A hierarchical System for autonomous musical Creation

M. Anthony Reimer
School of Music
University of Illinois, Urbana-Champaign
Urbana, Illinois
reimer2@illinois.edu

Guy E. Garnett
School of Music
Illinois Informatics Institute
University of Illinois, Urbana-Champaign
Urbana, Illinois
garnett@illinois.edu

Abstract

We describe work in progress on the development of a new hierarchical model of machine creativity operating in the domain of music. Similar to the way human brains work, our system separates low-level components associated with pattern recognition and analysis from the high-level creative components in two extensible layers. Separating this functionality in different layers of our system provides better visibility into the behavior of the creative component. This increased visibility has led to many improvements over previous iterations including the reward calculation for the creative component. Additionally, the design of an abstract input feature layer allows for greater flexibility in the number and combination of low-level features that can be used within our system.

Introduction

After working with previous system designs, such as those described in (Smith and Garnett 2012), we decided it might be useful to try to separate out the different evaluative components of the system. We therefore divided our system into a set of low-level perception and short-term memory components, and a high-level component that would focus on "creative" choice or novelty. In this way, our system separates the part of our system involved in making the most *coherent* choice, in terms of matching what has come before, from the part designed to make the most *interesting* choice. This strategy of encapsulation allows us to a) better refine the reward calculation for our creative component, b) better understand the nature of the choices made by the system, and c) allows for the easy addition of musical feature components through a uniform programming interface without adversely affecting the component concerned with creativity.

Motivation

Elementary rules and theories of musical coherence are well-known, but musical creativity is less well understood. Creativity, including musical creativity, can be a nebulous concept. And, seeking a computational model of creativity given its implicitly unpredictable nature may seem quixotic, or even impossible. For our purposes, we begin with a very general definition of creativity from (Vartanian, Bristol, and Kaufman 2013); they suggest that creativity is "the generation of novel and useful products in a specific context." Further, following (Boden 2004), we would like to distinguish between historical creativity, which has a broad cultural and social context, and psychological creativity, which is creative within the context of an individual's experience. Our focus here is on the psychological.

As a specific starting point, we take Schmidhuber's theory of creativity (Schmidhuber 2010) that derives from notions of fun and interesting-ness a computational model of creativity. First, we take from his theory the intuition that producing a creative artifact requires being able to imagine experiencing it first. But this is not simply shaking marbles in a bag. Combinatorial creativity, as described by (Boden 2004), still requires that the combined ideas come together in an intelligible way—this is what makes them "useful in a specific context." In Schmidhuber, the imagination serves as a creator of prototypes for the artist to choose from and explore. While the raw materials may already be known, or at least are familiar, the artist's exploration of their relationships and/or transformations will often lead to the discovery of a configuration that is unfamiliar, yet resonant with previous material. This discovery, potentially at both the moment of creation and at the moment of audition, is initially quite exciting and may bring the artist a great deal of pleasure. But as the exploration continues, and the artist or auditor is able to understand the new discovery better, familiarity will lead to decreasing interest which will ultimately push the artist to move on to new territory and, if unchecked, lead the auditor to boredom.

Second, Schmidhuber's computational model of creativity includes a reinforcement learner that attempts to maximize the rate of change in its internal world model. The less change required to represent the world, the more compressed the representation is—the more it tends toward abstraction. We find his notion of information compression for a creative agent is also relevant in the domain of music. To account for aspects of music perception many of the formalisms of music are centered around compressing high-dimensional information both horizontally (in terms of melodic patterning) and vertically (where groups of notes that sound simultaneously are often described according to their relative relationships, their harmonic/contrapuntal function). Indeed, manipulating abstractions and applying them to different contexts is another mode of creative behavior. With these similarities in mind, we seek to leverage neural models and Schmidhuber's computational ideas in our current work.

Related Work

Since our aim is to realize an autonomous creative agent in an online unsupervised machine learning implementation, we therefore require a method of classification to manage our problem space that is both efficient and simple to analyze. While others have used suffix trees (Pachet 2003), genetic algorithms (Biles 2002), and support vector machine algorithms (Le Groux 2011), we have selected a neural network model known as *fuzzy adaptive resonance theory* (ART) (Carpenter, Grossberg, and Rosen 1991). The ART model is very efficient and can both train and run in real time. It can be trained one example at a time and its simple architecture allows for a clear understanding of its decisions and, perhaps more importantly, an easily computable metric of the current state of its learning. An ART learns progressively by adding new knowledge—additional network 'nodes'—when it is unable to at least minimally match an input pattern to one of its internal patterns. When the ART is able to match an input pattern to an internal pattern, the matching elements of the internal representation are held constant while elements not found in the input are decreased. Thus, an unsupervised ART a) is able to encode new information, b) can progressively refine its internal representation, and c) is resistant to losing information. These attributes make it an excellent choice for the implementation of a creative agent.

