, 9/8 and 17/16 that were compatible with such rhythms in the learned style.

Except for RNNs, other types of networks have been explored for generating music. Convolutional neural networks (CNNs) are able to capture patterns in data through the employment of filters that adapt to specific regularities that appear often. In a study by Yang et al. (2017) a generative adversarial neural network (GAN) setting was presented that employed a generator and a discriminator based on CNNs for generating monophonic melodies. In GANs there are two networks "competing" with each other: the generator produces data that the discriminator tries to identify as "artificial" in comparison with given ground-truth data. The generator, therefore, gradually learns to generate data that more persuasively appear as being part of a given dataset, while the discriminator gradually becomes more sensitive in identifying artificial data produced by the generator, leading to a feedback loop that makes both parts of the network more effective in their task. The interesting aspect of GANs is that the generator is not necessarily straightforwardly trained from data, since it can start from generating random material with initially randomized internal parameters, that are gradually refined during training. Additionally, a lead-sheet music setting was employed where chord symbols were given and the network learned to generate musical surface that corresponded to a rhythmic and a melodic part (Liu et al., 2018). The aforementioned methodology has also expanded to incorporate multiple tracks (Dong et al., 2018). More studies on ANNs are discussed in Section 13.6, along with their potential to offer new possibilities.

Section summary

This section presented models for learning music, based on adaptation to training data. Explicit learning methods were first analyzed that have the advantage of being "transparent," i.e., it is clear what features the network learns. Next, implicit learning methods were discussed, which create abstractions from data that are not transparent, but can describe deep structures on many levels of information. Especially regarding deep implicit learning methods, some studies were discussed that allowed some form of control over the generated output, e.g., by defining the key signature. More work on ANNs is mentioned in Section 13.6, when discussing future perspectives of AI in generative music.

13.5 Evolutionary approaches

Evolutionary algorithms evolve generations of individuals by selecting and breeding individuals based on their fitness value, which interprets numerically some criteria for the goal to be achieved. Each individual is represented by a genotype (the genetic material that can be modified during the breeding stage) and a phenotype (the materialization of the genotype); the genotypes and phenotypes of individuals may coincide, depending on the formulation of the problem. Evolutionary algorithms have been studied for music generation under various setups regarding how the musical surface is represented (in terms of phenotype and genotype) and how a good musical surface should be formally described, i.e., what the fitness criteria should be. Especially regarding the fitness criteria, cognitive-based features extracted from data play an important role, making abstraction a necessary step towards assessing the fitness of musical individuals during evolution. This section focuses on two music generation approaches where evolutionary algorithms have given interesting results: feature-based composition and interactive composition.

13.5.1 Feature-based composition

Humans create abstractions from the musical surface that allow them to do content categorization and measure similarity (Cambouropoulos, 2001). As mentioned in Section 13.2, an example of this process is the categorization of a musical piece in the category of tonal music: if a piece employs some standard tonal harmonic devices, e.g., diatonic scales and cadences that resolve from highly dissonant to highly consonant harmonies (Huron, 2006), then this piece is included in the tonal category. Research has shown that such high-level features, e.g., how strong the presence of a diatonic scale in a musical excerpt is, can be directly computed from the musical surface, as with the pitch class profile of the excerpt, and can accurately predict the diatonicity of an excerpt based on the templates extracted by Krumhansl (2001). Numerous such examples have been shown in the literature, where features computed from the musical surface can indicate qualitative aspects of the data. Another example concerns the perception of rhythm, where empirical studies presented by Madison and Sioros (2014) and Sioros et al. (2014) have shown that there are strong correlations between the feature of syncopation and the sensation of groove in rhythms.

Widely used methodologies and software have been proposed and developed extracting symbolic music features (Eerola and Toiviainen, 2004; McKay and Fujinaga, 2006). A fact that makes the efficiency of such features more evident is that they are producing accurate results in various content categorization tasks, such as composer identification (Purwins et al., 2004; Wolkowicz et al., 2008; Kaliakatsos-Papakostas et al., 2010, 2011) and the style and genre classification (Kranenburg and Backer, 2004; Mckay and Fujinaga, 2004; Hillewaere et al., 2009b; Hillewaere and M, 2009a; Herremans et al., 2015a; Zheng et al., 2017). Furthermore, features that generate information-theoretic abstractions of data, e.g., Shannon information entropy or fractal dimension, have been studied for the characterization of "esthetic" quality in music, leading to models that examine relations between complexity and human perception in music (Shmulevich et al., 2001; Madsen and Widmer, 2007) and also to models of subjective preference (Manaris et al., 2002; Machado et al., 2003; Manaris et al., 2005; Hughes and Manaris, 2012).

On the one hand, such features can indicate the category, mood, or complexity of composed pieces. On the other hand, evolutionary methods can be used to generate novel excerpts that belong to a certain category, mood, or complexity, given proper fitness functions and representations of musical surfaces – mappings from "genotypes" to "phenotypes." The feature extraction methods discussed above are again "explicit," in the sense that their computation from the musical surface is transparent. Such features can be used in evolutionary generative methods during fitness evaluation to examine whether the generated material meets the criteria set from those features. Even from the early days of generative music systems, there were some exceptional studies on the evolutionary generation of melodies that employed "implicit" feature extraction methods, implemented with ANNs. The work of Spector and Alpern (1995) and the work of Pearce (2000) are such examples of using fitness functions in evolutionary methods for music generation that are based on implicit learning. Those so-called "artificial critics" are trained to give positive feedback to collected melodies that incorporate desired characteristics (e.g., Charlie Parker solos) and negative feedback to either random or empty melodies. They are therefore capable of providing "implicitly computed" fitness evaluation to melodies generated through evolutionary processes.

