

Interacting with a Musical Learning System: The Continuator



François Pachet

SONY CSL-Paris
Researcher

Presented by Ching-Hua Chuan



Introduction

- Real-time interactive musical instruments that are able to produce stylistically consistent music.
- The system learns music styles automatically, and is seamlessly integrated in the playing mode of the musicians.



Bernard Lubat



The Continuator and children



Inside the Continuator

- Automatic learning and generation
 - Hierarchical Markov models
 - Improvement of classic Markov models
- Interactive musical instrument
 - Real-time generation (timing issues)
 - Biasing Markov generation
 - Control and high-level structure



Possible Continuations variable-order Markov chain

- Sequence #1: {A B C D}
- Sequence #2: {A B B C}
- Ex #1: a subsequence {B} has two possible continuations: C (from sequence #1) and B (from sequence #2)
- Ex #2: {A B} → C (1) or B (2)
- Ex #3: {B B C} → D (1) (longest possible subsequence).
- Ex #4: {A} → {B, B} (repeat all similar continuations)



Improvement of classic Markov models

- No continuation is found.
- Less refined reduction function. (pitch region instead of pitch)
 - Fig1: {PR1 PR1 PR2 PR3 PR5}
 - Fig2: {PR1 PR1 PR2}
 - Continuation = {PR3}, which is G in this case.

Figure 1. An arpeggio learnt by the Continuator.

Figure 2. An input sequence which does not match exactly with a subsequence in learnt corpus.

A typical hierarchy of reduction functions

- 1- pitch * duration * velocity
- 2 – small pitch region * velocity
- 3 – small pitch regions
- 4 – large pitch regions

where the numbering indicates the order in which the functions are to be considered in cases of failure in the matching process.

Real-time Generation

- John McLaughlin, a mean duration of 66 milliseconds per note. A good estimation of the maximum delay between two fast notes is about 50 milliseconds.
- Incoming note detection, phrase end detection, step-by-step generation

Biasing Markov Generation

- Weighting the notes according to how they match the external input.

$$Fitness(p, piano) = \frac{nb \text{ notes common top and piano}}{nb \text{ notes in piano}}$$

$$Prob(x) = S * Markov_Prob(x) + (1 - S) * Fitness(x, Context)$$

S=1, we get a musical automaton insensitive to the musical context,
 S=0, we get a reactive system which generates the closest musical elements to the external input it finds in the database.


Control and High-Level Structure



- Parameters allow the musician to switch on or off basic functions such as the learning process or the continuation process.
- By default, the system stops playing when the user does, to avoid superposition of improvisations.
- Other parameters: the number of notes to be generated by the system, the tempo of the generated sequence...

Experimentations



- Indistinguishability 
- Attachment
- Subjective Impressions: the AHA effect



Claude Barthelemy

Demo



- It's show time!