OMAI: AN AI TOOLKIT FOR OM#

Anders Vinjar *

https://www.avinjar.no
Artistic Research Residency at IRCAM, 2019–2020
anders@avinjar.no

ABSTRACT

OMAI is a toolkit for composers wanting to explore the use of artificial intelligence and machine learning in computer assisted music composition. The OMAI library for the OM#CAC-application implements techniques for data classification, prediction and generation, in order to integrate these techniques in composition workflows.

Examples are provided using simple musical structures, highlighting possible extensions and applications.

A brief description of OM# - a new CAC environment derived from OpenMusic - is included. OM# is a visual programming language dedicated to musical structure generation and processing, available on Linux, MacOSX and Windows platforms.

1. INTRODUCTION

As soon as computers were conceived, composers entered the labs and started to explore potentials for computation and representation in search for new creative options. Computer assisted composition (CAC) became part of the emerging field of artificial intelligence (AI) [1].

Artificial intelligence and machine learning are commonly used in research on computational creativity [2], “autonomous" generative and/or improvisation systems [3, 4, 5], or real-time performance monitoring and interaction [6]. However, apart from a few examples [7, 8], machine learning and AI are rarely explored by composers as a means for composing music, and current techniques to assist composition tasks (e.g. [5]) generally do not operate directly in compositional workflows and environments.

CAC systems provide explicit computational approaches through the use of end-user programming languages [9]. OM#[10], derived from OpenMusic, is a recent newcomer amongst the visual programming environments for music and sound, allowing users to process and generate scores, sounds and many other kinds of musical structures, handle scheduling, input/output and interaction with external systems, and provide a platform for general programming and scripting.

This article presents ongoing work exploring the use of AI and machine learning techniques in the OM# environment. In contrast to approaches aimed at automatic creation or machine classification systems, the aim of OMAI is to provide useful techniques to aid in the composers workflow. The approach is a “composer-centered" machine learning approach [11] allowing users of CAC systems to implement experimental cycles including pre-processing, training, and setting the parameters of machine learning models for data generation, decision support or solving other generic problems.

The toolkit is designed to be used in a bottom-up workflow, supporting creative tasks where there’s not one correct answer. The goal is to provide pragmatic tools, having an adequate interface for end-users while retaining an open-ended environment in OM#s graphical programming environment.

2. OM#

OM# (om-sharp[1]) is a computer-assisted composition environment derived from OpenMusic: a visual programming language dedicated to musical structure generation and processing.