Other than the aforementioned approaches that employed implicit fitness evaluation through ANN critics, fitness evaluation by means of distance from targeted/desired explicit features has been the most usual approach with evolutionary music generation methods. Such methodologies incorporate a set of target features that either incorporate information-related target features or features related to music cognition and theory; the evolutionary component of those methods generates music that adapts to the target features as generations progress. Regarding information-related metrics, features that compute the fractal dimension in distributions of several diverse elements obtained from the musical surface (e.g., pitch, interval, or rhythm-related distributions) were presented in Manaris et al. (2007); similarly, the normalized compression distance in Alfonseca et al. (2007) was employed for generating music with specific complexity characteristics. Cognitive and music-theoretic target features have been developed in other studies that quantified approaches to describe rules for counterpoint (Herremans and Sörensen, 2012), four-part harmonizations theory (Donnelly and Sheppard, 2011; Phon-Amnuaisuk and Wiggins, 1999), and melodic harmonization (Phon-Amnuaisuk et al., 2006), or quantities related to how humans perceive and process music (Wiggins and Papadopoulos, 1998; Özcan and Ercal, 2008; Matic, 2010; Hofmann, 2015; Herremans et al., 2015b). It should be noted that some approaches employed ANNs as "artificial critics" (Manaris et al., 2007; Machado et al., 2003), but therein, ANNs actually evaluated the similarity of the explicitly defined features related with fractal dimension and transition probabilities between the generated and a set of training data. The neural networks in this context receive many such features as input and their goal is to create abstract/latent representations of these features (instead of the musical surface, as discussed in the previous paragraph).

Another point of distinction between methods that have been employed for music generation is the genotypical and phenotypical representation. The methods mentioned so far evolve individuals that directly represent musical surface, i.e., the genotype comprises representations of notes. Other methodologies attempt to leverage the structural coherence that nonadaptive AI methods (discussed in Section 13.3) present. Examples of such methods include grammatical evolution (de la Puente et al., 2002b) and genetic evolution of CA rules (Lo, 2012) and of FL-systems (Kaliakatsos-Papakostas et al., 2012d). Additionally, the evolution of parameters of dynamical systems that present chaotic behavior was examined in Kaliakatsos-Papakostas et al. (2013a), where the parameters were tuned using differential evolution (Price et al., 2006).

13.5.2 Interactive composition

Evolutionary processes offer ways for human users to affect the generative process in different ways. In evolutionary schemes that employ interactive evolution, fitness evaluation is given directly by the human user and, therefore, the results are directly guided by the human agent. Due to the interpretable nature of explicitly defined features, it is possible for evolutionary processes to involve the human user in processes that "indirectly" affect the outcome. In this chapter two such methods are identified: "dissimilarity-based interaction" and "active musical interaction"; all approaches are discussed in the remainder of this section.

"Interactive evolution" comprises methods that require human evaluation for assigning fitness values for the evolved individuals in the population in the form of rating, ranking, or selection. Several studies have focused on music generation through interactive evolution with individual selection based on ratings (Unehara et al., 2005; Fortier and Van Dyne, 2011; Kaliakatsos-Papakostas et al., 2012a; MacCallum et al., 2012) or direct selection for reproduction (Sánchez et al., 2007). Rhythm generation (Horowitz, 1994; Johanson and Poli, 1998) was the first field of application and subsequently more musical aspects were included, where interactive evaluation incorporated partial rating of different aspects of music, e.g., rhythm, tonality, and style (Fortier and Van Dyne, 2011; Moroni et al., 2000). Such methodologies for music generation have the theoretical advantage that fitness evaluation is absolutely adaptive to the human user and that esthetic convergence is possible, given sufficient time; however, additional problems are practically introduced in comparison with methods that are noninteractive. The main problem of interactive evolution methods for music generation is the practical infeasibility to combine and alter large numbers of individuals within the course of many generations. Human users are not able to undergo vast amounts of listening and rating (or selecting) sessions, since *user fatigue* occurs during the very few first minutes of rating/selective sessions. As an even more negative result, the uncertainty in user ratings or selections is also increased, leading to inconsistent ratings that eventually "detune" the evolutionary effectiveness. Early approaches (Tokui, 2000) involved intermediate steps in between generations, where many individuals were generated and only a small part of them were shown to the user, based on an intermediate evaluation offered by an ANN.

"Dissimilarity-based interaction" accepts a user-given musical segment and a user-defined dissimilarity value; genetically modified musical segments are then evolved towards generating segments that are desirably dissimilar to the user-given segment. The dissimilarity value is computed according to some features that offer a layer of human–machine communications, where information is interpretable both by human and machine. Some studies have focused on generating novel rhythms based on an input rhythm provided by the user and a value of dissimilarity for the new rhythms (Kaliakatsos-Papakostas et al., 2013b; Nuanáin et al., 2015; Vogl et al., 2016).

During "active musical interaction," the user generates musical objects and expects relevant musical responses from the system. Therefore, the human performance affects the AI performance and vice versa, leading to a "creative loop" between the human and the artificial agent. The first approaches that employed evolutionary algorithms for active musical interaction were presented by Biles (2002), Thom (2001), and, more recently, Manaris et al. (2011), where the human and the artificial agents were exchanging phrases, i.e., the artificial agent needed to record and encode the human phrase, analyze its features, and playback a proper response; such an implementation was presented by Weinberg et al. (2008), but the responses of the artificial agent were performed by a hardware robotic musician. In Kaliakatsos-Papakostas et al. (2012c), another approach was presented, which incorporated concurrent performance from the human and the artificial agent in the form of intelligent accompaniment. Therein, however, the system was able only to identify the current playing status of the human musician, failing to predict possible structures and therefore leading to “constraint-free” improvisation.

Section summary

Evolutionary computation methods for music generation have been mainly studied as methods that evolve musical individuals to capture explicitly defined feature. Having interpretable/explicit features, on the one hand, allows such methods to model and reproduce specific aspects of musical styles and, on the other hand, allows interactive applications, where human and artificial agents interact towards formulating a result that is interesting to the user. Even though there are inherent limitations in interactive evolutionary systems (i.e., user fatigue), other modes of interaction (dissimilarity-based and active musical interaction) potentially allow for better human–machine collaboration results.

13.6 Towards intuitive musical extrapolation with AI

Section 13.4 discussed the importance of capturing deep structures in music implicitly, without the necessity to employ human expertise for describing all the necessary information for representing an entire musical style. Section 13.5 presented evolutionary algorithms, which are based on feature design for capturing desired characteristics of generated pieces. Describing style can be achieved either by explicit or by implicit feature extraction methods. The questions that this paragraph tries to answer are the following:

1. How is it possible to cross the borders of musical style?
2. How can methods provide an intuitive, interpretable layer, for how stylistic crossing is perceived?

Answers to these questions are sought by examining recent work in three generative AI methods: (i) evolutionary computation, (ii) conceptual blending, and (iii) deep learning.

Evolutionary computation

Regarding evolutionary algorithms, an approach to interactively extrapolating and crossing stylistic boarders in a consistent way was presented by Kaliakatsos-Papakostas et al. (2016). In this approach, a human user was listening to quadruples of generated polyphonic melodies according to rhythmic and pitch characteristics.

