Musical Gesture Recognition Using Machine Learning and Audio Descriptors

Paul Best, Jean Bresson, Diemo Schwarz

Musical Gesture Recognition Using Machine Learning and Audio Descriptors

Paul Best
STMS Lab
Ircam, CNRS, Sorbonne Université
Paris, France
paul.best@ircam.fr

Jean Bresson
STMS Lab
Ircam, CNRS, Sorbonne Université
Paris, France
jean.bresson@ircam.fr

Diemo Schwarz
STMS Lab
Ircam, CNRS, Sorbonne Université
Paris, France
diemo.schwarz@ircam.f

***Abstract*—We report preliminary results of an ongoing project on automatic recognition and classification of musical "gestures" from audio extracts. We use a machine learning tool designed for motion tracking and recognition, applied to labeled vectors of audio descriptors in order to recognize hypothetical gestures formed by these descriptors. A hypothesis is that the classes detected in audio descriptors can be used to identify higher-level/abstract musical structures which might not be described easily using standard/symbolic representations.**

***Index Terms*—Machine learning, gesture, hidden Markov chains, audio descriptors.**

I. Introduction

Musical structures carry abstract perceptual features and characteristics that can be straightforward to identify by composers and/or listeners, but difficult or impossible to formally describe using the elements of standard score representations (e.g. identifying harmonic/melodic patterns etc.) Composers and authors often use the term of *gesture* to characterize these dynamic elements constituting musical forms, as an analogy with the idea of gesture in physical movements [1], [2].

Physical movements can be analyzed using machine learning techniques: Ircam's XMM library [3], for instance, uses a combination of Gaussian mixture models (GMM) and hidden Markov models (HMM) in order to recognize performed gestures from parallel streams of descriptors extracted from motion capture (Cartesian coordinates, speed, etc.) HMM have shown good performance processing movements in such time-series [4], [5], and require much smaller training sets than neural networks to build reliable models. We propose here that similar models could be used by composers working with audio descriptors to recognize and classify musical gestures.

This hypothesis is tested on a real, specific example, in the context a composer's musical research residency project at Ircam (Paris). We developed a binding of the XMM library in the OpenMusic composition and visual programming software [6], associated with audio description and analysis tools, and used this framework to run preliminary experiments.

A general objective of this work is to provide music composers with tools and interfaces to work with machine learning in a computer-aided composition environment [7]. HMM techniques fit well with this situation, where algorithms will be trained on data input by end-users, rather than on databases or other large datasets distributed for instance over the web. A crucial part of the work, however, resides in the strategy employed to efficiently test and select the relevant features and parameters to use in order to carry out a recognition task given some input data type and training set at hand.

II. Protocol

The dataset preparation, model testing and validation are carried out and controlled in OpenMusic visual programs such as the one visible in Fig. 1, using the OM-XMM library.

A. Dataset

The data-set used in this experiment is a 9-minutes-long extract of the piece *Zāmyād* for cello by contemporary music composer Alireza Fahrang.[1] This piece is widely built around the concept of musical gestures. Indeed, some easily noticeable patterns occur – with variations – throughout the piece. The definition of a gesture in such musical context can be relatively subjective, and based both on perceptual and more abstract compositional considerations (gestures are for instance defined by the composers in terms like *"an unstable glissando sound with reversed attack"*). The composer identified 20 different gesture classes in the score, and annotated the audio extract accordingly, cutting it into 187 labeled segments (the approx. duration of a segment is between 1 and 3 seconds). The number of elements per class varies between 1 (a gesture that appears only once in the whole training set) and 22.

B. Audio descriptors analysis

Each audio sample is analyzed and turned into a vector of signal descriptors using the IAE/pipo libraries [8], [9]. The main audio descriptors considered in this study are the Mel-frequency cepstral coefficients (MFCCs), the Fundamental Frequency, Energy, Periodicity, First-order autocorrelation coefficient (AC1), Loudness, Centroid, Spread, Skewness, and Kurtosis. We use the first 12 MFCCs and therefore work with a total of 21 descriptors, sampled according to a given analysis window size and overlap factor. Not all of these descriptors are relevant: further in this paper, we discuss strategies to select and combine them efficiently.

