The Sparsity of Simple Recurrent Networks in Musical Structure Learning

Kat R. Agres (kra9@cornell.edu)
Department of Psychology, Cornell University, 211 Uris Hall
Ithaca, NY 14853 USA

Jordan E. DeLong (jed245@cornell.edu)
Department of Psychology, Cornell University, 211 Uris Hall
Ithaca, NY 14853 USA

Michael Spivey (spivey@ucmerced.edu)
School of Social Sciences, Humanities, and Arts, UC Merced, P.O. Box 2039
Merced, CA 95344 USA

Abstract

Evidence suggests that sparse coding allows for a more efficient and effective way to distill structural information about the environment. Our simple recurrent network has demonstrated the same to be true of learning musical structure. Two experiments are presented that examine the learning trajectory of a simple recurrent network exposed to musical input. Both experiments compare the network's internal representations to behavioral data: Listeners rate the network's own novel musical output from different points along the learning trajectory. The first study focused on learning the tonal relationships inherent in five simple melodies. The developmental trajectory of the network was studied by examining sparseness of the hidden layer activations and the sophistication of the network's compositions. The second study used more complex musical input and focused on both tonal and rhythmic relationships in music. We found that increasing sparseness of the hidden layer activations strongly correlated with the increasing sophistication of the network's output. Interestingly, sparseness was not *programmed* into the network; this property simply arose from learning the musical input. We argue that sparseness underlies the network's success: It is the mechanism through which musical characteristics are learned and distilled, and facilitates the network's ability to produce more complex and stylistic novel compositions over time.

Keywords: Musical structure; Simple Recurrent Network; Sparsity.

Introduction

Work in the field of neural network modeling has been useful in creating simulations of functional machinations of human cognition and behavior. While many different architectures and learning algorithms exist, this paper will primarily focus on Elman's Simple Recurrent Network (SRN) (1990), which was originally developed to process and predict the appearance of sequentially ordered stimuli. This feature makes the SRN a prime candidate for processing the structure of music.

Modeling aspects of musical composition has shown that networks can be trained to 'compose' music after learning from many examples. One such network is Mozer's CONCERT, which is a modified Elman network that is trained on input stimuli and attempts to extract two key features: which notes in the scale are musically appropriate, and which of those selected notes is the best stylistically. While ratings of this network were better than compositions chosen from a transition table, they still were "compositions only their mother could love" (Mozer, 1994).

Other approaches have included aspects such as evolutionary algorithms (Todd, 1999) as well as utilizing self-organizing networks instead of relying on learning rules (Page, 1993). While most studies have concentrated on the success of these networks' compositions, the studies in this paper will concentrate on the internal state of the network as it learns. Additionally, subjects' ratings of the network's compositions over time will be examined, as well as other network statistics, such as sparse coding.

Sparse coding is a strategy in which a population of neurons completely encode a stimulus using a low number of active units. Taken to an extreme, this strategy is similar to the concept of a 'Grandmother Cell' that responds robustly to only one stimulus, and thus has a very low average firing rate. This is directly in contrast to a fully distributed system where every neuron takes part in encoding every stimulus and fires an average of half of the time.

Sparse coding allows for the possibility that as a distributed system learns the structure of the world, it begins encoding in a more sparse and efficient manner. The benefits of sparse coding have been reviewed in depth (Field, 1994; Olshausen and Field, 2004), however this paper will concentrate on two of them. The first reason is that encoding stimuli using fewer neurons allows for a complete representation without the biological demands of having every neuron firing (Levy, 1996). The second reason, which is highlighted in these studies, is that a sparse code develops in order to efficiently mirror the structure of the world.

By examining the neural network architecture over the learning trajectory, we can investigate how network sparsity changes with experience. Given the conventions of Western tonality in music (e.g. common chord progressions), as outlined by music theory, the progression of tones in music

obeys rules and patterns. These standard transitions impose order; notes do not skip randomly around the musical state space. When a SRN receives this structured musical input, it learns how best to efficiently code the information therein.

The developing internal structure of the network is of prime concern, but of equal importance is how the network's output reflects that internally changing structure. For external validation of the network's ability to produce increasingly stylistic output over training, listeners were recruited to rate the sophistication of the network's novel compositions. This external evaluation confirms the network's internal measures of sparsity and learning.