Based on those ratings, two evolutionary processes were utilized: an "upper-level" evolution of rhythmic and pitch features using the particle swarm optimization (PSO) algorithm (Kennedy, 2010) and a "lower-level" evolution of melodies with genetic algorithms with fitness evaluation targeting the features generated on the upper level. The goal of the upper level is to converge to rhythm and pitch features that the user prefers, while the goal of the lower level is to materialize those features to actual musical excerpts.

Even though this study incorporates some assumptions that need to be examined more thoroughly, it nonetheless offers a possible way to exploring generatively new musical areas by traversing potentially unforeseen areas of musical feature spaces. The cognitive advantage of this study, in comparison with studies that simply apply interactive evolution on musical excepts, is that evolution and user ratings concern the cognitively informed layer of musical features rather than the musical surface. Even though human evaluation of this method has not been implemented, the idea behind it is that evolving features towards directions given by the PSO algorithm encompasses a cognitive coherence, making the evolutionary process more meaningful. Contrarily, traditional methods for evolving musical excerpts by mere mutation and crossover of their parts do not guarantee to generate new excerpts that also combine high-level features.

Conceptual blending

The conceptual blending (CB) theory (Fauconnier and Turner, 2003) describes the cognitive processes that humans undergo when generating new concepts, based on the experiences of already known conceptual spaces. In CB theory, two input conceptual spaces are considered, which incorporate properties and relations between elements, and a blended space is generated by consistently combining properties and elements of the inputs. Initially, CB theory was used as a theoretic tool for interpreting creative artifacts created by humans, i.e., the blended space of a creative outcome was considered (e.g., a musical piece) and the task was to identify the constituent parts as independent input spaces (e.g., the musical/conceptual tool combined by the artist). Algorithmic approaches that use CB theory generatively have more recently been developed, i.e., two input spaces are given and a blended space is algorithmically constructed that consistently and creatively combines properties and relations in the input. This approach to computational creativity is related with *combinational* creativity, which Boden (2004) maintains is the most difficult to describe formally.

In music, generative formulations of generative CB have produced interesting results. In Eppe et al. (2015), it was shown that proper encoding of conceptual spaces describing cadences (defined as the last pair of chords in a chord sequence) can lead to the generation of interesting cadences. An interesting example presented therein was the algorithmic construction of the tritone substitution cadence, which is omnipresent in jazz music after the 20th century, by using two input cadences that were employed in music centuries earlier, namely, the perfect and the Phrygian cadences. The algorithmic materialization of this example agrees with music-theoretic perspectives that indeed relate the characteristics of the tritone substitution with the most salient characteristics of the perfect and the Phrygian cadences. Describing the characteristics of cadences and weighing them by importance (in order to be included in blends that are rated as more successful) is, however, a task that requires extensive musical expertise. Therefore, the algorithmic backbone of generative CB by itself is not sufficient for describing what are the most important characteristics of the inputs that should be included in the blends; depending on the domain of application, extensive human expertise and knowledge engineering are required for acquiring meaningful results. There have been attempts for automating the process of computing the salience/importance of features in the inputs through statistical approaches (Kaliakatsos-Papakostas and Cambouropoulos, 2019), but further examination is necessary before verifying that salience computation can be achieved effectively directly from data.

In practical terms, the employment of generative CB can prove useful for developing systems that generate entire melodic harmonizations. The CHAMELEON melodic harmonization assistant (Kaliakatsos-Papakostas et al., 2017) is such an example. This system expands on the ideas developed for the example of cadence blending for blending chord progressions between the Markov transition tables of two learned musical styles and generates blended transition matrices of two learned idioms. The blended styles integrate the most salient, in terms of statistical frequency, characteristics of the inputs, a fact that leads to results that not only interpolate between the two input styles but also extrapolate from them. An evaluation study with students in a music department, who were well aware of tonal and jazz music, indicated that they would categorize blended harmonizations either as "in between" tonal and jazz, or oftentimes as belonging to altogether "other" styles (Zacharakis et al., 2018).

Deep learning

Even though both evolutionary and blending approaches potentially offer the mechanisms to cross stylistic boundaries and interpolate between or even extrapolate from known styles, they require explicit knowledge description and extensive knowledge engineering. These requirements are not by themselves necessarily a drawback; there is extensive, however, continuous extensive research on how to allow computational methods do by themselves implicit feature extraction that is adaptive to the style of the training data. As discussed in Section 13.4.2, implicit learning methods do not require extensive human knowledge, which is very time consuming, tedious, prone to errors, and nonadaptive, in the sense that not all styles behave under comparable statistical rules. For instance, the polyphonic songs of Epirus present a "horizontal" rather than "vertical" interpretation of harmony, i.e., streams of voices move rather independently from each other, forming hard dissonances that cannot be accurately captured by the consonant organization of Western harmony (Kaliakatsos-Papakostas et al., 2014b). Therefore, modeling polyphonic songs from Epirus with statistical learning (e.g., through Markovian processes) on explicitly defined chord structures is problematic in generating persuasive results.

The implicit learning methods discussed in Section 13.4.2 can effectively learn latent representations of features directly from musical surfaces, alleviating the necessity for extensive human engineering. Such methods are therefore effective in learning a style though deep representations; are they, however, capable of meaningfully interpolating between or extrapolating from learned styles? Examples in image generation have shown that leveraging the spatially interpretational capabilities of the latent space in the variational autoencoder (VAE) (Kingma and Welling, 2013) can lead to generating new images that combine characteristics of existing images (Gulrajani et al., 2016). This is possible due to the sampling process that occurs during training, in combination with the fact that the latent space is trained not only to reconstruct faithful representations of the input data (e.g., of a given image) but also to follow a Gaussian distribution. Sampling from (initially "detuned") Gaussian distributions at each training epoch leads to latent representations that interpret meaningful information throughout the entire extent of the Gaussian distribution – given enough data.