Additionally, since our pursuit is a musical one, our system derives ideas from many previous implementations in that domain. Computer-assisted algorithmic music composition can be traced back to the first experiments by Hiller, et al. in 1958 (Hiller and Beauchamp 1965). However, systems like ours that attempt to model creative impulses, as opposed to modeling rules or generative structures, are relatively recent. Still, many examples exist from which to draw inspiration. For example, Pachet's use of multiple musical viewpoints from (Conklin and Witten 1995) in his *Continuator* system (Pachet 2003) can be seen as a precedent for the separation of our system's analytical components into a different layer from the creative creative component. And similar to Biles' GenJam (Biles 2002), our system seeks to exert some measure of control over the ratio of novelty to sameness. Like (Mozer 1994; Eck and Schmidhuber 2002), our system does "note-by-note" generation. Since we are concerned to model creativity rather than more general musical constructs, we focus here exclusively on melodic generation since that simplifies the models and makes it easier to judge results. We have also made extensive use of concepts from (Smith and Garnett 2012).

Architecture

In order to clearly separate perceptual from creative components, our system has two distinct layers. The bottom layer, called the *feature component layer*, is made up of components designed to define and remember features. The top layer, the *creative layer*, chooses the most interesting pitch.

Similar to (Smith and Garnett 2012), we started with a configuration wherein each component contributed a measure of its preference to an overall reward calculation. However, comparing the internal representations of the lower level components revealed relatively large differences in their distributions and corresponding large differences in the reward contribution. When the internal representations of the various feature components were substantially different, their contributions to the reward were incommensurate.

In the new implementation, each component of the feature layer independently provides a candidate pitch to the creative layer based on its independent representation, rather than contributing to the reward score. The candidate pitch is the one receiving the highest reward (i.e., the *best* choice for that component). This allows the calculation of each feature component's reward to be independent of the rest of the system, and to use any measure of goodness of fit that works best for a particular feature. The feature components' candidate pitches are passed to the creative layer.

As in (Smith and Garnett 2012), and following (Schmidhuber 2010), we take the amount of change in the internal representation of the creative layer as a measure of how much the system is learning. This is derived from Schmidhuber's theory that postulates that an improvement in a system's ability to compress its world model is an indication of its learning or assimilating a new concept and in turn serves as an impetus for further exploration (Schmidhuber 2010). In our case, we assume that if an ART node's weights are changing substantially, that implies it is still learning. If the node is still learning, that implies that it is still exploring, still creating something new. Therefore, for each candidate pitch proposed by the lower layer, we calculate the change in internal weights that pitch would induce in the creative layer's ART nodes. We call this change in a node's weights δW.

This reward based on δW is calculated for each of the feature layer's candidate pitches. Once the pitch generating the highest δW is determined, it is selected as the new pitch. This selected pitch must now be learned by all the modules. Thus, it is presented to the lower feature layer so that each component can update their independent world models to accommodate what the upper layer has selected. This is shown schematically in Fig. 1 where the input, representing the currently selected pitch, is presented to each of the features components, which each then propose a new pitch to the creative layer, the 'Composer.' Once the composer has selected a new pitch, it is sent to the output and back around to the input to continue the process.

Feature Component Layer

The feature component layer is designed to accommodate a variety of representations of the system's inputs and serve as a filter ensuring the creative layer is presented with a meaningfully coherent set of pitches from which to make its choice. Extensible in design, each component on this layer is responsible for 1) keeping track of its own representation