Experiment 1

In this study, we tested how a Simple Recurrent Network learns tonal structure over time by asking: What internal changes occur in order to produce increasingly more sophisticated compositions? This experiment explores how a SRN learns to predict the next note in a musical sequence by looking at the sparsity of its hidden layer activations. To elucidate the relationship between sparsity and the sophistication (complexity and style) of the network's compositions, participants rated the novel compositions from several points along the learning trajectory. We hypothesize that the sparsity of the network will increase as it is trained, and that subject ratings will similarly increase.

Method

Network Architecture

Matlab software was used to program and run the SRN. The network was given one note at a time during training; it learned musical structure by predicting the next note in the sequence, and then compared its prediction with the actual next note in the training melody. The error signal (difference between predicted and actual) was then backpropogated through the network.

The network was trained on five simple, 8-measure long melodies composed specifically for this study (see Figure 1). They were monophonic, of a piano timbre, and contained no rhythmic variation (all of the tones were quarter notes). Notes were held at equal duration in order investigate the probabilistic distribution of tonal relationships during training.

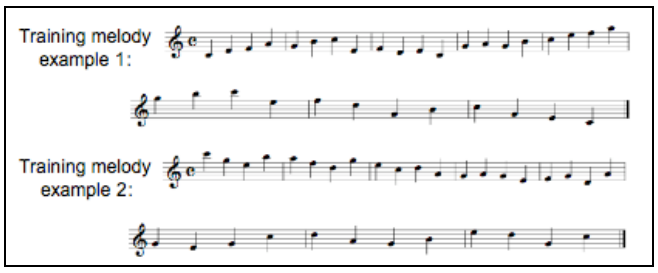

Figure 1: Examples of training melodies used as input.

The input and output layers of the network consisted of 15 nodes each, while the context and hidden layers contained 30 nodes (see Figure 2). The format of the input was such that one note (which was represented by turning on a corresponding node of the 15 present in the input layer) would be presented per timestep. For every timestep, the network predicted the next note in the training series, and each epoch of learning was comprised of 32 timesteps. The network randomly selected one of the five training melodies for every epoch. Hidden and output layer activations were transformed using a logistic function, 1/(1+e^(-x)), and varied between 0 and 1. Because the last note of one training melody is not musically related to the first note of the next training melody, the context layer activations were reset after each epoch of training.

Sparsity was measured in the hidden layer of each network by looking at the proportion of hidden layer nodes with an activation value greater than .3. These values were averaged over six iterations of the network, and were measured at 5, 25, 75, 150, 300 and 450 epochs.

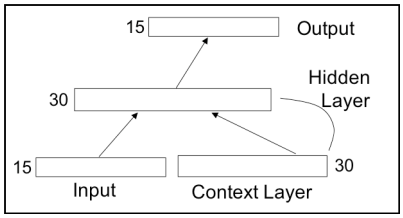

Figure 2: SRN architecture used in Experiment 1.

Behavioral study 1

External validation is required to draw any conclusions regarding the relationship between increasing sparsity over training and improvement in the quality of the network's compositions. Therefore, listeners rated ten sample compositions from epochs 5, 25, 75, 150, 300, and 450. These compositions were created by inputting the note 'Middle C' at each of these benchmark epochs. The network then predicted the next note, which was in turn fed back into the network as input. This method of sequence prediction is a strength of the SRN architecture, and has been used primarily to study grammatical aspects of language (Elman, 1991).

Participants Twenty Cornell undergraduates volunteered to participate in the experiment for extra credit in a psychology class. All participants had normal hearing, and had an average of 6.2 ± 3.7 years of musical training.

Materials After completing a particular number of epochs of training, sixteen notes of the network's compositional output were recorded. Ten examples were recorded from each level of training (5, 25, 75, 150, 300, or 450 epochs). Each compositional sample was manually transferred from Matlab to Finale, a music software program, and converted into .wav sound files. All compositions were set to a piano

timbre, and rhythm was kept constant (each tone was one quarter note in duration). Each trial consisted of a 16-note composition (four-measures in 4/4 time), and was 8 seconds in duration. The experiment was administered on a Dell Inspiron laptop running E-Prime software, and participants wore Bose Noise Canceling headphones set to a comfortable listening volume.