In music, interesting results have been presented with the utilization of VAE for music generation (Roberts et al., 2018). Learning the latent space of various musical excerpts allowed the system to both interpolate and extrapolate from two given excerpts in a meaningful way. For instance, if two melodies were given that were primarily different in one feature, e.g., one was polyphonic and the other monophonic, sampling from interpolated points between the latent representations of the inputs generated new melodies with intermediate characteristics. Specifically, sampling from latent points that were closer to, e.g., the polyphonic excerpt generated more polyphonic output than sampling closer to the monophonic end. This creates a "morphing continuum" between any two points in the latent space, which is musically meaningful, in the sense that the different features that are implicitly captured in the latent space are consistently mapped to the musical output, creating excerpts that morph between two extremes. Furthermore, extrapolating from the line that connects the latent representations of two input excerpts would generate excerpts that "exaggerate" the feature differences towards the extrapolation end. For instance, in the example of the polyphonic and monophonic input excerpts, extrapolating towards the end of the polyphonic excerpt would produce an excerpt that has even more polyphony than the input excerpt in the polyphonic extreme.

Section summary and discussion

This section discussed methods that potentially allow music generation that intuitively and meaningfully interpolates and extrapolates from learned styles. Such methods were presented that either follow explicit approaches to representing features or implicit learning from data. The intuitiveness offered by methods that employ explicit feature representations comes at the cost of extensive human design, which is not only inaccurate and tedious but also style-specific and error-prone.

Regarding implicit learning, VAEs constitute a promising example of how machines can learn interconnected and meaningful latent representations that not only homogeneously connect areas that represent samples of the training data, but are also able to infer connections with latent points that represent data beyond the borders of what has been learned. The typical VAE, even though effective, incorporates a latent space that has “entangled” representations of features, i.e., a single feature (e.g., polyphony) might be expressed by more than one latent variable with complex relations. A modification of the typical VAE has been proposed recently, namely, the *beta*-VAE (Higgins et al., 2017), which generates disentangled representations of the latent features, in the sense that unique features are represented by the fewest possible latent variables. For example, the polyphony feature can be represented only by a single latent feature; therefore, changing the value of this feature could simply change the polyphony of an excerpt without necessarily having to provide a second excerpt for computing the interpolation/extrapolation direction in the latent space for achieving the desired result in polyphony.

13.7 Conclusion

This chapter has presented a review on AI methods for music generation. Initially, the important role of abstraction in music was highlighted, giving an example with the application of NMF on abstracted information obtained from the music surface of a set of Bach chorales, leading to an information-based extraction of the Schenkerian analysis. Subsequently, AI methods for music generation were presented in three categories, according to how they implement abstraction from the musical surface: (i) nonadaptive, (i) probabilistic, and (iii) evolutionary methods. The role of a human design was highlighted in nonadaptive systems, where abstractions are taking the form of rules designed by the artist/researcher who creates the method. Probabilistic methods, in contrast to nonadaptive that employ fixed rules, adapt to data by learning statistical relations between musical elements. In probabilistic methods, two categories were defined according to what knowledge they acquire: explicit modeling of knowledge involves learning specific features/abstractions defined in a way that is computationally transparent to the human user, and implicit modeling allows the model itself to create proper abstractions and learn their statistical behavior. Finally, evolutionary methods were analyzed, which are based on explicit modeling but also allow interactive interventions by users, based on the intuitive modeling they offer.

The chapter then focused on work that will expectedly open new possibilities for the involvement of AI in music generation. Specifically, methods based on evolutionary computation, conceptual blending, and deep learning are analyzed, based on the fact that they attempt to offer genuinely new/creative results that cross stylistic borders, allowing interpolations between and extrapolations from musical styles. Especially for deep learning methods, it seems possible that recent advantages will allow intuitive interpretations on what they produce, allowing them to be employed in more interactive and useful settings in real-world applications. Those developments appear to be significant since deep learning methods learn by themselves abstractions that cover multiple aspects of the learned styles, covering deep relations that a human designer might fail to properly describe with explicit feature design.

References

Alfonseca, M., Cebrián, M., Ortega, A., 2007. A simple genetic algorithm for music generation by means of algorithmic information theory. In: IEEE Congress on Evolutionary Computation. 2007, CEC 2007, pp. 3035–3042.

Allan, M., Williams, C.K.I., 2004. Harmonising chorales by probabilistic inference. In: Advances in Neural Information Processing Systems, vol. 17. MIT Press, pp. 25–32.

Apergis, A., Floros, A., Kaliakatsos-Papakostas, M., 2018. Sonoids: interactive sound synthesis driven by emergent social behaviour in the sonic domain. In: Proceeding of the 15th Sound and Music Computing Conference (SMC). SMC 2018.

Biles, J.A., 2002. GenJam: evolution of a jazz improviser. In: Bentley, P.J., Corne, D.W. (Eds.), Creative Evolutionary Systems. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 165–187.

Blackwell, T.M., 2003. Swarm music: improvised music with multi-swarms. In: Proceedings of the 2003 AISB Symposium on Artificial Intelligence and Creativity in Arts and Science, pp. 41–49.

Blackwell, T.M., 2007. Swarming and music. In: Miranda, E.R., Biles, J.A. (Eds.), Evolutionary Computer Music. Springer, London, pp. 194–217.

Blackwell, T., 2008. Swarm granulation. In: Romero, J., Machado, P. (Eds.), The Art of Artificial Evolution. In: Natural Computing Series. Springer, Berlin, Heidelberg, pp. 103–122.

Blackwell, T.M., Bentley, P., 2002a. Don't push me! Collision-avoiding swarms. In: Proceedings of the 2002 Congress on Evolutionary Computation. 2002, CEC '02, vol. 2. IEEE, pp. 1691–1696.

Blackwell, T.M., Bentley, P., 2002b. Improvised music with swarms. In: Proceedings of the 2002 Congress on Evolutionary Computation. 2002, CEC '02, vol. 2. IEEE, pp. 1462–1467.

Blackwell, T., Young, M., 2004. Swarm granulator. In: Applications of Evolutionary Computing, Proceedings of the 2nd European Workshop on Evolutionary Music and Art. EVOMUSART 2004. Springer, Berlin, Heidelberg, pp. 399–408.

Boden, M.A., 2004. The Creative Mind: Myths and Mechanisms. Routledge.

Borrel-Jensen, N., Hjortgaard Danielsen, A., 2010. Computer-assisted music composition – a database-backed algorithmic composition system. B.S. Thesis. Department of Computer Science, University of Copenhagen, Copenhagen, Denmark.

Burraston, D., Edmonds, E., 2005. Cellular automata in generative electronic music and sonic art: a historical and technical review. Digital Creativity 16 (3), 165–185.

Burraston, D., Martin, A., 2006. Wild nature and the digital life. In: Special Issue. Leonardo Electronic Almanac 14 (7-8).

