Using the Universal Similarity Metric to Model Artificial Creativity and Predict Human Listeners Response to Evolutionary Music

Nils Svangård[1], Jon Klein[1,2] and Peter Nordin[1]

[1]Physical Resource Theory,
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg, Sweden
[2]Hampshire College
Amherst, MA 01002
nils@svangard.org

Abstract

In this paper we present a new technique for modeling Artificial Creativity in Evolutionary Music (EM) systems and predicting how appealing musical pieces are to human listeners. We use a k-Nearest Neighbor classifier where we approximate the Information Distance between the new, unclassified, musical piece and a corpus of observed musical pieces rated by the user with the Universal Similarity Metric. We approximate the Information Distance with two different methods, using standard binary compression of MIDI files, and using MP3 encoding of raw audio streams.

Our experiments indicate that the universal similarity metric can be used to discriminate between music that do and do not appeal to human listeners. Even though classification results is not perfect, it performs significantly better than the random baseline and when we combine the predictions made independently by the MIDI and MP3 classifiers, we obtain an even higher classification accuracy, ranging up to 77% on the test set. These results is in the same range as our results in predicting the aesthetic value of visual art, which indicates that the Universal Similarity Metric is a very general and versatile approach to modeling Artificial Creativity.

Introduction

Creativity is usually associated with the ability to create art, and the study of Artificial Creativity can provide significant insights into both Artificial Intelligence and Life. However, the vast majority of computational systems developed in the past few years for generating artwork require human interaction to be creative. They have neither the ability to perceive the artworks they, or other artists produce, nor are they able to perform aesthetic judgments during their creative process. In other words, they are completely blind and deaf to the outside world.

This is a well-known limitation and several attempts to solve it have been made, in both visual arts and music. In particular Rob Saunders and John Gero (2001) gives a good introduction and motivation to the study of Artificial Creativity, and they use Artificial Neural Networks (ANN) to measure the aesthetic value of computer-generated images. Other similar attempts had been made earlier by Shumeet Baluja et al. (1994), who trained a set of artificial neural networks with low resolution images as input in order to predict whether an image looks good or not. However, both had limited success with their predictions and their systems did not generalize well. Lately, Penousal Machado et al. (1998) have made other attempts using "expert knowledge" about the artworks to make aesthetic judgments with better results, even though they where unable to guide the process of an evolutionary artist using their measures.

There have also been numerous attempts in the domain of evolutionary music. For example Spector and Alpern (1994, 1995) used a combination of Genetic Programming to evolve Jazz tunes, and an Artificial Neural Network trained on combinations of Charlie Parker's as a critic, with promising, but somewhat "unsatisfactory" results. Recently Machado et al. (2003) has developed a framework for Artificial Art Critics using feature extraction, evaluators and validation processes as a foundation for further research in this area. In their paper they raise an interesting issue, stating that:

> The amount of information contained in some artworks is huge. In visual arts, for instance, even a relatively small picture can fill a lot of memory. Taking into account the current state-of-the-art in adaptive systems (e.g., neural networks, genetic algorithms) it is clear that these techniques cannot currently handle such vast amounts of information.

They also empathize the point that an Artificial Art Critics should be general enough to easily adapt to new domains and should base it's assumptions on only the artwork itself, no other information about the artwork should be required. However, the need for generality clashes with the need to handle the particularities of each domain, which requires specialized techniques for each domain. Moreover, to the best of our knowledge there has not been any single method presented that theoretically can understand and model the aesthetic values of all possible forms of art.

However, there are new general techniques that require no expert knowledge that seem promising; recently there have been great advances in the mathematical theory of similarity where a new universal metric for calculating the similarity of any two objects has emerged. We have successfully applied this technique to the visual arts domain (Svangård et al. 2004), where we successfully used it to predict what images produced by an Evolutionary Art system would be best appreciated by the observer (having the highest "aesthetical fitness"). The Universal Similarity Metric has previously been used to categorize MIDI music (Cilibrasi et al. 2003), where it was successfully used to distinguish between music genres and could even cluster pieces by composer.