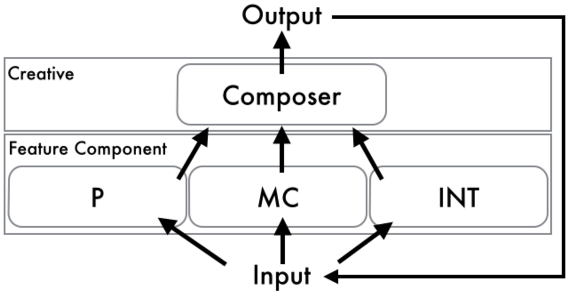

Figure 1: High-level architecture

of the world, 2) incorporating new inputs into that representation, and 3) making a choice of the next note that most closely aligns with its representation. The mechanisms by which it performs these tasks are left up to each component allowing for a great deal of flexibility in how components are designed. The autonomy of each component additionally negates the need for these components to be comparable. We are currently experimenting using three feature components: *pitch* (P), which captures recent pitch activity and nominates a pitch based on its ability to match its memory of recent pitches; *melodic interval* (INT), which measures the interval from one pitch to the next and nominates a new pitch based on how well the interval it makes with the previous pitch matches remembered intervals; and, *melodic continuity* (MC), which serves as a melodic continuity detector as explained further below. In order to better observe and understand our creative component, we are not currently explicitly modeling rhythm or simultaneity; the system generates a single note at each time step. Importantly, we do encode memory in a fundamental way, described next.

Pitch and Melodic Interval Both the P and INT components are comprised of: a *short-term memory unit* (STM); a classifying long-term memory, an ART; and a decision process. When asked to supply a candidate pitch to the creative layer, each component in the feature component layer evaluates each possible next note. The STM for both of these components is based on a continuously running spatial encoding approach that both allows them to keep track of the set of most recent events as well as their order in time by applying decay as each new event arrives (Gjerdingen 1990). The only differences between the P and INT components lie in the space and makeup of the features they are representing. In the case of P, incoming semitone pitches are represented locally, as a set of weights associated with integer values from $0 - 11$, see Fig. 2. For INT, signed melodic intervals are encoded in semitone units directly into a vector containing 23 weights associated with integers spanning -11 to $+11$ (e.g., an ascending whole step is encoded as $+2$, a descending fourth as -5, etc.). As each new event is processed, the STM's weights are decayed in a manner consistent with short-term auditory memory, on the order of ~ 5 seconds (Le Groux 2012).

The long-term memory for P and INT is taken care of by an implementation of fuzzy adaptive resonance theory, an

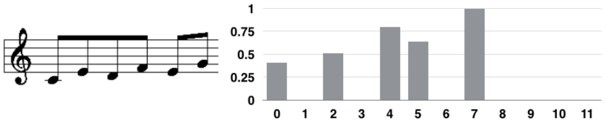

Figure 2: Example encoding of simple melody

ART. The internal representation of the ART is a dynamic set of nodes with the same dimension as that of the STM. The ART calculates a winner determined by a *reward score* (R) based on the fuzzy match between each of its nodes and its STM. In (Smith and Garnett 2012) the total R for each proposed pitch was further modified by the amount of learning required to represent it. In the present model, the P and INT components simply calculate R based on the best match according to their memory of previous pitches and intervals. That is, they focus on making the most coherent choice, leaving the creative decision to the creative layer. A 'vigilance' check, as in (Carpenter, Grossberg, and Rosen 1991), is in place to ensure that a chosen node's weights represent the STM to a reasonable degree. A higher vigilance threshold will produce output that is more reflective of the STM, whereas lower vigilance will allow for matches with nodes that are less similar to the STM. Importantly, this allows for longer term memory, and hence longer term unfolding of relationships. Once the P or INT component has calculated R for each possible pitch, it chooses the pitch with the best R. To repeat, the components at this feature level do not try to be creative, they are solely concerned with finding the *best* match according to their individual criteria. Each component then sends its candidate pitch to the creative layer. See (Carpenter, Grossberg, and Rosen 1991) for details of the matching, vigilance and the learning algorithm.

Melodic Continuation The goal of MC is to provide the system with the ability to choose pitches that enhance *melodic* continuity, as opposed to the pitch or interval continuity of the other modules. Our default uses semitone continuity, but later we show an example of a more diatonic model. While P chooses pitches that best fit remembered pitch patterns, and INT chooses pitches that best fit remembered interval patterns, MC chooses pitches close to, but not the same as pitches that have already been chosen. That is, it will tend to prefer neighbor notes to previously heard pitches. MC is simpler in structure than P and INT in that it only uses a single structure to encode and make its predictions. Long term memory is inherent in its design since a given pitch will remain in memory indefinitely until it is 'cancelled' by a neighbor note—it remembers edge pitches. Pitches are encoded using a local representation in MC, but then pitches above and below the current input pitch, n, are decayed exponentially

$$w_{n\pm s}^{new} = dw_{n\pm s}^{old} \quad (1)$$

where s is the number of steps we are considering to be in proximity, and d is the decay factor. After encoding the probe pitch, the reward for the current test pitch is calculated as:

$$R_n = \frac{\sum_{i=1}^{D} w_i^{old}}{\sum_{i=1}^{D} w_i^{new}} \qquad (2)$$

where D is the size of the vector MC is encoding and w represents all the weights of MC before and after the encoding. This leads to a score that is always greater than or equal to 1 due to the way in which the inputs are encoded. Using this calculation, MC nominates the pitch that provides the best continuity to previous inputs.