Cambouropoulos, E., 2001. Melodic cue abstraction, similarity, and category formation: a formal model. Music Perception: An Interdisciplinary Journal 18 (3), 347–370.

Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. Preprint. arXiv:1412.3555.

Colombo, F., Brea, J., Gerstner, W., 2018. Learning to generate music with BachProp. Preprint. arXiv:1812.06669.

Colton, S., Charnley, J.W., Pease, A., 2011. Computational creativity theory: the face and idea descriptive models. In: ICCC, pp. 90–95.

Cook, N., 1994. A Guide to Musical Analysis. Oxford University Press, USA.

de la Puente, A.O., Alfonso, R.S., Moreno, M.A., 2002a. Automatic composition of music by means of grammatical evolution. In: Proceedings of the 2002 Conference on APL: Array Processing Languages: Lore, Problems, and Applications. APL '02, pp. 148–155.

de la Puente, A.O., Alfonso, R.S., Moreno, M.A., 2002b. Automatic composition of music by means of grammatical evolution. In: ACM SIGAPL APL Quote Quad, vol. 32(4), pp. 148–155.

Demorest, S.M., Morrison, S.J., 2016. Quantifying culture: the cultural distance hypothesis of melodic expectancy. In: The Oxford Handbook of Cultural Neuroscience, p. 183.

Donahue, C., Simon, I., Dieleman, S., 2018. Piano genie. Preprint. arXiv:1810.05246.

Dong, H.W., Hsiao, W.Y., Yang, L.C., Yang, Y.H., 2018. Musegan: multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In: Thirty-Second AAAI Conference on Artificial Intelligence.

Donnelly, P., Sheppard, J., 2011. Evolving four-part harmony using genetic algorithms. In: European Conference on the Applications of Evolutionary Computation, pp. 273–282.
Ebcioglu, K., 1988. An expert system for harmonizing four-part chorales. Computer Music Journal 12 (3), 43–51. ISSN 01489267.
Eck, D., Schmidhuber, J., 2002. Learning the long-term structure of the blues. In: International Conference on Artificial Neural Networks, pp. 284–289.
Eerola, T., Toiviainen, P., 2004. Midi toolbox: Matlab tools for music research.
Eppe, M., Confalonier, R., Maclean, E., Kaliakatsos-Papakostas, M., Cambouropoulos, E., Schorlemmer, M., Codescu, M., Kühnberger, K.U., 2015. Computational invention of cadences and chord progressions by conceptual chord-blending. In: International Joint Conference on Artificial Intelligence (IJCAI). 2015.
Fauconnier, G., Turner, M., 2003. The Way We Think: Conceptual Blending and the Mind's Hidden Complexities, reprint edition ed. Basic Books, New York.
Fortier, N., Van Dyne, M., 2011. A genetic algorithm approach to improve automated music composition. International Journal of Computers 5 (4), 525–532.
Gick, M.L., Holyoak, K.J., 1983. Schema induction and analogical transfer. Cognitive Psychology 15 (1), 1–38.
Granroth-Wilding, M., Steedman, M., 2014. A robust parser-interpreter for jazz chord sequences. Journal of New Music Research 43 (4), 355–374.
Gulrajani, I., Kumar, K., Ahmed, F., Taiga, A.A., Visin, F., Vazquez, D., Courville, A., 2016. Pixelvae: a latent variable model for natural images. Preprint. arXiv:1611.05013.
Hadjeres, G., Pachet, F., Nielsen, F., 2017. DeepBach: a steerable model for Bach chorales generation. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1362–1371.
Hanlon, M., Ledlie, T., 2002. CPU Bach: an automatic chorale harmonization system. http://www.timledlie.org/cs/CPUBach.pdf.
Hedblom, M.M., Kutz, O., Neuhaus, F., 2016. Image schemas in computational conceptual blending. Cognitive Systems Research 39, 42–57.
Herremans, D., Sörensen, K., 2012. Composing first species counterpoint with a variable neighbourhood search algorithm. Journal of Mathematics and the Arts 6 (4), 169–189.
Herremans, D., Sörensen, K., Martens, D., 2015a. Classification and generation of composer-specific music using global feature models and variable neighborhood search. Computer Music Journal 39 (3), 71–91.
Herremans, D., Weisser, S., Sörensen, K., Conklin, D., 2015b. Generating structured music for bagana using quality metrics based on Markov models. Expert Systems with Applications 42 (21), 7424–7435.
Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A., 2017. beta-VAE: learning basic visual concepts with a constrained variational framework. In: International Conference on Learning Representations, vol. 2(5), p. 6.
Hiller, L.A., Isaacson, L.M., 1979. Experimental Music; Composition with an Electronic Computer. Greenwood Publishing Group Inc.
Hillewaere, R., Manderick, B., Conklin, D., 2009a. Global feature versus event models for folk song classification. In: Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009). Kobe, Japan, 26–30 October, pp. 729–733.
Hillewaere, R., Manderick, B., Conklin, D., 2009b. Melodic models for polyphonic music classification. In: Proceedings of the 2nd International Workshop on Machine Learning and Music (MML 2009). Held in Conjunction with The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2009), Bled, Slovenia, 7 September, pp. 19–24.
Hofmann, D.M., 2015. A genetic programming approach to generating musical compositions. In: Evolutionary and Biologically Inspired Music, Sound, Art and Design. Springer, pp. 89–100.
Horowitz, D., 1994. Generating rhythms with genetic algorithms. In: Proceedings of the Twelfth National Conference on Artificial Intelligence. AAAI'94, vol. 2. American Association for Artificial Intelligence, Menlo Park, CA, USA, pp. 1459–1460.
Huang, C.Z.A., Cooijmans, T., Roberts, A., Courville, A., Eck, D., 2019a. Counterpoint by convolution. Preprint. arXiv:1903.07227.

Huang, C.Z.A., Hawthorne, C., Roberts, A., Dinculescu, M., Wexler, J., Hong, L., Howcroft, J., 2019b. The Bach Doodle: approachable music composition with machine learning at scale. Preprint. arXiv: 1907.06637.

Hughes, D., Manaris, B., 2012. Fractal dimensions of music and automatic playlist generation: similarity search via MP3 song uploads. In: 2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP). Piraeus, Greece, pp. 436–440.

Huron, D.B., 2006. Sweet Anticipation: Music and the Psychology of Expectation. MIT Press.