Instead of requiring detailed knowledge about the objects, and problem domain, to compute the similarity, this metric can detect all similarities between the objects that any other effective metrics could detect. This metric was developed by Li et al. in (Li et al. 2001, 2002 & 2003) and is based on the "information distance" described in (Li and Vitányi 1997) and (Bennet et al. 1998). Roughly speaking, two objects are said to be similar if we can significantly compress one given the information in the other, the idea being that if two objects are more similar then we can more easily describe one given the other. In this context, compression is based on the ideal mathematical notion of Kolmogorov complexity, which unfortunately is not effectively computable, but can be approximated with good results using existing compression algorithms.

The idea we present in this paper is to use the universal similarity metric to compute how similar pieces produced by an Evolutionary Music (EM) system is to a corpus of musical pieces previously rated by how good it sound to the listener. Then we use this information to predict how the listeners will respond to the new piece, i.e. whether they will like it or not.

We will approximate the Information Distance using both compression of the MIDI files produced by the EM system, similar to what Cilibrasi et al. did, but also using MP3 compression of the raw audio streams produced when the MIDI files is played. Using MP3 encoding to compress the songs is particularly interesting, compared to using a standard binary compressor, since it is designed to exploit known limitations of the human ear, notably the fact that there are certain harmonics in audio signals that humans cannot perceive, as well as harmonics that obscure the presence of other frequencies. Thus, MP3 is intended for compressing music that will be listened to by humans, which is exactly what we try to model.

This paper is organized as follows; the first section will give a brief introduction to our Evolutionary Music system, followed by a section that give a detailed introduction to Kolmogorov complexity and the Universal Similarity Metric. Finally, we discuss our implementation and experiments as well as an analysis of our results.

Evolutionary Music

A major obstacle to constructing interactive evolutionary music is user interaction. Especially as pieces become longer and longer, and as it takes more and more generations to produce appealing results, it becomes unreasonable to expect that users will want to patiently listen and rate music through an entire evolutionary process. This is another area where the study of Artificial Creativity could be used to speed up the interactive process and only present the user with new, interesting pieces removing those who will sound bad based on the users previous actions.

Another way that our system, "iEvolve", tries to address this problem is by allowing the user to rate music at their own pace, using software and hardware they may already have, exactly as the software and hardware are intended to be used. We thus take advantage of an existing, user-friendly process of rating music, and use it to drive the creation of evolutionary music. We do so by tapping into Apple's popular iPod music player and the freely available iTunes jukebox software. The iPod and its software companion iTunes allow users to rate songs on a scale from one to five stars. If the ratings are made using an iPod, the rating information is automatically synchronized when the iPod is plugged in. This rating information is stored in the iTunes database and can be accessed by the user to, among other things, sort the music and create playlists.

iEvolve works by automatically retrieving rating information from the iTunes database, using this information to breed and mutate music and then to upload the new music into the user's iTunes library and iPod. From the user's perspective, the entire process is completely transparent. They simply use their iPod or iTunes software to listen to and rank the music, just as they might do with any other non-evolving music.

The software relies on a hodgepodge of different technologies provided by the Mac OS X operating system in order to make the evolutionary process completely transparent to the listener. Because there are several steps required, each requiring access to different technologies, iEvolve is implemented as a series of simple programs and scripts that are connected together by a Perl script. These individual programs are described below, in the order in which they are called from the Perl script:

GetRatings.scpt: an AppleScript that retrieves ratings information from the iTunes library.

Trainer: a script that stores the observations about the musical pieces and their rating in a database for later use by our Artificial Art Critic.

Breeder: a C program which loads in a set of raw-data genomes and the current song ratings. The song ratings are then used to drive the genetic operators and to generate the next generation of genomes.

Translate: a C program which translates the raw-data genomes into their corresponding phenotypes, in the form of MIDI files. We have

experimented with several different genome-to-MIDI translation schemes.
Miditoaiff: a C program which uses QuickTime to convert MIDI files into the AIFF files that the iPod can play.
Classifier: a script that takes a set of genomes, MIDI and AIFF files and makes a prediction how well the user will like them based on the previous observed rating of musical pieces.
UpdateMusic.scpt: an AppleScript that removes the previous generation of songs from the iTunes library and replaces them with a new generation. If an iPod is connected to the machine, it also synchronizes the new music onto the iPod.