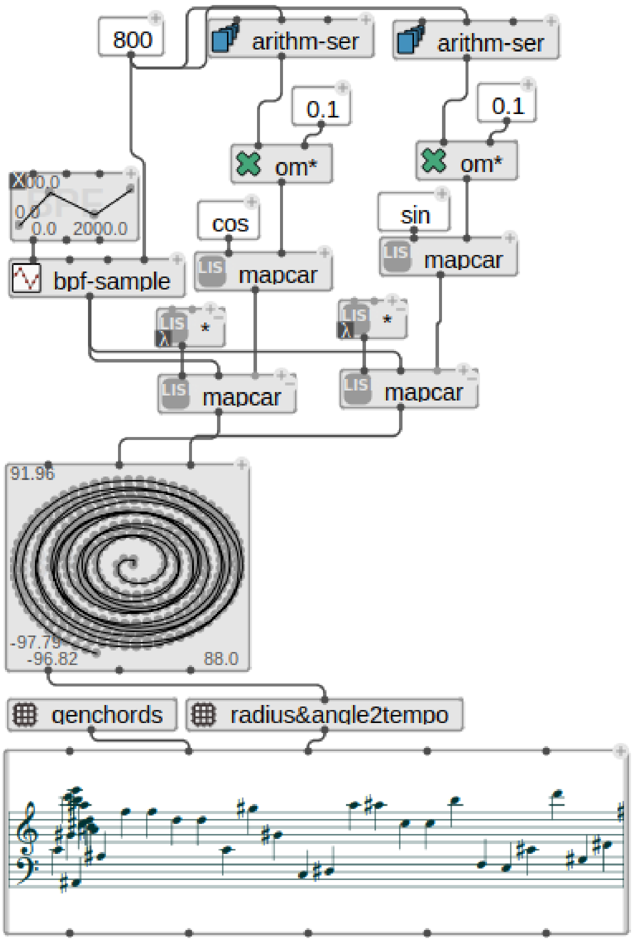

Figure 1: OM# — graphical programming

* This work was supported by IRCAMs Artistic Research Residency Program

[1] https://cac-t-u-s.github.io/om-sharp/

The visual language is based on Common Lisp, and allows to create programs that are interpreted in this language. Visual programs are made by assembling and connecting icons representing Lisp functions and data structures, built-in control structures (e.g. loops), and other program constructs. The visual language can therefore be used for general-purpose programming, and reuse any existing Common Lisp code. A set of in-built tools and external libraries make it a powerful environment for music composition: various classes implementing musical structures are provided, associated with graphical editors including common music notation, MIDI, OSC, 2D/3D curves, and audio buffers.

OM# is available for Linux, macOS and Windows. The first successful port of OpenMusic to Linux was done in 2013[12], and since then the development of both OpenMusic and OM# has taken place on these 3 platforms. This software is free, distributed under the GPLv3 license.

3. TOOLS AND ALGORITHMS

The OMAI library for OpenMusic provides basic tools from the domain, with elementary algorithms to classify vectors in a multidimensional feature space [13].

3.1. Vector Space

A generic data structure called VECTOR-SPACE is used to store vectorized data and information necessary to train and run machine learning and classification models. The structure is simple and generic; it is initialized with a list of entries (key, value) for a hash-table of vectors, where keys can be strings or any other unique identifiers for the different vectors.

Feature-vectors are also stored as hash-tables using descriptor names as keys. Descriptor names can also be input to the VECTOR-SPACE initialization for facilitating visualization and query operations. A graphical interface allows the 2D and 3D visualization of vectors in the feature space, selecting two or three descriptors as projection axes (see Figure 2).

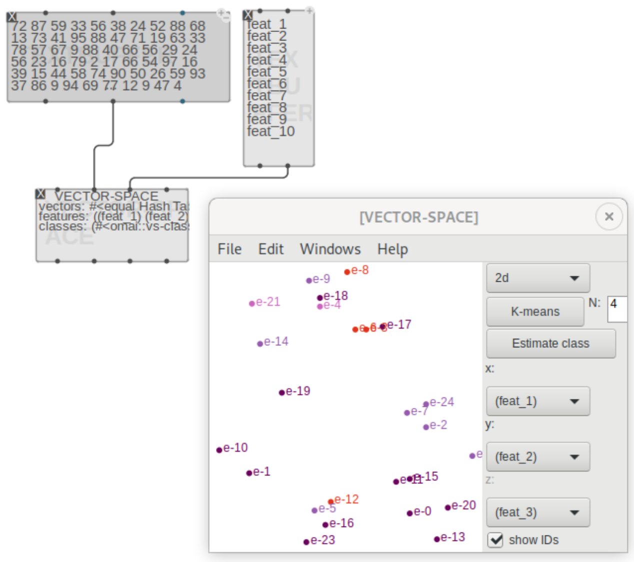
```
72 87 59 33 56 38 24 52 88 68
13 73 41 95 88 47 71 19 63 33
78 57 67 9 88 40 66 56 29 24
56 23 16 79 2 17 66 54 97 16
39 15 44 58 74 90 50 26 59 93
37 86 9 94 69 77 12 9 47 4
```
```
feat_1
feat_2
feat_3
feat_4
feat_5
feat_6
feat_7
feat_8
feat_9
feat_10
```
```
VECTOR-SPACE
vectors: #<equal Hash Ta
features: ((feat_1) (feat_2)
classes: (#<omai::vs-clas
```

Figure 2: Simple 2-D vector space visualization.

3.2. Clustering and Classification

Within the VECTOR-SPACE, distance-functions are used to retrieve information and compute similarity between feature-vectors. These operations can be applied in various algorithms for automatic clustering and classification.

The *k-means* algorithm performs "unsupervised" clustering by grouping feature-vectors around a given number of centroids. This process can be done in a visual program (see Figure 3) or interactively from within the VECTOR-SPACE graphical interface (as in Figure 2).