Procedure After reading the instructions, a brief practice session consisting of four trials preceded the experiment. No feedback was given during the practice or experimental trials; the practice session simply functioned to familiarize participants to the types of melodies they would be rating. The practice trials were drawn from different points along the learning trajectory, including 5, 75, 150, and 450 epochs, and were different from those included in the experiment. The sixty experimental trials were completed without interruption and presented in random order using E-Prime software. After listening to each trial, the listener rated the composition on a 'goodness' scale from 1 to 7, where '1' represented a "poor example of classical music" and '7' represented an "excellent example of classical music". Participants were urged to use the whole scale as they found appropriate.

Results and Discussion

Network Internal Structure

By examining the activations of the hidden layer at different stages along its learning trajectory, we see that sparsity increases over time. In other words, as the network completes more epochs of training, the internal structure of the hidden layer becomes more sparse (see Figure 3).

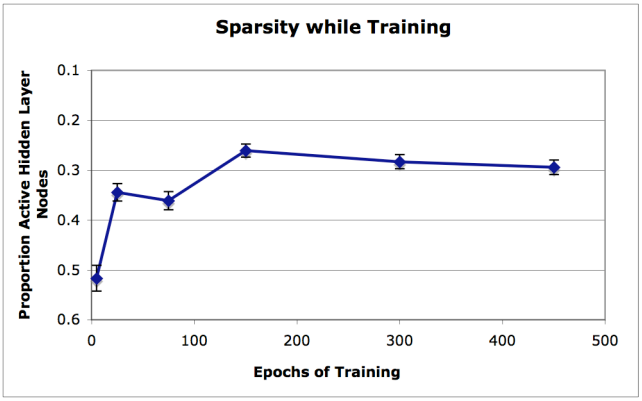

Figure 3: The proportion of active hidden layer nodes (sparsity) over the learning trajectory.

As shown above, during the early stages of the network's development, there is a dramatic increase in the sparsity of the hidden layer representations, as indicated by a reduction in the proportion of hidden nodes with activations greater than .3 (note inverted Y axis). Again, these values are derived by taking the average over six networks of the proportion of hidden activations above .3 (for each training epoch in question). After rapidly distilling structure from the training melodies, this decreasing trend begins to plateau around 150 epochs of training.

Behavioral study 1

To assess how well the internal measure of sparsity corresponds to the sophistication of the network's compositions, we tested whether sparsity was an informative predictor of listeners' goodness ratings. Indeed, listeners displayed a general preference for melodies produced after more epochs of training (see Figure 4).

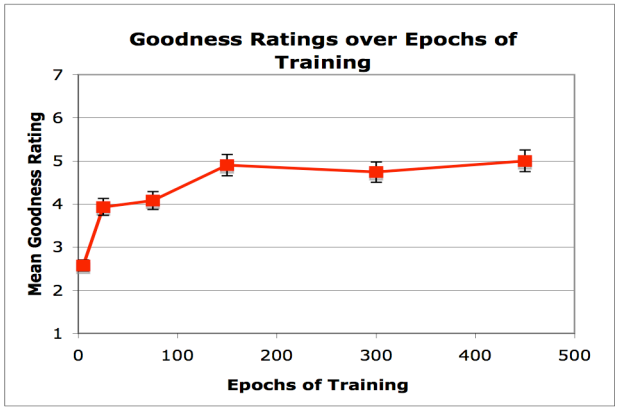

Figure 4: Average of listeners' goodness ratings over epochs of training.

Because the sparsity measurements and goodness ratings followed roughly the same trend over time, sparsity did prove to be an excellent predictor of how sophisticated the melodies sounded to listeners, $R^2 = .95$, $F = 84$, $p < .001$.

Experiment 2

The second experiment examines the same network structure as the first, but utilizes more complex input stimuli, many more training epochs, and employs a new sparsity metric. Three movements from J.S. Bach's Suite No.1 in G Major for Unaccompanied Violoncello were selected for the network's training input because they are musically complex and sophisticated, yet monophonic (there is a single, unaccompanied voice). The Prelude, Allemande, and Courante were chosen because they can all be performed at a similar tempo. These pieces are more complex than those used in the first experiment because each features different note durations and musical themes.