We use a simple Genetic Algorithm (GA) that converts the genomes into MIDI files. It is a simple byte-code language interpreted by a register based virtual machine that supports the basic algebraic operations, as well as some mathematical functions (sin, cos, exp, log), conditional statements, and loops, but the two most important features is that it can create tracks in the MIDI file, and play notes. The purpose of this paper is not to go into detail how this system works, but a good introduction to evolving music using GA is (Papadopoulos 1998).

Aesthetic Selection

The question about how aesthetic selection work is in hot debate, and has been for hundreds of years, but recently there have appeared a number of interesting theories, and computer models using machine learning, explaining how it might work. One of the most popular way to approach this problem seem to be using prior knowledge of images and genomes that produce visually interesting images. Since most genetic artists usually work with a large quantity of genomes that looks good to seed new runs, there exist plenty of such information. Unfortunately there's no given method to assign aesthetic fitness values to genomes based on this information, and most techniques for classification, such as artificial neural networks or statistical analysis, usually requires expert knowledge about the features of the problem and does not provide any universal method for solving this problem.

However, since the introduction of the universal similarity metric there is a very powerful tool that can be applied to this tasks without any prior knowledge about aesthetics and which conforms to the principle of Occam's Razor; "Less is more."

Kolmogorov Complexity

Given an object, x, that can be encoded as a string over a finite alphabet, e.g. the binary alphabet, the Kolmogorov complexity, $K(x)$, is the length of the shortest compressed binary version from which x can be fully reproduced. Here "shortest" means the minimum taken over every possible decompression program, both real and imaginary. In fact, there does not even have to be a program that can compress the original object to the compressed form, but if there is one so much the better.

Technically the definition of Kolmogorov complexity is as follows: First, we fix a syntax for expressing all computable functions. The usual form is as an enumeration of all Turing machines, but an enumeration of all syntactically correct programs in some universal programming language like Java, Lisp or C is also possible. Then we define the Kolmogorov complexity of a finite binary string as the length of the shortest Turing machine that can generate it in our chosen syntax. Which syntax we use is not important, but we have to use the same syntax in all calculations. With this definition, the Kolmogorov complexity of any finite string will be a definite positive integer.

Even though Kolmogorov complexity is defined in terms of a particular machine model, it is actually machine-independent up to an additive constant. This means it is asymptotically universal and absolute through Church's thesis, and from the ability that universal machines can simulate one another and execute any effective process. The Kolmogorov complexity of an object can then be seen as an absolute and objective quantification of the amount of information it contains.

So $K(x)$ gives the length of the ultimately compressed version x^* of x which can be thought of as the amount of information, in number of bits, contained in the string. Similarly, $K(x|y)$ is the minimal number of bits required to reconstruct x given y. In a way $K(x)$ expressed the minimum amount of information required to communicate x when the sender and receiver has no knowledge where x comes from. For more information on individual information content, see (Li and Vitányi 1997).

The Similarity Metric

As mentioned our approach is based on a new and very general similarity distance that can categorize objects depending on how much information they share. In mathematics, there are many different types of distances, but they are usually required to be 'metric' in order to avoid undesired effects. A metric is a distance function, $D(\cdot,\cdot)$, that assigns a non-negative distance, $D(a,b)$, to any two objects a and b under the following conditions:

1. $D(a,b) = 0$ only when $a = b$
2. $D(a,b) = D(b,a)$ (symmetry)
3. $D(a,b) \leq D(a,c)+D(c,b)$ (triangle inequality)

