Coming to “Terms” with Creativity

M. Anthony Reimer

May 13, 2014

1 Introduction

Intuitively, it might not be easy to understand that the creative process of an artist is also an explorative one. But, bringing something into existence requires being able to imagine experiencing it first. In this instance, the imagination serves as an unmapped territory for the artist to explore. The objects residing in this territory are often familiar. Yet, the artists exploration of their relationships and/or transformations will often lead to the discovery of a configuration that is unfamiliar. This discovery is initially quite exciting and may bring the artist a great deal of pleasure. But as the exploration continues, and the artist is able to understand the new discovery better, familiarity will lead to restlessness which will ultimately push the artist to move on to new territory. This pattern of exploration and discovery leading to comprehension in turn leading to further exploration aligns closely with Jürgen Schmidhuber’s theory of creativity[2] which serves as the part of the impetus for this current research project.

2 Motivation

The present work described in these pages is motivated by two somewhat unrelated projects: an exploration of computational creativity; and a data mining project based around large record sets concerned with musical taste.

The first part of this present endeavor is an attempt to design a creative, generative system for composing music that is not predicated on repeating elements nor pre-figured rules governing the unfolding of a score’s musical ideas. Instead, a system design is imagined that leverages a machine learning technique known as Adaptive Resonance Theory[1], or simply ART. In short, a generic fuzzy ART implementation[3] has been extended to provide the ability to identify patterns in an input vector, dynamically encode them into a neural network. Changes in this network are subsequently monitored and leveraged to make decisions. This system is designed to be not only capable of reflecting a individual’s ideas but is also able to respond *creatively* to input from external sources.

To provide some context, the features of an autonomous agent designed to compose music are many and varied. They include at least: pitch content and pitch sequences, interval sequence and content, timbre, rhythm. Ultimately, consideration for all of these features will be necessary to produce a wholly satisfying compositional system. For now, efforts are focused on the design of a component that can make creative decisions and a system that can effectively leverage that component’s capabilities to produce coherent content.

The second project being brought to bear currently is centered around a contest previously run by the International Conference on Auditory Display (ICAD). The contest, a part of the 2012 conference embracing the theme 'Listening to the World Listening,' was focused on a number of statistics derived from Twitter data (see http://icad2012.icad.org/contest.html for details regarding the competition). A web service was created for use by contest participants to analyze and leverage the summarized Twitter data in the contest entries.

For the current endeavor, the data from the second project has been mined and refined to train the computationally creative system from the first project.

3 Design of the Creative Component

3.1 Overview

In beginning our exploration of Schmidhuber's theories, a system has been devised that comprises two layers. The bottom layer, or feature component layer, is made up of components designed to provide a candidate pitch to the higher layer, the learner layer. A typical implementation would include a training epoch in which both layers are presented with inputs presumably reflecting the characteristics of the desired output. Then, the learner is asked to start making decisions.

This begins with the bottom layer. A candidate is chosen by each of the feature components and will be a pitch that receives the highest reward score for that component. The calculation of the feature component's reward is independent of the rest of the system. The feature components' candidates are passed to the higher layer which will choose its output from among these candidates. However in the case of the higher layer, the pitch chosen will be the one that will effect the most change in the upper layer's current world model. The winner of the upper layer is then presented to the feature component layer, so that its components may adjust their component-specific world models to accommodate the upper layer's choice.

3.2 Feature Component Layer

The feature component layer is designed to accommodate a variety of ways to extract features from the system's inputs. Each component needs to be able to keep track of the world, be able to make a prediction that most closely aligns with its model, update its world model from external input, and learn from the updates to its world model where necessary. The mechanisms by which it performs these tasks are left up to each component providing a great deal of flexibility in how components are designed. Current experiments use three feature components: match(M), melodic continuation(MC), and melodic interval(Int).

3.2.1 Match

The Match component is designed to produce a candidate pitch that most closely aligns with its current world model. It contains an STM and an ART. When asked to make a prediction, this component will loop through every potential candidate. Each potential candidate is used to temporarily update the component's world model. This updated world model is in turn presented to the *ART* which calculates a reward for each of its nodes according to the formula:

$$R = \frac{\sum_{i=1}^{D} \left(\min\left(I_i,\ w_i\right)\right)}{a\ +\ \sum_{i=1}^{D} w_i}$$

where D is the size of the weight vector for a given node, I is the input vector passed in from the ART, w is the vector of the nodes weights, a is called the choice factor which is a very small number intended to ensure we do not end up dividing by 0.