In addition to musical changes, a new sparsity metric was adopted from single-cell recording (Rolls and Tovee, 1995), in which the square of the mean is divided by the mean of the squares (Figure 5). While the metric used in Experiment 1 is mostly equivalent, the Rolls sparsity metric is used pervasively in the literature. Both the previous sparsity .3 criterion and the Rolls sparsity metric will be used to assess

the sparsity of the hidden layer activations in this experiment.

$$S = \frac{\left(\frac{1}{n}\sum_i r_i\right)^2}{\frac{1}{n}\sum_i r_i^2}$$

Figure 5: Equation for Rolls sparsity metric, where *n* is defined here to be the number of hidden layer nodes, and *r* is the rate of activation for each node.

Method

Network Architecture

The same basic SRN architecture from Experiment 1 was used in this study. Because of the increased complexity of the musical input, MIDI numbers and note durations were combined into the input for each timestep. This was encoded in the input and output by turning on one pitch node and one duration node per note. Duration values were represented by sixteen nodes, with each node being representative of a note duration ranging from a 16th note to a whole note. Due to this increase in complexity of the input (a larger pitch range and rhythmic information), the number of nodes in each layer was increased. The input and output layers now consist of 144 nodes (128 MIDI notes and 16 durations), and the hidden and context layers contain 64 nodes.

This same network architecture was used for two different training techniques. The Normal network was fed a 32-note sequence, randomly selected from one of the movements of Bach, for each epoch of training. A second network, the Bigram network, was also trained on 32 notes per epoch, but the sequence of notes lacked musical structure: After an initial note was randomly chosen from one of the movements of Bach, the network's predictions of the next note in the sequence were compared with the actual next note. Then, however, the Bigram network skipped to another random note within the musical corpus (thus, the network was only able to learn musical structure via a series of bigrams). This effectively limits the Bigram network's predictive capability to the note played immediately prior, thereby reducing the amount of structure the network is able to learn. Context layer activations were reset in both the Normal and Bigram networks after each training epoch.

A sample of the network's hidden layer was captured every 10 training epochs and used to measure the network's sparse structure. The entire network was captured at each level of training in order to compose novel melodies using sequence prediction as in Experiment 1.

Behavioral study 2

Participants Ten Cornell undergraduates volunteered to participate in the experiment for extra credit in a psychology class. All participants had normal hearing, and had an average of 2.4 ± 2.7 years of musical training.

Materials For each level of training tested (5, 50, 500, 5 thousand, 50 thousand, 500 thousand, and 5 million epochs), ten 32-note compositions were recorded for both the Normal and Bigram networks. Each compositional sample was manually transferred from Matlab to Finale and converted into a wav sound file. The compositions were all of a piano timbre, and the compositions' rhythmic variation was included. Because of the increased complexity of the musical material, each trial consisted of a 32 tones. Due to some variation in note duration, the trials were of slightly different lengths (average length = 12 sec). The experiment was administered on a Dell Inspiron laptop running E-Prime software, and participants wore Bose Noise Canceling headphones set to a comfortable listening volume.

Procedure The same procedural protocol was used as in the first study: After reading the instructions, a brief, four-trial practice session preceded the experiment. These practice trials included an example from 50, 5k, 500k, and 5m epochs, and were different from any test trials in the experiment. A total of 140 test trials were presented, with the 70 trials from the Normal network and 70 trials from the Bigram network combined into one large block of trials and presented in random order. Listeners rated each composition on a goodness scale from '1' to '7' as outlined for the first experiment.

Results and Discussion

Network Architecture

As predicted, the internal representations of both networks do become more sparse as the network learns structural relationships inherent in the music (see Figure 6). This pattern continues until roughly 1 million training epochs, even while adopting the alternative Rolls (1995) metric of sparsity.

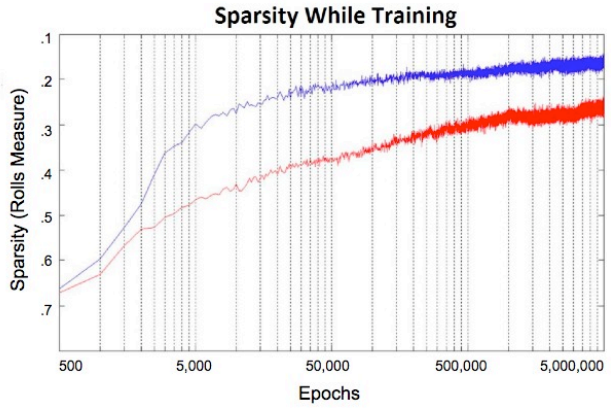

Figure 6: Rolls sparsity metric over epochs of training for the Normal (blue) and Bigram (red) networks.