In (Li et al. 2003) a new theoretical approach to a wide class of similarity metrics was proposed: the "normalized information distance." This distance is a metric and universal in the sense that this single metric uncovers all similarities simultaneously that the any metric in the class

uncovers separately. This can be understood in the sense that if two objects are similar (that is, close) according to a particular feature described by a particular metric, they are also similar in the sense of the normalized information distance metric, which justifies calling the latter *the* similarity metric. Oblivious to the problem area concerned, simply using the distance according to the similarity metric can be used to automatically classify objects of any kind. Mathematically the similarity metric is defined as the distance of any pair of objects, x and y, where:

$$d(x,y) = \frac{\max\{K(x|y),K(y|x)\}}{\max\{K(x),K(y)\}}$$

It is clear that $d(x,y)$ is symmetric and in (Li et al. 2003) it is shown to be metric. It is also universal in the sense that every metric expressing some similarity that can be computed from the objects concerned is comprised in $d(x,y)$.

To compute the conditional measure, $K(x|y)$, we use a sophisticated theorem from (Li and Vitányi 1997) known as the "symmetry of algorithmic information":

$$K(x|y) = K(xy) - K(x)$$

So in order to compute the conditional complexity, $K(x|y)$, we can just take the difference of the unconditional complexities, $K(xy)$ and $K(x)$, which allows us to easily approximate $K(x|y)$ for any pair of objects.

Distance Based Classification

Given a set of objects with unknown aesthetic fitness, and a corpus of objects that we known are attractive to the user, we have to figure out which of the new genomes the user will find most interesting. In order to do this we have to figure out which objects shares the most information with the library of prior knowledge using the similarity metric. One method that is suited for this kind of classification tasks, which also happens to be one of the simplest ones, is k-Nearest Neighbors (kNN) (Mitchell 1997).

The kNN algorithm works by first calculating the distance, $D(a,b)$, between the object we want to classify, x, and all observations in the corpus. It then takes the 'k' observations nearest to x and sums the weight for each class and observation, $W(a,b)$. The class with the highest weight is thus the class most likely to be the correct guess for x.

Method

Since aesthetic selection is probably as far from an objective system you can get, there is no given way to measure how well our method really work. However, we wanted to get some statistical indication on the predictive power of our system so we decided to test the classification accuracy of our system by letting the user label a new set of musical pieces and then having our system trying to predict which were the most interesting ones.

First, we let our test subject use our application to build a database of pieces (s)he find interesting. Then we test the system by repeatedly presenting the user with two new pieces and the user has to pick the one he thinks sounds best. Afterwards we let our system process all such 2-tuples of tunes and predict which sounded the best, and with the users choice as a label we can evaluate the classification accuracy of our system as a binary classification task.

Since the tunes are categorized in six numeric classes, corresponding to the rating of how good they sound, we use a slightly modified version of the kNN classifier to predict which sounds best. Instead of picking the class having the most weight, as you usually do, we calculate the average class by multiplying the weight of the class with the numeric value and dividing by the total weight for all observations. This gives us a less discreet prediction that lets us consider the relation to all classes.

To calculate the information distance between the MIDI files we use LZ77 (a.k.a. Lempel-Ziv, as used in gzip), and to compute the conditional complexities we simply concatenate the two MIDI files and compress it. For the AIFF files we use a combination of MP3 encoding, with compression set to max, using Variable Bit Rate (VBR) and psycho-acoustic models, followed by LZ77 to get the maximum compression of the files. To compute the conditional complexities of the audio files we create a new AIFF file with correct headers and concatenate all channels and frames from the two files, and then use the MP3 encoding scheme as before.

One drawback with MP3 compression is that it is lossy, compared to LZ77, which is not, and the approximation of Kolmogorov complexity requires the compression algorithm to be able to completely reproduce the object. However, this only manifests itself as a linear error term in the model, which we think is a reasonable cost for the drastically improved compression ratios. After all, the MP3 algorithm is designed to remove only the frequencies the human ear cannot perceive, so in a way it is lossless to the listener even though the binary representation is not identical.

Results

In order to test our system we carried out two independent experiments, one where we use genomes with 256 genes, and one with 1024 genes. In both experiments we use 'zlib' for the LZ77 compression, and 'LAME' for the MP3 encoding[1]. Table 1 shows the number of observations in

[1] LAME is an LGPL MP3 encoder: http://lame.sourceforge.net/

each class for the two experiments, and the number of genomes in the test set.