Johanson, B.E., Poli, R., 1998. GP-Music: an interactive genetic programming system for music generation with automated fitness raters. Technical Report CSRP-98-13. University of Birmingham, School of Computer Science.

Jones, D., 2008. AtomSwarm: a framework for swarm improvisation. In: Proceedings of the 2008 Conference on Applications of Evolutionary Computing. Evo'08. Springer-Verlag, Berlin, Heidelberg, pp. 423–432.

Kaliakatsos-Papakostas, M., Cambouropoulos, E., 2014. Probabilistic harmonisation with fixed intermediate chord constraints. In: Proceeding of the Joint 11th Sound and Music Computing Conference (SMC) and 40th International Computer Music Conference (ICMC). ICMC–SMC 2014.

Kaliakatsos-Papakostas, M., Cambouropoulos, E., 2019. Conceptual blending of high-level features and data-driven salience computation in melodic generation. Cognitive Systems Research 58, 55–70.

Kaliakatsos-Papakostas, M.A., Epitropakis, M.G., Vrahatis, M.N., 2010. Musical composer identification through probabilistic and feedforward neural networks. In: Applications of Evolutionary Computation. In: LNCS, vol. 6025. Springer, Berlin/Heidelberg, pp. 411–420.

Kaliakatsos-Papakostas, M.A., Epitropakis, M.G., Vrahatis, M.N., 2011. Feature extraction using pitch class profile information entropy. In: Mathematics and Computation in Music. In: Lecture Notes in Artificial Intelligence, vol. 6726. Springer, Berlin/Heidelberg, pp. 354–357.

Kaliakatsos-Papakostas, M.A., Epitropakis, M.G., Floros, A., Vrahatis, M.N., 2012a. Interactive evolution of 8-bit melodies with genetic programming towards finding aesthetic measures for sound. In: Proceedings of the 1st International Conference on Evolutionary and Biologically Inspired Music, Sound, Art and Design. EvoMUSART 2012, Malaga, Spain. In: LNCS, vol. 7247. Springer Verlag, pp. 140–151.

Kaliakatsos-Papakostas, M.A., Floros, A., Vrahatis, M.N., 2012b. Intelligent generation of rhythmic sequences using Finite L-systems. In: Proceedings of the Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP 2012). Piraeus, Athens, Greece, 18–20 July, pp. 424–427.

Kaliakatsos-Papakostas, M.A., Floros, A., Vrahatis, M.N., 2012c. Intelligent real-time music accompaniment for constraint-free improvisation. In: Proceedings of the 24th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2012). Piraeus, Athens, Greece, 5–7 November, pp. 444–451.

Kaliakatsos-Papakostas, M.A., Floros, A., Vrahatis, M.N., Kanellopoulos, N., 2012d. Genetic evolution of L and FL–systems for the production of rhythmic sequences. In: Proceedings of the 2nd Workshop in Evolutionary Music Held During the 21st International Conference on Genetic Algorithms and the 17th Annual Genetic Programming Conference (GP). GECCO 2012, Philadelphia, USA, 7–11 July, pp. 461–468.

Kaliakatsos-Papakostas, M., Epitropakis, M., Floros, A., Vrahatis, M., 2013a. Chaos and music: a study of tonal time series and evolutionary music composition. International Journal of Bifurcation and Chaos.

Kaliakatsos-Papakostas, M., Floros, A., Vrahatis, M.N., 2013b. evoDrummer: deriving rhythmic patterns through interactive genetic algorithms. In: Evolutionary and Biologically Inspired Music, Sound, Art and Design. In: Lecture Notes in Computer Science, vol. 7834. Springer, Berlin, Heidelberg, pp. 25–36.

Kaliakatsos-Papakostas, M., Floros, A., Drossos, K., Koukoudis, K., Kyzalas, M., Kalantzis, A., 2014a. Swarm lake: a game of swarm intelligence, human interaction and collaborative music composition. In: ICMC.

Kaliakatsos-Papakostas, M., Katsiavalos, A., Tsougras, C., Cambouropoulos, E., 2014b. Harmony in the polyphonic songs of Epirus: representation, statistical analysis and generation. In: 4th International Workshop on Folk Music Analysis (FMA).

