OM-AI: A Toolkit to Support AI-Based Computer-Assisted Composition Workflows in OpenMusic

Anders Vinjar, Jean Bresson

OM-AI: A TOOLKIT TO SUPPORT AI-BASED COMPUTER-ASSISTED COMPOSITION WORKFLOWS IN OPENMUSIC

Anders Vinjar
Composer Researcher
Oslo, Norway
anders@avinjar.no

Jean Bresson
STMS lab: IRCAM – CNRS – Sorbone Université
Paris, France
jean.bresson@ircam.fr

ABSTRACT

We present ongoing works exploring the use of artificial intelligence and machine learning in computer-assisted music composition. The OM-AI library for OpenMusic implements well-known techniques for data classification and prediction, in order to integrate them in composition workflows. We give examples using simple musical structures, highlighting possible extensions and applications.

1. INTRODUCTION

The idea of making programs capable of composing appeared early in the history of computer music [1]. Today artificial intelligence and machine learning are commonly used for research on computational creativity [2], "autonomous" generative and/or improvisation systems [3–5], or real-time performance monitoring and interaction [6]. However, apart from a few examples [7, 8], machine learning and AI are rarely exploited by composers as a means for writing music.

Computer-assisted composition systems develop explicit computational approaches through the use of end-user programming languages [9]. At the forefront of this approach, OpenMusic is a popular visual programming environment allowing users to process and generate scores, sounds and many other kinds of musical structures [10].

We present ongoing works exploring the use of AI and machine learning techniques in this environment. In contrast to approaches aimed at autonomous creative systems, our aim is to apply these techniques as composition assistance in the classification and processing of musical structures and parameters. Therefore we target a "composer-centered" machine learning approach [11] allowing users of computer-assisted composition systems to implement experimental cycles including preprocessing, training, and setting the parameters of machine learning models for the data generation, decision support or solving other generic problems.

2. TOOLS AND ALGORITHMS

The OM-AI library for OpenMusic provides basic tools from the domain, with elementary algorithms to classify vectors in a multidimensional feature space [12].

2.1 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. It is assumed that each feature-vector contains the same set of descriptors. 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 1).

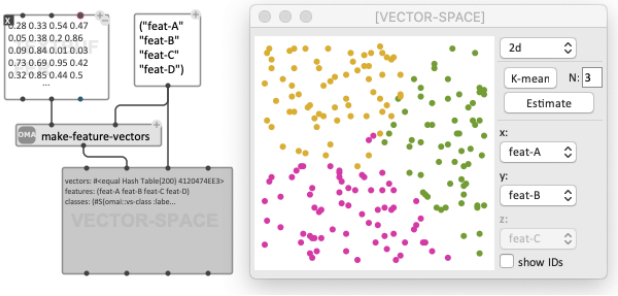

Figure 1. 2-D vector space visualization.

2.2 Clustering and Classification

Within the VECTOR-SPACE, built-in or user-defined distance-functions are used to compute measurements of similarity between feature-vectors (i.e. distance within the feature space) and centroids for any set of vectors. These operations are 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 2) or interactively from within the VECTOR-SPACE graphical interface (as in Figure 1).

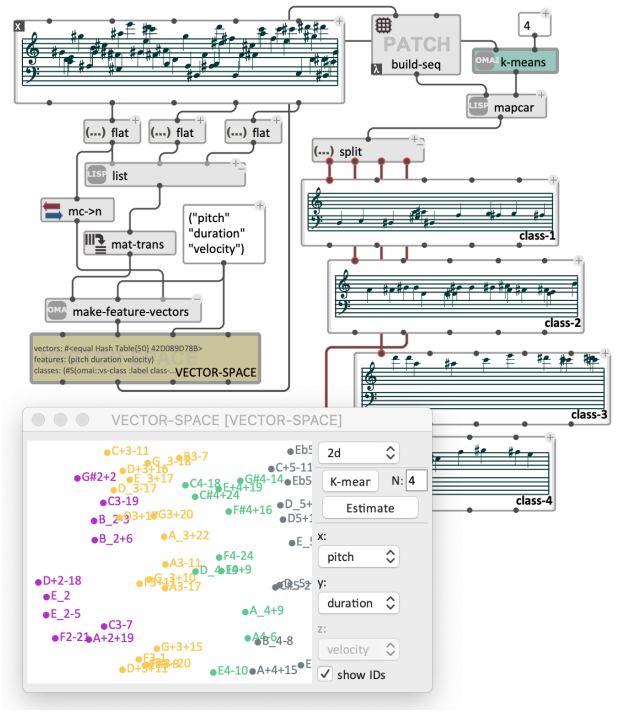

Figure 2. A simple example of clustering applied to a sequence of notes. Feature-vectors are formatted and input to the VECTOR-SPACE object. From there, the *k-means* algorithm clusters the data based on the specified features (*pitch*, *duration*, *velocity*). Vectors are output and split by clusters to generate four separate voices.

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. MUSICAL DESCRIPTORS

An extensible set of description algorithms allows to extract features from musical objects (harmonies, chord sequences, temporal attributes etc.). These features can be combined freely to constitute the vectors representing musical data in the different OM-AI algorithms.

Currently available descriptors include pitch statistics: most common pitch or pitch class, pitch histograms, mean pitch, pitch variety, interval between prevalent pitches, relative pitch prevalence, importance of different registers, tonal centers, etc., and melodic features: intervals, arpeggios and repetitions, melodic motions and arcs, etc.

We are currently working on extending the set of provided feature extraction algorithms and experimenting with more advanced musical examples.

4. RESOURCES AND DOWNLOAD

The sources of OM-AI are open and available along with documentation and examples at:
`https://github.com/openmusic-project/OMAI`

Acknowledgments

This work is supported by the PEPS 3IA programme of the French National Center for Scientific Research, IRCAM's Artistic Research Residency Program and Notam.

5. REFERENCES

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

[2] P. Pasquier, A. Eigenfeldt, O. Bown, and S. 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*, G. E. Corazza and S. Agnoli, Eds. Springer, 2015.

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

[5] L. Crestel and P. Esling, "Live Orchestral Piano, a system for real-time orchestral music generation," in *Proc. Sound and Music Computing (SMC'17)*, Espoo, 2017.

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

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

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

[9] G. Assayag, *et al.*, "Computer Assisted Composition at IRCAM: From PatchWork to OpenMusic," *Computer Music Journal*, vol. 23, no. 3, 1999.

[10] J. Bresson, C. Agon, and G. Assayag, "OpenMusic. Visual Programming Environment for Music Composition, Analysis and Research," in *Proc. ACM MultiMedia (MM'11)*, Scottsdale, 2011.

[11] M. Gillies, R. Fiebrink, A. Tanaka, *et al.* "Human-Centered Machine Learning Workshop at CHI'16," in *Proc. CHI'16 – Extended Abstracts on Human Factors in Computing Systems*, San Jose, 2016.

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