Rating (Class)	256 genes	1024 genes
0%	49	33
20% ★	8	18
40% ★★	8	15
60% ★★★	10	8
80% ★★★★	4	11
100% ★★★★★	1	3
Total:	80	88
Test set:	22	11

Table 1 The number of observed musical pieces for the different ratings in the training corpus for both experiments and the total count of observations in the training and test sets

In both experiments we use our improved kNN classifier (with k set to 21) to guess which piece sounded the best to the user with our two distance measures, d_{MIDI} for compression of the MIDI phenomes, and d_{MP3} for compression of the audio genomes. The weight function was simply one minus the distance between the two objects, since the distance is in the range [0,1] and with our limited training set we wanted the distance between objects to count more than the number of observations. Table 2 shows the classification accuracy we obtained using these parameters.

Classifier	256 genes	1024 genes
d_{MIDI}	59%	64%
d_{MP3}	68%	64%

Table 2 The accuracy of the kNN-classifier using different distance measure when predicting which musical piece sounded best to the user

We noticed that the MIDI and MP3 classifiers did not give the same answer for all the examples in the test set, so we believed we could improve the classification performance by using the information from both of them simultaneously. In Table 3 you can see the *overlap* of the two classifiers, which simply is the number of times both classifiers gave the same answer divided by the number of all answers. This led us to create a new distance measure, d_{both}, which is simply the length of the vector (d_{MIDI}, d_{MP3}):

$$d_{both}(x,y) = \sqrt{d_{MIDI}(x,y)^2 + d_{MP3}(x,y)^2}$$

The classification results using this measure can also be found in Table 3, and it is apparent that this measure significantly improved our classification performance.

The performance of the musical classification is in the same range as when we applied it to estimating image aesthetics, where we had 75% accuracy when using JPEG compression of the images and 65% when we compressed genomes using LZ77 (Svangård et al. 2004). That system was designed a little differently though, since we computed the conditional probability between the test object and a concatenation of all the training objects, but in one experiment similar to this kNN approach we calculated the mean distance to all training genomes we got an accuracy of 59%, so it seems like our methods perform equally well on both visual art and music.

Classifier	256 genes	1024 genes
Overlap:	55%	45%
d_{both}	77%	73%

Table 3 The classification results obtained when combining the MIDI and the MP3 distance into a single distance measure. Overlap represents the percentage of independent answers that where identical between the MIDI and AIFF classifiers, while d_{both} represents the accuracy our classifier had when we considered both the MIDI and AIFF classification simultaneously

Future Work

The classifier based on the MIDI compression is very fast. Every distance measure usually takes less than three milliseconds on a modern computer and with our training corpus it takes less than a second to classify an unknown musical piece. The MP3 compression however, is as expected much slower; a single distance measure can take anything from 1 second up to several minutes depending on how long the piece lasts, and classifying an unknown genome usually takes over an hour. Therefore, the MP3 distance measure have to be optimized to be feasible to use in this context, in particular when we are going to work with larger example sets. Either by finding a better compression algorithm, smarter kNN algorithms that prune some of the observations, or in worst case parallelizing the algorithm and using clusters for the computations.

One of the first extensions to this system that springs to mind is completely automatic evolution without any user interaction (short of listening to the final piece). Therefore, instead of letting the user choose which piece to pick as parents as the next generation, we would take those that our aesthetic system believes the user would have picked. However, previous attempts using Artificial Art Critics that performed well in theory did seem to have some trouble when used to control the evolution autonomously, so this probably requires some further improvements to the model to be made.

Our experiments had a quite small training and test set, so in order to get better statistical significance how well this method works we need to make larger experiments, using many test subjects. In addition to this, one interesting extension would be to use a large library of previously rated real music instead of our evolutionary produced tunes. Since our method is general, there is no real limitation to what kind of music you could use.

