A Self-Critical Compositional Algorithm

Robert Rowe
NYU Dept. of Music
35 West Fourth St. Suite 777
Washington Square
New York, New York 10003

A real-time system has been built which analyzes features of performed music and their evolution over time. Complementing this capacity are a set of compositional methods, which can generate or transform MIDI-level data. This paper describes a set of production rules, which have been implemented to allow the program itself to select compositional methods, and apply them, in real time. These production rules are expressed in the terms made possible by the analysis part of the program: the conditions evaluate logical expressions on feature classifications or descriptions of behavior. The execution part of the rules calls up composition methods to generate or modify material. The production rules implement a compositional critic, which is used to evaluate and modify the output of a real-time music system before the generated music is sent to the synthesizer. In other words, the critic forms the second pass of a system which generates material following some methods, then uses the critic to evaluate and correct the material emanating from the first pass. A discussion of the production rules and several musical examples are included.

Overview

Cypher, an interactive music analysis/performance program written by the author [Rowe91; Rowe92], produces classifications of musical behavior and responds with new musical output in real time. There are two major components: the listener part analyzes MIDI input, and the player part composes the program’s responses. The program has been used in the performance of composed works, and in completely improvisational settings, always in conjunction with human players. Both the listener and player components of the program are hierarchical, in the sense that higher levels of processing correspond to longer spans of time, and deal with abstractions produced by lower levels. In terms of analysis, the lowest level classifies several features of incoming MIDI data: density, register, loudness, speed, duration, and harmony. Higher levels are concerned with finding groups and regularity in the behavior of these features over time. Further high-level tasks include beat tracking and key identification. On the performance side, three classes of compositional methods are available for constructing responses: sequencing, compositional algorithms, and transformation. The user of the program determines which methods will actually be used to compose a response by establishing connections between listener messages.
and player methods, where a connection causes a method to be invoked when the corresponding listener message is received.

In the diagram above, a schematized representation of the program's processing is shown. First, MIDI arrives from some source. This input is analyzed by one copy of the Cypher listener, which produces the classifications outlined above. These classifications will cause certain compositional methods to be invoked, depending on connections between the two specified by the user. This paper describes the function of the next process in the chain, the critic. The critic comprises a second instance of a Cypher listener, and a set of production rules. The critic listener will analyze the output of the composition methods, using criteria identical to those active in the analysis listener. The productions are a set of rules associating listener messages coming from the critical analysis with operations to be performed on the output of the composition methods engaged in the previous stage. The revised material makes up the output of the system as a whole, shown as the final link in the chain.

Aesthetic Productions

The conditional parts of the critic's production rules are logical expressions on the values of featural and regularity classifications produced by the critic's listener. Classifications are generated for such perceptual categories as density, register, speed, loudness, etc. A simple example is the following: the critic will prefer to reduce the vertical density of material being presented at a high speed (high horizontal density). The conditional part of such a rule can be expressed as: if (FastVal(featurespace) > MEDIUM_FAST). The action part of the rule would apply a transformation to the incipient output, removing some pitches from chords about to be played at a high speed.

The action part of a rule is not limited to modifying the output of the compositional methods – several other strategies have been implemented as well. One approach allows the critic to change the connections between the analysis listener messages and the original composition methods: in other
words, between stages two and three of the processing diagram shown above. In that case, the critic is not correcting the output of the compositional methods, but automatically assisting in the choice of which methods to apply in the first place. Further, the critic may choose to associate particular sounds with certain playing styles. That is, the timbres used to perform the system’s output can be chosen as a function of the character of the output played. The critic may prefer to perform high material using woodwind-like sounds, for example, while registrally lower output is played with string timbres.

Interestingness

Another permutation of the processing stages diagrammed above sends the output of the system back to the input, completing a feedback loop in which the stage 2 listener analyzes, and the stage 3 composition methods respond to, the output of Cypher itself. Such a feedback loop allows the program to produce novel musical material without human input, reacting to and elaborating its own output. However, processing all of the program’s material this way could quickly swamp the capacities of the system, not to mention the comprehension of the listener. Therefore, another function of the critic is to select “interesting” material from the output of the methods for further treatment by the feedback loop. Output events which differ markedly from the recent past are considered “interesting” by the critic. The same basic criterion is used by a Cypher listener to find boundaries between groups of events: when discontinuities are noticed across several features between adjacent events, a boundary is asserted. For the critic, such events are marked as interesting.

Already in such a simple application of the critic, we can see that the rules associated with it form, in effect, an aesthetic bias for the program. Declaring events whose feature classifications differ markedly from their predecessor to be “interesting” is no more or less than an aesthetic choice; there is no inherent reason to find any musical event more interesting than any other. Aesthetic biases can be codified in terms of preferred actions to be taken in response to material exhibiting certain kinds of regularity, or feature classifications, or harmonic behavior – in short, preferences can be established for all of the situations reported out of the listener.

Musical Applications

Some of the critic’s functions are always active: production rules to ensure consistency and coherence of output, and the “interestingness” evaluation. Others are enabled and disabled for specific musical contexts, as during an improvised performance with saxophonist Steve Coleman in April of 1991.
At some points, a human operator wired connections between the analysis listener and response methods onstage. At others, an added critical rule set was engaged to allow the program to select its own composition methods, according to the nature of the saxophone performance and the program’s output. Much of the power of this approach lies in the capacity of the compositional transformation routines to adjust event lists to match analyzed contexts. One such adjustment aligns program output with any beat period found in the human performance. Another can use various harmonic memories in modifying output to accord with the pitch material arriving from the outside. Both of these techniques were used in the Colenman performance; one can imagine several different critical rule sets which would induce varying behaviors at different points in the performance. As a straightforward extension, meta-critic rules could swap in or out complete production rule sets as a function of the musical style being detected in the performance of the program’s partner.

Conclusion

Adding a critical faculty to an interactive music system advances both musical and technical agendas. Technically, a program so equipped is prepared to give an explanation for its choices; using traces of critical activity, we can better evaluate why the program reacts the way it does in a given context. Musically, a self-critical computer performer can have internalized heuristics for dealing with unknown musical input in an appropriate and consistent way. Moreover, a generalized mechanism for evaluating and critically modifying output opens additional levels of musical processing to compositional thought, particularly when such behaviors can themselves be selectively introduced or suppressed in specific musical contexts.

Acknowledgements

The research described in this paper was carried out at the MIT Media Laboratory, and supported by a grant from the Yamaha Corporation.

References