Kaliakatsos-Papakostas, M., Floros, A., Vrahatis, M.N., 2016. Interactive music composition driven by feature evolution. SpringerOpen 5 (1), 826.
Kaliakatsos-Papakostas, M., Queiroz, M., Tsougras, C., Cambouropoulos, E., 2017. Conceptual blending of harmonic spaces for creative melodic harmonisation. Journal of New Music Research 46 (4), 305–328.
Kennedy, J., 2010. Particle swarm optimization. In: Encyclopedia of Machine Learning, pp. 760–766.
Kingma, D.P., Welling, M., 2013. Auto-encoding variational Bayes. Preprint. arXiv:1312.6114.
Koops, H.V., Magalhães, J.P., de Haas, W.B., 2013. A functional approach to automatic melody harmonisation. In: Proceedings of the First ACM SIGPLAN Workshop on Functional Art, Music, Modeling & Design. FARM '13. ACM, New York, NY, USA, pp. 47–58.
Kranenburg, P.V., Backer, E., 2004. Musical style recognition – a quantitative approach. In: Parncutt, R., Kessler, A., Zimmer, F. (Eds.), Proceedings of the Conference on Interdisciplinary Musicology (CIM04). Graz, Austria, 15–18 April, pp. 1–10.
Krumhansl, C.L., 2001. Cognitive Foundations of Musical Pitch. Oxford University Press.
Lee, D.D., Seung, H.S., 1999. Learning the parts of objects by non-negative matrix factorization. Nature 401 (6755), 788.
Leman, M., 2012. Music and Schema Theory: Cognitive Foundations of Systematic Musicology, vol. 31. Springer Science & Business Media.
Liang, F.T., Gotham, M., Johnson, M., Shotton, J., 2017. Automatic stylistic composition of Bach chorales with deep LSTM. In: ISMIR, pp. 449–456.
Liu, H.M., Wu, M.H., Yang, Y.H., 2018. Lead sheet generation and arrangement via a hybrid generative model. In: Proc. Int. Soc. Music Information Retrieval Conf., Late Breaking and Demo Papers.
Lo, M.Y., 2012. Evolving cellular automata for music composition with trainable fitness functions. Ph.D. thesis. University of Essex.
Lourenc, B.F., Brand, C.P., 2009. L–systems, scores, and evolutionary techniques. In: Proceedings of the SMC 2009–6th Sound and Music Computing Conference, pp. 113–118.
MacCallum, R.M., Mauch, M., Burt, A., Leroi, A.M., 2012. Evolution of music by public choice. Proceedings of the National Academy of Sciences 109 (30), 12081–12086.
Machado, P., Romero, J., Manaris, B., Santos, A., Cardoso, A., 2003. Power to the critics – a framework for the development of artificial critics. In: Proceedings of 3rd Workshop on Creative Systems, 18 th International Joint Conference on Artificial Intelligence (IJCAI 2003). Acapulco, Mexico, pp. 55–64.
Madison, G., Sioros, G., 2014. What musicians do to induce the sensation of groove in simple and complex melodies, and how listeners perceive it. Frontiers in Psychology 5.
Madsen, S.T., Widmer, G., 2007. A complexity-based approach to melody track identification in MIDI files. In: Proceedings of the International Workshop on Artificial Intelligence and Music (MUSIC-AI 2007). Held in Conjunction with the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), Hyderabad, India, 6–12 January, pp. 1–9.
Makris, D., Kaliakatsos-Papakostas, M., Karydis, I., Kermanidis, K.L., 2017. Combining LSTM and feed forward neural networks for conditional rhythm composition. In: International Conference on Engineering Applications of Neural Networks, pp. 570–582.
Makris, D., Kaliakatsos-Papakostas, M., Karydis, I., Kermanidis, K.L., 2019. Conditional neural sequence learners for generating drums' rhythms. Neural Computing & Applications 31 (6), 1793–1804.
Manaris, B., Purewal, T., McCormick, C., 2002. Progress towards recognizing and classifying beautiful music with computers – MIDI-encoded music and the Zipf-Mandelbrot law. In: Proceedings IEEE SoutheastCon. 2002. IEEE, pp. 52–57.
Manaris, B., Romero, J., Machado, P., Krehbiel, D., Hirzel, T., Pharr, W., Davis, R.B., 2005. Zipf's law, music classification, and aesthetics. Computer Music Journal (ISSN 0148-9267) 29 (1), 55–69. ArticleType: research-article / Full publication date: Spring, 2005 / Copyright © 2005 The MIT Press.
Manaris, B., Roos, P., Machado, P., Krehbiel, D., Pellicoro, L., Romero, J., 2007. A corpus-based hybrid approach to music analysis and composition. In: Proceedings of the 22nd National Conference on Artificial Intelligence, vol. 1. AAAI Press, Vancouver, British Columbia, Canada, pp. 839–845.
Manaris, B., Hughes, D., Vassilandonakis, Y., 2011. Monterey mirror: combining Markov models, genetic algorithms, and power laws. In: Proceedings of 1st Workshop in Evolutionary Music, 2011 IEEE Congress on Evolutionary Computation (CEC 2011), pp. 33–40.

Manousakis, S., 2006. Musical L-systems. Master's thesis. The Royal Conservatory, the Hague, the Netherlands.
Matic, D., 2010. A genetic algorithm for composing music. Yugoslav Journal of Operations Research 20 (1), 157–177.
McCormack, J., 1996. Grammar-based music composition. Complexity International 3.
Mckay, C., Fujinaga, I., 2004. Automatic genre classification using large high-level musical feature sets. In: Proceedings of the 5th International Society for Music Information Retrieval Conference (ISMIR 2004). Barcelona, Spain, 10–14 October, pp. 525–530.
McKay, C., Fujinaga, I., 2006. jSymbolic: a feature extractor for MIDI files. In: ICMC.
Millen, D., 2005. An interactive cellular automata music application in Cocoa. In: Proceedings of the 2004 International Computer Music Conference (ICMC 2004).
Miranda, E.R., 2001. Evolving cellular automata music: from sound synthesis to composition. In: Proceedings of the Workshop on Artificial Life Models for Musical Applications (ECAL 2001), pp. 87–98.
Miranda, E.R., Al Biles, J., 2007. Evolutionary Computer Music. Springer.
Moroni, A., Manzolli, J., Zuben, F.V., Gudwin, R., 2000. Vox populi: an interactive evolutionary system for algorithmic music composition. Leonardo Music Journal, 49–54.
Mozer, M.C., 1994. Neural network music composition by prediction: exploring the benefits of psychoacoustic constraints and multi-scale processing. Connection Science 6 (2–3), 247–280.
Mozer, M.C., Soukup, T., 1991. Connectionist music composition based on melodic and stylistic constraints. In: Advances in Neural Information Processing Systems, pp. 789–796.
Nuanáin, C.O., Herrera, P., Jorda, S., 2015. Target-based rhythmic pattern generation and variation with genetic algorithms. In: Sound and Music Computing Conference.
Özcan, E., Ercal, T., 2008. A genetic algorithm for generating improvised music. In: Monmarché, N., Talbi, E.G., Collet, P., Schoenauer, M., Lutton, E. (Eds.), Artificial Evolution. In: Lecture Notes in Computer Science, vol. 4926. Springer, Berlin/Heidelberg, pp. 266–277.
Pachet, F., Roy, P., 2001. Musical harmonization with constraints: a survey. Constraints 6 (1), 7–19.
Pachet, F., Roy, P., Barbieri, G., 2011. Finite-length Markov processes with constraints. In: Twenty-Second International Joint Conference on Artificial Intelligence.
Paiement, J.F., Eck, D., Bengio, S., 2005. A probabilistic model for chord progressions. In: Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR).
Paiement, J.F., Eck, D., Bengio, S., 2006. Probabilistic melodic harmonization. In: Conference of the Canadian Society for Computational Studies of Intelligence, pp. 218–229.
Pearce, M., 2000. Generating rhythmic patterns: a combined neural and evolutionary approach. Ph.D. thesis, Master's thesis. Department of Artificial Intelligence, University of Edinburgh, Scotland. http://www.soi.city.ac.uk/ek735/msc/msc.html, 2000.
Phon-Amnuaisuk, S., Wiggins, G.A., 1999. The four-part harmonisation problem: a comparison between genetic algorithms and a rule-based system. In: Proceedings of the AISB'99 Symposium on Musical Creativity, pp. 28–34.
Phon-Amnuaisuk, S., Smaill, A., Wiggins, G., 2006. Chorale harmonization: a view from a search control perspective. Journal of New Music Research 35 (4), 279–305.
Price, K., Storn, R.M., Lampinen, J.A., 2006. Differential Evolution: a Practical Approach to Global Optimization. Springer Science & Business Media.
Prusinkiewicz, P., 1986. Score generation with L–systems. Computer, 455–457.
Prusinkiewicz, P., Lindenmayer, A., 2012. The Algorithmic Beauty of Plants. Springer Science & Business Media.
Purwins, J., Graepel, T., Blankertz, B., Obermayer, C., 2004. Correspondence analysis for visualizing interplay of pitch class, key, and composer. In: Mazzola, G., Noll, T., Luis-Puebla, E. (Eds.), Perspectives in Mathematical and Computational Music Theory. Universities Press, pp. 432–454.
Raczyński, S.A., Fukayama, S., Vincent, E., 2013. Melody harmonization with interpolated probabilistic models. Journal of New Music Research 42 (3), 223–235.
Reynolds, C.W., 1987. Flocks, Herds and Schools: A Distributed Behavioral Model, vol. 21(4). ACM.
Roberts, A., Engel, J., Raffel, C., Hawthorne, C., Eck, D., 2018. A hierarchical latent vector model for learning long-term structure in music. Preprint. arXiv:1803.05428.

