Generating Rhythms with Genetic Algorithms

Damon Horowitz

MIT Media Laboratory, 20 Ames St. F15-488, Cambridge, MA 02139
damon@media.mit.edu

Abstract

My system uses an interactive genetic algorithm to learn a
user’s criteria for the task of generating musical rhythms. Interacti
| genetic algorithms (Smith 91) are well suited to
solving this problem because they allow a user to simply
execute fitness functions (that is, to choose which rhythms or
features of rhythms he likes), without necessarily understanding
the details or parameters of these functions. As the system learns
| (develops an increasingly accurate model of the function which
represents the user’s criteria), the quality of the rhythms it
produces improves to suit the user’s taste. This approach is
largely motivated by Richard Dawkins, who succinctly
| summarizes the attraction of IGAs for artistic endeavors in
stating: “Effective searching procedures become, when the
search space is sufficiently large, indistinguishable from true
creativity” (Dawkins 86).

In the context of this project, rhythms are one measure
long sequences of notes and rests occurring on natural
pulse subdivisions of a beat; I only deal with a specific
subset of the enormous class of rhythms, in order to
provide a well-defined domain for the application of the
learning algorithm. The benefit of this reduction of the
domain is that a rhythm phenotype can now be viewed as
a simple vector. Thus, the set of rhythms satisfying
the user’s criteria could be represented by a Boolean
formula. I actually use a slightly more complex representation for
the rhythm genotype, motivated by the benefits of using a
diploid genetic structure, consisting of several short array
templates; the order of the layering of these templates in
creating the phenotype effectively determines the
dominance hierarchy between the genes.

The simplest mode of interaction is for the user to
playback each of the rhythms in a randomly generated
population, and then subjectively assign them fitness values based upon their satisfaction of his criteria. The
system then uses standard GA selection (with fitness
scaling), reproduction (with crossover monitors), and
mutation operators. In order to deal with the difficulties
resultant from the subjectivity and variability of the user’s
criteria, there are also several objective functions with
which the system can automatically evolve generations of
rhythms: syncopation, density, downbeat, beat repetition,
cross rhythm, and cluster functions are currently included.
Each of these functions represents an axis in a feature
space which is useful for distinguishing rhythms. While
these are only a few of the many possible objective
functions that could be implemented, they provide a
richset of possibilities with which to begin exploring.
The user can specify the ideal target value for each of
these fitness functions, and also their relative importance
(weighting of coefficients) in determining the overall
fitness of a rhythm. The system then automatically
| evolves the indicated number of successive generations,
using the objective fitness values to determine selection.

The system also makes use of a meta-level genetic
algorithm designed to evolve populations of parameters
(target values and weights) to the objective fitness
functions defined above. This is motivated by the
research done in the application of genetic algorithms to
the k-nearest-neighbor technique of classification (Punch
et al. 93); each meta-level individual represents a warping
of K-NN space, such that the fitness of each individual is
determined by how well its warping of the feature-space
helps to discriminate useful features, and thus correctly
perform classifications. Evolving populations of meta-
individuals allows a user to quickly reduce the search
space by subjective evaluation of the rhythms generated
by the meta-individuals, without having to directly specify
values for the objective functions.

This combination of methods proves to be a powerful
hybrid approach to the subjectivity problem, one which
| allows for greater coverage of the search space than would
have been possible ordinarily using a small population
(which is necessitated by most IGA’s, and is particularly
important when dealing with sequential acoustic data), and
more efficient convergence on a satisfying solution. The
system is able to converge on near-optimal solutions
(acceptable to test users) after about fifty user-evaluations
of rhythms. While the GA itself is mechanically quite
simple, it is important to note that the implementation of
appropriate fitness functions is difficult, and largely
determines the musicality of the output. The major future
improvement will involve adding the capacity for the
system to learn to design its own fitness functions to
represent features characteristic of rhythms selected by
users in past sessions.

References

Selection and Classification Using Genetic Algorithms. In
Proceedings of the Fifth International Conference on
Genetic Algorithms.
Smith, J. 1991. Designing Biomorphs with an Interactive
Genetic Algorithm. In Proceedings of the Fourth
International Conference on Genetic Algorithms.