Creative Layer

Structurally, the creative layer has a lot in common with both the P and INT feature components. However, there are some differences between the ART in this layer and those in the lower layer. First, the reward calculation is quite different. It is not trying to make the closest match to previous inputs, like the feature layer components, rather it is trying to make an interesting choice. To that end, the creative layer's reward is calculated using how much change in the ART's weights a chosen pitch might cause. By itself, this process would lead to the ART always selecting new material. But in this case, this does not happen due to our hierarchical structure because only the coherent pitches approved by the low-level feature classifiers are available. This is accomplished using the calculation:

$$\delta W = \sum_{i=1}^{D} w_i^{old} - w_i^{new} \qquad (3)$$

Previously, δW was used as a gate with its value being compared against a threshold. Using δW in this way signaled when a significant change had taken place in a node. (Smith and Garnett 2012) also incorporated the sum of the number of nodes created in each ART of their hierarchy in their reward calculation. We removed the node counting factor of the calculation since our current way of calculating δW automatically gives a high reward for new nodes which allows for a more direct understanding of the learning behavior of the creative ART.

When using δW in the reward calculation, it is helpful to be able to balance the high reward the creative component receives for making the most novel choice (the one causing the most change in a node) against the rewards it receives for making choices that reflect something it has already learned. To give us a means to experiment with different levels of novelty tolerance, we apply a Wundt curve to the δW factor before it is used in the reward. This allows us to select an adventurous or more conservative composer at will.

Thus, the current model gathers the reward functionality concerned with *interesting-ness* into one high-level unit, our creative layer. This allows the feature layers to focus on pattern matching and proposing candidate events that best fit their individual criteria and leaves the creative functions unified in one top level component.

After calculating δW, the creative layer's ART also uses a vigilance test to trigger the creation of a new node when it fails to find a node that is minimally able to represent a change to its world model. In this way, it is ensured that the creative layer's ART has an accurate representation of its input.

Once all the candidate pitches have been processed by the ART and a winner is determined, that winner is presented to all the members of the feature component layer. Each of these components is, in turn, responsible for incorporating that winner into its long term memory and world model. In the case of the P and INT components, this means updating each STM and ART node weights, adding new nodes as necessary. For MC, it encodes its inputs as before, but in a final step applies an urgency factor (U) to the entire set of its weights, pushing them asymptotically toward 1 as long as they have no neighbors. This imbues MC with a greater need to return to those pitches over time and, in the long run, prevents wild divergences in the melodic shape of the system's output.

Results

While it cannot really be described as exciting, the performance under this architecture produces output that is both sensible, in that the output can be seen to derive from the mechanics in place and the small seed it is given, as well as controllable, for the general effects of changing various parameters can be predicted, even though particular solutions are novel. Figure 3 shows output reflecting representative behavior under the following nominal settings. For the feature component layer, the P and INT components each had an STM with an exponential decay rate of 0.8 and each ART had a learning rate of 0.4 and a vigilance threshold of 0.5. The MC component had an exponential decay rate of 0.0, applied to pitches one semitone removed from its input and an urgency factor (U) of 0.1. On the creative layer the STM also had an exponential decay rate of 0.8, and the ART had a learning rate of 0.4, and a slightly lower vigilance threshold of 0.4.To facilitate understanding of the creativity component, tests were framed within a limited set of pitches encompassing only a single octave using a relatively small seed of three pitches, C#-D#-G, shown below in hollow note heads.

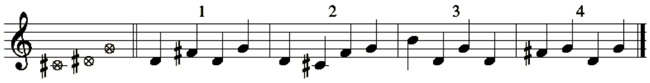

Figure 3: Output of system, nominal settings

In Figure 3, we examine the output of the system with these nominal settings. The effects of MC can be seen right from the top in this case: the D it nominates clearly shows its predilection to connect to the C# and D# from the seed. MC's influence can also be observed in the neighboring semitones chosen later in the excerpt at bar 2, beat 2 and bar 4, beat 1. The INT component's contribution to the output is most easily seen for the 3 beats beginning at bar 2, beat 3. The M3, M2, M3 sequence is entirely derived from the intervals introduced in the seed. P's effect is the hardest to uncover in this short passage. The introduction of new pitches on the first two beats has forced P's ART to expand, creating a new node in order to provide minimally useful candidates to the upper layer. Thus, P nominates the D from 1:1 and the last pitch of the seed (G) while the first two notes from the seed do not appear as candidates.