Supervised classification approaches (based on preliminary labelling information) are also available. In our generic model, a *class* is represented by a unique label and a list of IDs corresponding to known members of this class (this is typically determined during a preliminary training stage). Based on this information (which implicitly labels all known class members), it is possible to compare any unlabelled vector with centroid feature-vectors of the different classes, or its similarity with an established neighbourhood in the multidimensional feature space (*k-NN*). Such comparison allow to determine a measure of likelihood for this vector belonging to a certain class.

3.3. Musical Descriptors

An extensible set of descriptor algorithms allowing to extract features from objects are included with the system, many of which are aimed at musical content, pitch attributes, harmonic attributes, chord sequences, temporal attributes, variability, repeatability at various levels, degree of recurring figures etc. These features can be combined freely to constitute the N-dimensional vectors representing musical data in different OMAI algorithms.

The set of provided feature extractors can easily be extended using OM#s graphical programming environment or by coding in Lisp.

An example application could be: given a set of existing or generated material, extract a feature-vector for each of these using a selection of features, the system would cluster the input material accordingly.

For a composer these clusters could represent musical contrasts, variations, similarities etc. Having the ability to generate new material without necessarily knowing or caring exactly how it was generated, only that it ends up together with other material within a certain cluster, greatly reduces the amount of time needed to search for wanted solutions.

4. AI AND CAC, MUSICAL COMPOSITION, ARTISTIC NEEDS

Algorithms for ML, HMM, kNN, Neural Networks, Viterbi and more are part of the OMAI project. Tools and editors based on these algorithms are being developed as part of OMAI.

The Machine Learning and Clustering approaches described above are useful together with other AI algorithms to handle musical data traditionally worked with in CAC applications. They can also provide effective handles for more ill-defined, but arguably important and readily perceivable musical features such as "texture", "structure", "entropy" etc., musical qualities many modern composers use much time trying to control.

A possibly interesting observation of this project is that often rather simple versions of more advanced techniques together with an