Rohrmeier, M., 2011. Towards a generative syntax of tonal harmony. Journal of Mathematics and Music 5 (1), 35–53.
Sánchez, A., Pantrigo, J.J., Virseda, J., Pérez, G., 2007. Spieldose: an interactive genetic software for assisting to music composition tasks. In: Proceedings of the 2nd International Work-Conference on The Interplay Between Natural and Artificial Computation, Part I: Bio-Inspired Modeling of Cognitive Tasks. IWINAC '07. Springer-Verlag, Berlin, Heidelberg, pp. 617–626.
Serquera, J., Miranda, E.R., 2010. Evolutionary sound synthesis: rendering spectrograms from cellular automata histograms. In: European Conference on the Applications of Evolutionary Computation, pp. 381–390.
Shmulevich, I., Yli-Harja, O., Coyle, E., Povel, D.J., Lemstram, K., 2001. Perceptual issues in music pattern recognition: complexity of rhythm and key finding. Computers and the Humanities 35, 23–35.
Simon, I., Morris, D., Basu, S., 2008. MySong: automatic accompaniment generation for vocal melodies. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 725–734.
Sioros, G., Miron, M., Davies, M., Gouyon, F., Madison, G., 2014. Syncopation creates the sensation of groove in synthesized music examples. Frontiers in Psychology 5.
Solomos, M., 2005. Cellular automata in Xenakis' music. Theory and practice. In: Proceedings of the International Symposium Iannis Xenakis, pp. 120–137.
Spector, L., Alpern, A., 1995. Induction and recapitulation of deep musical structure. In: Proceedings of International Joint Conference on Artificial Intelligence. IJCAI, vol. 95, pp. 20–25.
Sturm, B., Santos, J.F., Korshunova, I., 2015. Folk music style modelling by recurrent neural networks with long short term memory units. In: 16th International Society for Music Information Retrieval Conference.
Sturm, B.L., Santos, J.F., Ben-Tal, O., Korshunova, I., 2016. Music transcription modelling and composition using deep learning. Preprint. arXiv:1604.08723.
Suzuki, S., Kitahara, T., Univercity, N., 2013. Four-part harmonization using probabilistic models: comparison of models with and without chord nodes. In: Proceedings of the Sound and Music Computing Conference, pp. 628–633.
Temperley, D., 2004. The Cognition of Basic Musical Structures. MIT Press.
Thom, B., 2001. Machine learning techniques for real-time improvisational solo trading. In: Proceedings of the 2001 International Computer Music Conference (ICMC).
Thornton, C., 2009. Hierarchical Markov modeling for generative music. In: International Computer Music Conference.
Todd, P.M., 1989. A connectionist approach to algorithmic composition. Computer Music Journal 13 (4), 27–43.
Tokui, N., 2000. Music composition with interactive evolutionary computation. Communication 17 (2), 215–226.
Unehara, M., Onisawa, T., 2005. Music composition by interaction between human and computer. New Generation Computing 23 (2), 181–191. Tokyo, Japan.
Vogl, R., Leimeister, M., Nuanáin, C.Ó., Jordà, S., Hlatky, M., Knees, P., 2016. An intelligent interface for drum pattern variation and comparative evaluation of algorithms. Journal of the Audio Engineering Society 64 (7–8), 503–513.
Weinberg, G., Godfrey, M., Rae, A., Rhoads, J., 2008. A real-time genetic algorithm in human-robot musical improvisation. In: Computer Music Modeling and Retrieval. Sense of Sounds. Springer-Verlag, Berlin, Heidelberg, pp. 351–359.
Wiggins, G., Papadopoulos, G., 1998. A genetic algorithm for the generation of jazz melodies. In: Proceedings of the Finnish Conference on Artificial Intelligence (STeP'98). Jyväskylä, Finnland.
Wilson, S., 2008. Spatial swarm granulation. In: Proceedings of the 2008 International Computer Music Conference.
Wolkowicz, J., Kulka, Z., Keselj, V., 2008. *n*-gram based approach to composer recognition. Archives of Acoustics 33 (1), 43–55.
Worth, P., Stepney, S., 2005. Growing music: musical interpretations of L-systems. In: EvoWorkshops. In: Lecture Notes in Computer Science. Springer, pp. 545–550.

Yang, L.C., Chou, S.Y., Yang, Y.H., 2017. MidiNet: a convolutional generative adversarial network for symbolic-domain music generation. Preprint. arXiv:1703.10847.

Yi, L., Goldsmith, J., 2007. Automatic generation of four-part harmony. In: Laskey, K.B., Mahoney, S.M., Goldsmith, J. (Eds.), BMA. In: CEUR Workshop Proceedings, vol. 268. CEUR-WS.org.

Yogev, N., Lerch, A., 2008. A system for automatic audio harmonization.

Zacharakis, A., Kaliakatsos-Papakostas, M., Tsougras, C., Cambouropoulos, E., 2018. Musical blending and creativity: an empirical evaluation of the chameleon melodic harmonisation assistant. Musicae Scientiae 22 (1), 119–144.

Zheng, E., Moh, M., Moh, T.S., 2017. Music genre classification: a n-gram based musicological approach. In: Advance Computing Conference (IACC), 2017 IEEE 7th International, pp. 671–677.