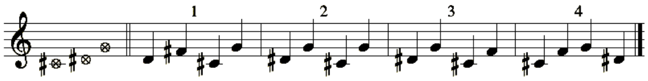

Figure 4: Output of system, P's vigilance raised

In Figure 4, the vigilance for P's ART is raised to 0.8 to see if encouraging it to remain truer to the seed has an effect on output. This results in changed output due to P no longer nominating the D from 1:1. The result includes a repetitive G from the seed. Additionally, the C# and D# are not discarded in this case and are nominated several times over the course of the next few bars.

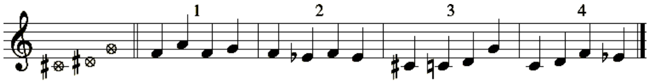

Figure 5: Output of system, expanded MC influence

To observe the manner in which MC connects the pitches it has previously encoded, we expand the range of neighboring pitches upon which it acts. This is accomplished through increasing the value of s in Eq. 1 from 1 to 2. Gone are the semitones, replaced by whole-step motion. (see Fig. 5).

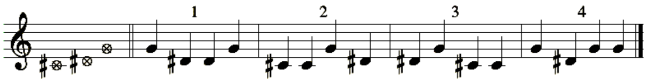

Figure 6: Output of system, C's vigilance raised

Even though C's rewards are based off of a different calculation, we performed some experiments to better understand the importance of vigilance on the creative layer. It is possible to observe the equivalent effect of manipulating the vigilance of C's ART, whether it is increased (see Fig. 6), or decreased (see Fig. 7).

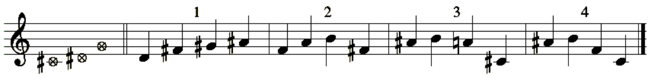

Figure 7: Output of system, C's vigilance lowered

Future Directions

The current performance of the creative component is encouraging. However, a couple of enhancements could potentially be pursued. Experiments seemed to reveal that some of the ART parameter tuning may be automated. Automatically calculating optimal values for parameters such as the low δW threshold or those associated with the Wundt function would allow the system to work in arbitrary musical contexts. Further experiments with a greater range of input features, most noticeably rhythm, as well as a more sophisticated means of evaluation would be required to fully evaluate this and are also needed to provide more context to the system. The flexibility builtin to the architecture allows for quick prototyping of new potential feature components as well as testing combinations of different features. Thus, quick evaluation of new input feature configurations incorporating concepts from the disciplines of neurocognition and psychology may be seamlessly commingled in this system. Given our current success at a relatively low level, it would also be exciting to extend our system to include a musically higher context. Whether that involves simply creating stacks of our current creative component or incorporating it into a different architecture, we have a solid foundation on which to build.

References

Biles, J. A. 2002. Genjam in transition: from genetic jammer to generative jammer. In *Generative Art*, volume 2002.

Boden, M. A. 2004. *The creative mind: Myths and mechanisms*. Psychology Press.

Carpenter, G. A.; Grossberg, S.; and Rosen, D. B. 1991. Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks* 4(6):759–771.

Conklin, D., and Witten, I. H. 1995. Multiple viewpoint systems for music prediction. *Journal of New Music Research* 24(1):51–73.

Eck, D., and Schmidhuber, J. 2002. Finding temporal structure in music: Blues improvisation with lstm recurrent networks. In *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, 747–756. IEEE.

Gjerdingen, R. O. 1990. Categorization of musical patterns by self-organizing neuronlike networks. *Music Perception* 339–369.

Hiller, L., and Beauchamp, J. 1965. Research in music with electronics. *Science* 150(3693):161–169.

Le Groux, S. 2011. Situated, perceptual, emotive and cognitive music systems.

Le Groux, S. 2012. *The Smuse: An Embodied Cognition Approach to Interactive Music Composition*. Ann Arbor, MI: MPublishing, University of Michigan Library.

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.

Pachet, F. 2003. The continuator: Musical interaction with style. *Journal of New Music Research* 32(3):333–341.

Schmidhuber, J. 2010. Formal theory of creativity, fun, and intrinsic motivation (1990-2010). *Autonomous Mental Development, IEEE Transactions on* 2(3):230–247.

Smith, B., and Garnett, G. 2012. Machine listening: Acoustic interface with art. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, 293–296. ACM.

Vartanian, O.; Bristol, A. S.; and Kaufman, J. C. 2013. *Neuroscience of Creativity*. MIT Press.