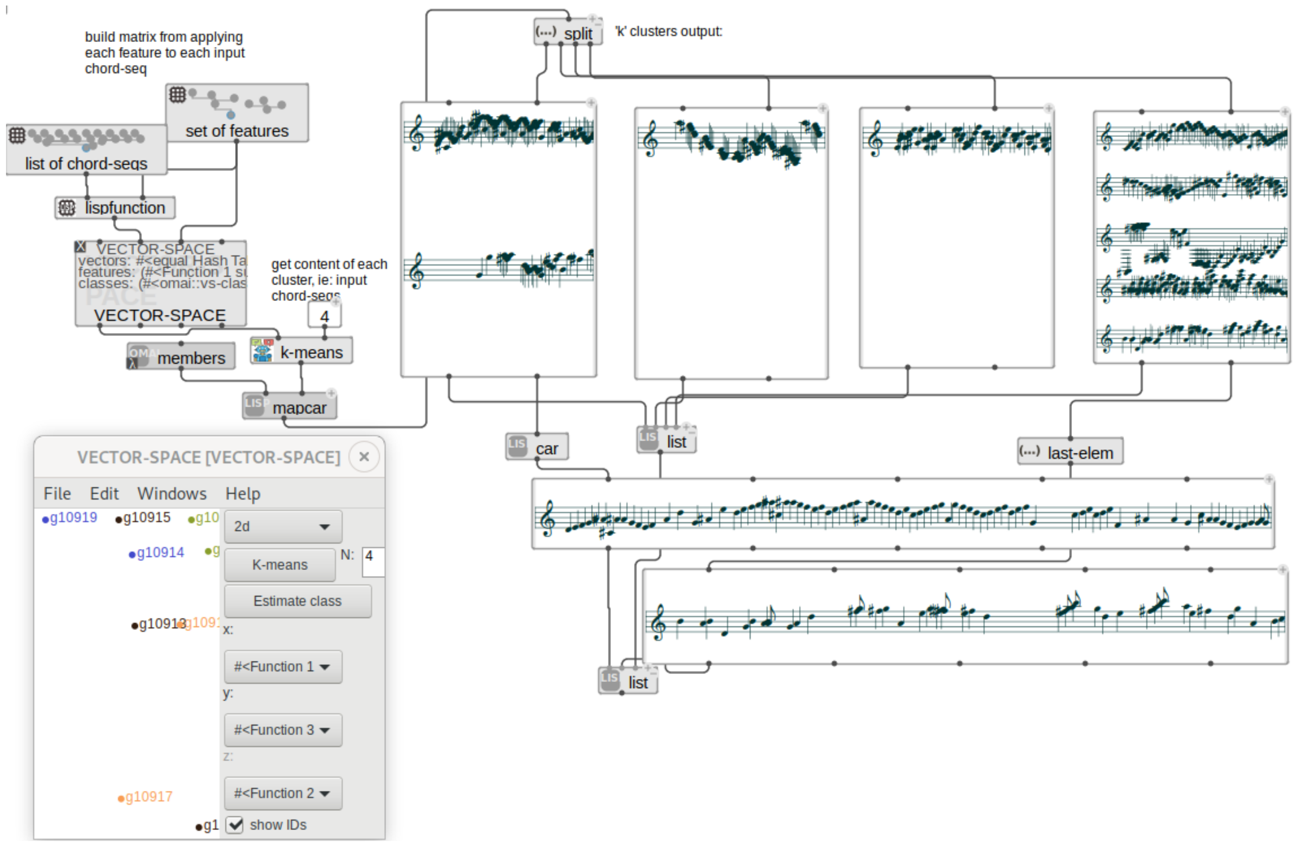

Figure 3: An example of clustering applied to a set of input structures. The output of the k-Means algorithm is 4 clusters. One sample from each cluster is displayed at the bottom

exploratory approach is very useful in the context of creative work. Stochastic methods are everywhere, and here as well, exact answers and reproducibility is often not particularly useful as such. In contrast: degrees of precision, possibly even errors, new solutions within certain constraints - can all be potential triggers for cool things to happen while composing.

Where a researcher most often would make sure to use large amounts of data to train an HMM or Neural Network, this is not necessarily interesting in our context. Just as Markov generation models seldom provide interesting results in composition work beyond 2-3 order, an ANN can output interesting results already after 2-10 iterations, and the *difference* between 2 given iterations are by itself very useful, e.g. to provide variants, development, embellishment etc.

Miller Puckette is quoted in the preface to "The OM Composer's Book"[14], illustrating some of the challenges:

> "CGM (Computer Generated Music, ie. Audio) is in effect building instruments (which were previously made of wood and the like), but CAC is in effect making the computer carry out thought processes previously carried out in human brains. Clearly, a piece of wood is easier to understand than even a small portion of a human brain. ... Ultimately, CAC researchers will have to settle for much less than a full understanding of even a single musical phenomenon. The best that can be hoped for is partial solutions to oversimplified versions of the real problems."

Good AI integrated in CAC tools may help bridge the gap between the composers mind and the systems they work with.

5. MODELLING, GENERATIVE ALGORITHMS

The OMAI system has been used by the author during recent composition work, e.g. to optimize fingering positions while scoring for guitar, and extracting generative patterns from analysis of input musical sequences using HMMs (Hidden Markov Models).

While developing the models used in this particular piece, the resulting scores have been evaluated along the way together with a professional guitarist to verify their level of 'guitaristicity'.

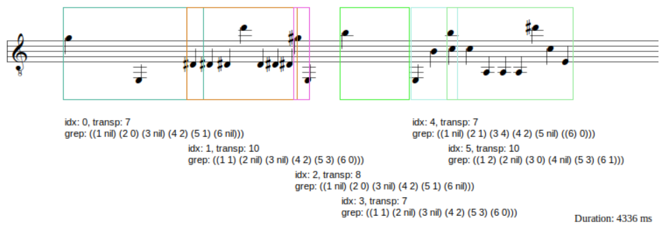
idx: 0, transp: 7
grep: ((1 nil) (2 0) (3 nil) (4 2) (5 1) (6 nil))
idx: 1, transp: 10
grep: ((1 1) (2 nil) (3 nil) (4 2) (5 3) (6 0))
idx: 2, transp: 8
grep: ((1 nil) (2 0) (3 nil) (4 2) (5 1) (6 nil))
idx: 3, transp: 7
grep: ((1 1) (2 nil) (3 nil) (4 2) (5 3) (6 0))
idx: 4, transp: 7
grep: ((1 nil) (2 1) (3 4) (4 2) (5 nil) ((6) 0))
idx: 5, transp: 10
grep: ((1 2) (2 nil) (3 0) (4 nil) (5 3) (6 1))
Duration: 4336 ms