The Normal network displays more sparsity in its hidden layer activations than the Bigram network. In order to shed

light on the nature of the hidden layer activations of the network while composing, sparsity was also examined while the network produced output. Both networks display an increase in sparsity at 5,000 epochs, but return to a less sparse state by 5 million epochs. Though both networks display similar degrees of sparsity, the Bigram network exhibited sparser coding during composition at 50,000 and 500,000 epochs (see Figure 7). The Bigram network also created simpler melodies than those of the Normal network. This is mainly due to the fact that while the Normal network is more efficient at encoding the stylistic structure from which it is trained, it has more difficulty encoding its own output during composition. The Bigram network does not have this limitation, as the structure it learns during training is similar to what it is capable of composing. In addition, the Mean Squared Error (MSE) of both networks decayed quickly and reached a plateau with little variation by 30,000 epochs of training. The Bigram network's MSE was slightly lower than that of the Normal network.

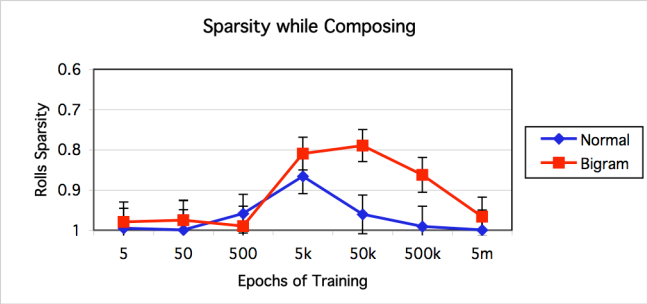

Figure 7: Rolls sparsity metric while composing after different amounts of training.

Behavioral study 2

Interestingly, the compositions of the Bigram network are better rated by participants than those of the Normal network, $R^2 = .95$, $F = 19.30$, $p < .01$, as shown below in Figure 8.

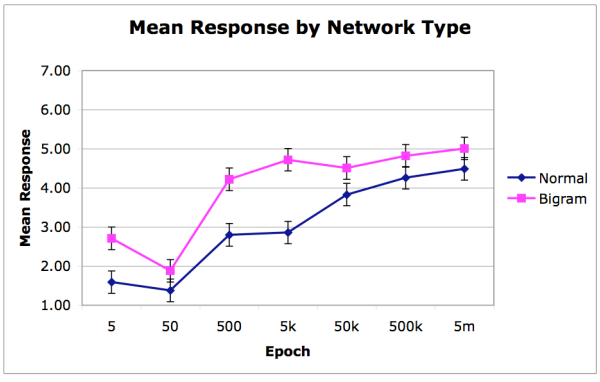

Figure 8. Participant mean response over epochs of training for the Normal and Bigram networks.

A comparison was made between the .3 criterion sparsity measure and the Rolls sparsity metric (from training) in predicting the behavioral data. The sparsity criterion was not a significant predictor of goodness ratings for the Normal network, $R^2 = .57$, $F = 3.93$, $p = .14$, but was significant for the Bigram network, $R^2 = .81$, $F = 12.65$, $p < .05$. The Rolls sparsity metric was performed similarly: It was not a significant predictor of ratings for the Normal network, $R^2 = .62$, $F = 4.87$, $p = .11$, but was significant for the Bigram network, $R^2 = .77$, $F = 9.99$, $p = .05$.

General Discussion

Examining the way in which neural networks learn musical structure can point to ways in which humans learn music. In both the human cortex and neural network models, a distributed, sparse structure appears to be an optimal way to encode musical information.

In comparing the Normal and Bigram data, both networks displayed increasingly sparse internal representations over their developmental trajectory. Listeners' ratings follow a general increase that corresponds with the amount of training that a network has received as well as the sparsity of the network's hidden layer while learning. While we expected that subject ratings would increase with training, the fact that sparsity also increased with training shows that the learning algorithm of the networks picked up sparse structure in the input. While many models attempt to build sparsity into their network, sparse coding simply arises in these networks as they learn.