Conclusion

Our experiments indicate that the universal similarity metric can be used to successfully discriminate between music that do and do not appeal to human listeners. Even though classification results is not perfect, it performs significantly better than the random baseline and in the same range as our results in predicting the aesthetic value of visual art.

The best distance measure we used seems to be the MP3 information distance, having an average accuracy of over 64% on both our test sets, but when combined with the MIDI distance the accuracy is improved by almost 10%.

We have shown that the Universal Similarity Metric can be used for Artificial Art Critics in both visual arts and for music. Even though it can be somewhat computationally expensive, the general nature of this method, and that it doesn't require any particular expert knowledge about the target domain, makes it a very promising technique to model Artificial Creativity and will be investigated further.

References

Baluja, S., Pomerlau, D. and Todd, J., 1994: Towards Automated Artificial Evolution for Computer-Generated Images. Connection Science, 6 (1994), 325-354

Bennet, C., Gács, P., Li, M., Vitányi, P., Zurek, W., 1998: Information Distance, IEEE Transactions on Information Theory, 44:4(1998), 1407-1423

Cilibrasi, R., de Wolf, R., Vitanyi, P., 2003: Algorithmic clustering of music. http://arxiv.org/archive/cs/0303025

Li, M., Vitányi, P., 1997: An Introduction To Kolmogorov Complexity and its Applications, Springer-Verlag, New York, 2nd Edition (1997)

Li, M., Badger, J. H., Chen, X., Kwong, S., Kearney, P., Zhang, H., 2001: An information-based sequence distance and its application to whole mitochondrial genome phylogeny, Bioinformatics, 17:2(2001), 149-154

Li, M., Vitányi, P., 2002: Algorithmic Complexity, International Encyclopedia of the Social & Behavioral Sciences, Smelser, N, Baltes, P., Eds., Pergamon, Oxford (2001/2002), 376-382

Li., M., Chen, X., Li, X., Ma, B., Vitányi, P., 2003: The Similarity Metric, in proceedings at the 14th ACM-SIAM Symposium on Discrete Algorithms (2003)

Machado, P., Cardoso, A., 1998: Computing Aesthetics. In: Oliveira, F. (Ed.), Procs. XIVth Brazilian Symposium on Artificial Intelligence SBIA'98, Porto Alegre, Brazil, Springer-Verlag, LNAI Series (1998), 219-229, ISBN 3-540-65190-X

Machado, P.,. Romero, J,. Manaris, B,. Santos, A., and Cardoso, A., 2003: "Power to the Critics – A Framework for the Development of Artificial Critics," in Proceedings of 3rd Workshop on Creative Systems, 18th International Joint Conference on Artificial Intelligence (IJCAI 2003), Acapulco, Mexico (2003), 55-64

Mitchell, T.M., 1997: Machine Learning, Boston MA: McGraw-Hill, 1997, 230-247

Papadopoulos, G., Phon-Amnuaisuk, S., Wiggins, G., and Tuson, A.L., 1998: Evolutionary Methods for Music Composition. Submitted to the International Conference on Computer Music (ICMC 98)

Saunders, R. and Gero J. S., 2001: Artificial Creativity: A Synthetic Approach to the Study of Creative Behaviour, in J. S. Gero (ed.), Proceedings of the Fifth Conference on Computational and Cognitive Models of Creative Design, Key Centre of Design Com-putting and Cognition. (2001)

Spector, L., and A. Alpern, 1994: Criticism, Culture, and the Automatic Generation of Artworks. In Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI-94, pp. 3-8. Menlo Park, CA and Cambridge, MA: AAAI Press/The MIT Press

Spector, L., and A. Alpern, 1995: Induction and Recapitulation of Deep Musical Structure. In Working Notes of the IJCAI-95 Workshop on Artificial Intelligence and Music . pp. 41-48

Svangård, N. and Nordin, P., 2004: Automated Aesthetic Selection of Evolutionary Art by Distance Based Classification of Genomes and Phenomes using the Universal Similarity Metric, To Appear in Proceedings of the 2nd European Workshop on Evolutionary Music and Art (EvoMUSART 2004)