The pitch with the highest R is named M's candidate pitch.

3.2.2 Melodic Continuation

The goal of the *MC* is to provide the learner with some small notion of where it has been recently and a desire to choose pitches close to those locations. This serves to work in concert with M's intrinsic drive to choose things that have already been chosen. The *MC* is simpler than M in that it only uses a single structure to encode and make its predictions. An input is encoded in the *MC* using the following:

$$w_n^{new} = \max\left(w_n^{old},\ U\right)$$

where n is the current probe input. Additionally, the pitches surrounding the input pitch are decayed by:

$$w_{n\pm s}^{new} = w_{n\pm s}^{new} d$$

where s is the number of steps we are considering to be in proximity. The reward for the current probe sum of the current weights for the MC are divided by the sum of the new weights after the encoding of the probe pitch:

$$R_n = \sum_{i=1}^{D} \frac{w_i^{old}}{w_i^{new}}$$

where w represents all the weights of the *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 probe inputs are encoded.

Once all the probe pitches have been processed by the MC and the ART and a winner is determined, the MC, in a final step, then applies the urgency factor (U) to the entire set of its weights:

$$w_i^{new} = w_i^{old}\left(1\ +\ \left(U\left(1\ -\ w_i^{old}\right)\right)\right)$$

The goal here is to have the weights gradually increase at a faster rate the longer they remain unchosen. This is to imbue the MC with a greater need to return to notes over time. In the long run, this approach prevents wild divergences in the melodic shape of the system's output.

3.2.3 Melodic Interval

This component functions identically to M. However, it encodes its inputs slightly differently. Where M encodes its input in a 1:1 fashion, Int encodes the difference between two successive inputs. This acts as a foil to both M and MC in that it will favor pitches that create distances to the previous pitch that are similar to the distances represented by its encoding.

3.3 Learner Layer

The learner shares a common structure with both the M and Int feature components. However, there are some differences between the learner and these other components. Firstly, the reward calculation for the learner component is quite different. Since the current system is intended to be a creative system, it follows that it would be desirable for the learner to make an interesting choice. To that end, the learner's reward is calculated using how much change in the ART's weights a chosen pitch might cause using the calculation:

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

Another difference between the learner and the other layers is that it is only concerned with the pitches that are presented to it by the bottom layers. This is in contrast to the evaluation of the entire pitch space that is performed by the lower components.

A third difference between the learner and the lower layers has to do with when its ART expands its collection of nodes to encode more information. Both layers use a vigilance test trigger the creation of a new node if the ART fails to find a node that is minimally able to accurately represent a change to its world model. This vigilance score is calculated as:

$$V = \frac{\sum_{i=1}^{D} (\min(I_i,\ w_i))}{a\ +\ \sum_{i=1}^{D} I_i}$$

However, the learner's ART will also expand its representation of the world by creating a new node in circumstances where the δW, as calculated above, falls below a threshold. In this way, it is ensured that the learner's ART not only has an accurate representation of its input, but also that that representation does not become stagnant and overly specific. In this way, the learner is further encouraged towards acting creatively.

4 Sonifying the System

Having run the system described above through a number of experiments and verified that it behaves as designed, a more significant test using real-world data seemed warranted. The data from the ICAD contest previously mentioned, while not ideal, was more than adequate for this purpose.

The ICAD dataset consisted of summary data pertaining to the mentions of musical genres, called *terms*, in messages, called tweets, on the popular social media tool, Twitter. Additionally, *terms* were correlated to the mention of musical artists within the same tweets. The data points included which term, which artist, the frequency of use of the term's association to the artist, the relative popularity of the artist, the artists's location, etc.

The summaries were generated every 5 seconds and made available to the participants via a web service. However, the implementation currently used only consumed the web service once every 5 minutes. This seemed a good trade-off between capturing trends and storage size. Thus, the data was collected in batches every 5 minutes between December 12th, 2011 and January 26th, 2012. This resulted in the collection of 1,114 unique *terms*, 12,000 unique artists and over 7 million records where *terms* were associated with artists.