The structure of music may in fact lend itself to sparse coding. Of the vast number of notes that could be used to compose a musical work, only a subset of them are selected given the harmonic structure from which the tonal relationships are determined. In other words, tonality has a hierarchical structure, and its foundation is centered around a particular group of tones. This inherent organization can be optimally encoded with a sufficient amount of training.

The Normal and Bigram networks from Experiment 2 show the difference in hidden layer sparsity that results from differing amounts of structure in the network's input. The Bigram network did exhibit less sparsity while training, a hallmark of less structure in the signal (because transitional relationships *between* bigrams were random). While the Normal network is more sparse during training, the Bigram network interestingly shows more sparsity during some stages of composition, and receives better ratings overall. This may be because while the Normal network has a more sparse representation during training, it is more likely than the Bigram network to enter into a repetitive series of notes while composing (such as the tonic triad) because it was trained on melodies with a longer musical context (it can utilize information from more previous timesteps when training).

There are many possible directions for future study. For example, follow-up experiments can implement more recent advances in recurrent neural network architectures that encode for time information in different ways. Some of the

newer models used to generate and predict musical output are Long Short Term Memory networks (Eck & Schmidhuber, 2002) and Echo State Networks (Jaeger, 2001). Additionally, the network could use an interval-based representation rather than a pitch-based representation to examine whether differences in learning and composition would arise.

Future iterations of this study will also examine to what extent the network over-learns the training music. Overfitting could be investigated by testing how quickly the network can learn a novel melody after various amounts of training. Also to this end, participants could rate how similar the network compositions were to the training music. It is possible that differing levels of musical training between participants in Experiment 1 and Experiment 2 contributed to different rating strategies for the compositions. A t-test comparing the participants' training across the two studies demonstrated a significant difference in musical training, $t = -3.08$, $p < .01$. Because this may have contributed to rating differences, musical training will be controlled in future work.

Furthermore, continuing to explore the different internal characteristics of a network that is composing versus one that is learning may yield interesting results. The counterintuitive fact that the Bigram network in the second study exhibited greater sparsity and higher subject ratings shows that the process of composition in a SRN may be more multifaceted than previously appreciated. When a network feeds itself its own output during composition, the inherent complexity of the recurrent loop generates highly variable output that warrants further investigation.

Acknowledgments

We would like to thank Professor David Field for his helpful advice regarding the measure of sparsity. Also, we wish to thank our Christine Lee for her assistance in formatting stimuli and running participants in the second study.

References

Eck, D., & Schmidhuber, J. (2002). A First Look at Music Composition using LSTM Recurrent Neural Networks. *Technical Report IDSIA-07-02*, Instituto Dalle Molle di studi sull intelligenza artificiale, Manno, Switzerland.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science, 14(2)*, 179-211.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning, 7*,195-224.

Field DJ. (1994). What is the Goal of Sensory Coding? *Neural Computation, 6,* 559-601.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks. In *GMD Report 148, German National Research Center for Information Technology*, 2001.

Lerdahl, F., & Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. Cambridge, MA: MIT Press.

Levy W.B., Baxter R.A. (1996). Energy efficient neural codes. *Neural Computation, 8,* 531-543.

Mozer, M. (1994). Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multi-scale Processing. *Connection Science, 6,* 247-280.

Olshausen B.A., and Field D.J. (2004). Sparse Coding of Sensory Inputs. *Current Opinion in Neurobiology*, *14*, 481-487.

Page, M.P.A. (1993). Modeling Aspects of Music Perception Using Self-organizing Neural Networks. Unpublished doctoral dissertation. University of Wales.

Rolls, E.T. and Tovee, M.J. (1995). Sparseness of the neuronal representation of stimuli in the primate temporal visual cortex. *Journal of Neurophysiology*, *73(2)*, 713-726

Todd, P.M. (1989). A connectionist approach to algorithmic composition. *Computer Music Journal*, *13(4)*, 27-43

Todd, P.M. (1999). Evolving musical diversity. In *Proceedings of the AISB'99 Symposium on Creative Evolutionary Systems*, 40-48. Sussex, UK: Society for the Study of Artificial Intelligence and Simulation of Behavior.