Figure 4: *Tabulature composition, generative algorithms use models of hands and fingers to suggest possible sets of notes*

6. RESOURCES AND DOWNLOAD

The sources of OMAI are open source, and distributed under the GPLv3 license, available along with documentation and examples at:
`https://github.com/openmusic-project/OMAI`

7. CONCLUSIONS

A project developing AI-based tools and techniques for Computer assisted composition and creative work, OMAI, is presented. The tools are part of OM# and the OpenMusic family of CAC environments.

Acknowledgments

This work is supported by IRCAMs Artistic Research Residency Program and the Norwegian Cultural Council.

8. REFERENCES

[1] Lejaren A. Hiller and Leonard M. Isaacson, *Experimental Music: Composition With an Electronic Computer*, McGraw-Hill, 1959.

[2] Philippe Pasquier, Arne Eigenfeldt, Oliver Bown, and Shlomo Dubnov, "An Introduction to Musical Metacreation," *Computers in Entertainment*, vol. 14, no. 2, 2016.

[3] F. Ghedini, F. Pachet, and P. Roy, "Creating Music and Texts with Flow Machines," in *Multidisciplinary Contributions to the Science of Creative Thinking (Creativity in the Twenty First Century)*, G. E. Corazza and S. Agnoli, Eds. Springer, 2015.

[4] Gérard Assayag, Shlomo Dubnov, and Olivier Delerue, "Guessing the Composer's Mind: Applying Universal Prediction to Musical Style," in *Proc. International Computer Music Conference (ICMC'99)*, Beijing, China, 1999.

[5] Léopold Crestel and Philippe Esling, "Live Orchestral Piano, a system for real-time orchestral music generation," in *Proceedings of the Sound and Music Computing Conference (SMC'17)*, Espoo, Finland, 2017.

[6] J. Françoise, N. Schnell, and F. Bevilacqua, "A Multimodal Probabilistic Model for Gesture-based Control of Sound Synthesis," in *ACM MultiMedia (MM'13)*, Barcelona, Spain, 2013.

[7] David Cope, *Experiments in Musical Intelligence*, A-R Editions, 1996.

[8] Shlomo Dubnov and Greg Surges, "Delegating Creativity: Use of Musical Algorithms in Machine Listening and Composition," in *Digital Da Vinci: Computers in Music*, Newton Lee, Ed. Springer, 2014.

[9] Gérard Assayag, Camilo Rueda, Mikael Laurson, Carlos Agon, and Olivier Delerue, "Computer Assisted Composition at IRCAM: From PatchWork to OpenMusic," *Computer Music Journal*, vol. 23, no. 3, 1999.

[10] Jean Bresson, Dimitri Bouche, Thibaut Carpentier, Diemo Schwarz, and Jérémie Garcia, "Next-generation Computer-aided Composition Environment: A New Implementation of OpenMusic," in *International Computer Music Conference (ICMC'17)*, Shanghai, China, 2017, Proceedings of the International Computer Music Conference.

[11] Marco Gillies, Rebecca Fiebrink, Atau Tanaka, Baptiste Caramiaux, Jérémie Garcia, Frédéric Bevilacqua, Alexis Heloir, Fabrizio Nunnari, Wendy Mackay, Saleema Amershi, Bongshin Lee, Nicolas D 'alessandro, Joëlle Tilmanne, and Todd Kulesza, "Human-Centered Machine Learning Workshop at CHI'16," in *Proc. CHI'16 – Extended Abstracts on Human Factors in Computing Systems*, San Jose, USA, 2016.

[12] Anders Vinjar and Jean Bresson, "OpenMusic on Linux," in *Linux Audio Conference*, Karlsruhe, Germany, 2014.

[13] C. D. Manning, P. Raghavan, and H. Schütze, *An Introduction to Information Retrieval*, Cambridge University Press, 2009.

[14] Miller Puckette, ," in *The OM Composer's Book 1*, Jean Bresson, Carlos Agon, and Gérard Assayag, Eds. Editions Delatour France / Ircam-Centre Pompidou, 2006, cote interne IRCAM: Agon06a.