From a practical standpoint, it was necessary to set up constraints regarding what data would be used. As the original inspiration for the project was borne out of an interest in what *terms* were in use to describe music, the pruning of the dataset began there. For the first step, the 25 most frequently used *terms* were retained for use. From this list, a number of overlap *terms* could be identified. Obviously owing to personal opinion, overlap *terms* were arbitrarily defined as *terms* that were similar or simply a sub-genre of other *terms*, (e.g. pop, j-pop, dance pop). This method of pruning practically reduced the top 25 *terms* to 12 (in order most frequently mentioned to least):

- rock
- hip hop
- pop
- electronic
- rap
- heavy metal
- alternative rock
- jazz
- punk
- funk
- blues
- neo soul

Upon arriving at this set of *terms*, the challenge became how to use them as part of the sonification. Further interest in exploring the relationships of these *terms* to other *terms*, the top 12 were simply filtered out from the rest. However, the recordset's size still proved a little unwieldy. At that point, it seemed prudent to attempt to isolate the *terms* in time. Therefore, the 5 minute segment, or batch, in which the *terms* were most often mentioned relative to other *terms* became the focus of the sonification for that *term*.

To give an example, for most batches the term *jazz* appeared in only about 5 percent of the tweets. However, in a batch from 2:25p on 12/23/11, *jazz* was found in 12 percent of all tweets. This was the highest percentage that *jazz* ever reached during the data collection period. In other words, this was the most the term *jazz* ever dominated the conversation in the Twitter-verse during the period of observation. Similar peak batches were identified for each of the terms listed above.

From there, the data from each particular batch was then used to train the creative agent, or learner, described in Section 3. Once trained, the learner was then asked to generate some output. In turn, its output was used as input to itself to generate new output. Both the training set data and the output of the learner were sonified using parameterized MIDI messages.

While the actual work of refining the sonification of the data is somewhat still a work in progress, some examples of strategies currently being applied to the training set include:

- mapping a *term* to pitch
- mapping how frequently a *term* is used to MIDI velocity
- mapping an artist's longitudinal location to a place in the stereo field
- mapping the importance of a *term*'s occurrence to note duration

Since the output of the learner does not share the same attributes as the ICAD dataset, slightly different sonification strategies are used in this case. Some of these are:

- mapping a *term* to pitch
- mapping relative reward score to MIDI velocity
- mapping relative differentiation of reward scores for candidate pitches to note duration

Additionally, some random elements have been introduced in an effort to mitigate some of the unnatural effects arising unintentionally from unobserved and/or uncontrolled musical parameters.

5 Conclusions

While the process has proved quite interesting, the resulting sonifications to date have been somewhat unsatisfactory. In retrospect, a couple of reasons for this are somewhat obvious: one having to do with the mining of the data and the other with the approach to the actual sonification. Potential future efforts will likely attempt to improve in both of these areas.

First, the data is not being derived from a musical process. While this is not a prerequisite for a sonification project, the causal relationships in the dataset (at least using the approaches outlined above) are somewhat obscured making it difficult to differentiate the sonification from noise. To one who is familiar with the learner, it is possible to see the learning and creative process work. And, further investigation shows that the learner is performing as expected.

But in order to produce more coherent or musical output from this dataset, it is likely that it will need to be viewed from an artist-centric perspective instead of a term-centric perspective. The appearance of an artist on a TV show, or an album release has a quantifiable effect on the frequency of an artist appearing in the dataset. There really is not correlate for a *term*. Thus, more coherent output can potentially be derived by focusing on artists instead of *terms*.

Secondly, constraining the sonification to a MIDI realization has continually proved somewhat frustrating. While a number of parameters can potentially be controlled via MIDI, it is simply not well suited to the more sophisticated types of transformations commonly used by contemporary electro-acoustic composers. Even though a great deal was learned regarding the programmatic generation and manipulation of MIDI messages during the course of the project, this approach will likely be avoided in the future as it has, thus far, led to undesirable results.

In all, this project has proved both interesting and useful. A real world test has been applied successfully to an ongoing project. This has helped to vet the processes employed in that parallel project. Additionally, the opportunity to explore an interesting dataset allowed for the discovery of some new ideas regarding sonification. The work here is certainly yet complete, but the foundation laid this semester has proved quite stable and should serve future endeavors well.

6 References

References

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

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

[3] Smith, B., and Garnett, G. Art in music: Fuzzy adaptive resonance theory for real-time musical knowledge extraction. *unpublished* (2012), 1.