[1] https://www.youtube.com/watch?v=loy6JaW7L2c

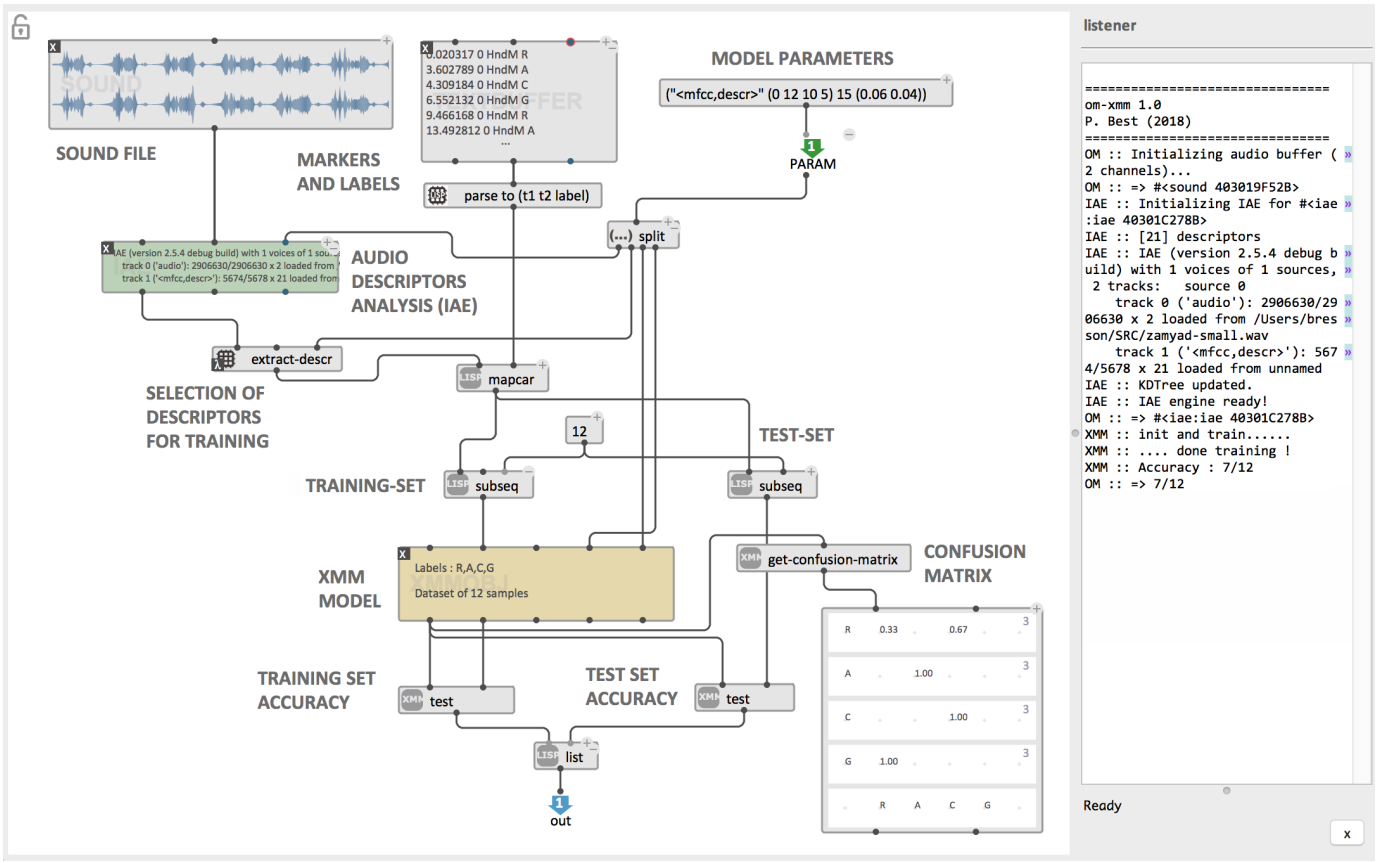

Fig. 1. Example of an OpenMusic patch and graphical interface (v. o7) including XMM model testing and validation tools applied to audio signal descriptors.

C. Model validation

We use a 10-fold cross-validation to measure our model's performances: for 10 folds, we successively train the model with 90% of the data, and compute the accuracy of recognition for the training and remaining data sets. Eventually the average of each accuracy is computed. Our measure for the model's performance thus consists in two average accuracy values: the training-set accuracy and the test-set accuracy. The final purpose of this project being to recognize gesture on new audio pieces, we aim at improving the test set accuracy first and foremost.

Confusion matrices allow to identify among the different classes of segments, the ones that are problematic to recognize. Table I reproduces part of a matrix such as the one output by the OpenMusic visual program in Fig. 1. We can observe how some classes (such as C) are easily recognized, while some others are not (e.g. K, 45% times confused with L).

D. Hyperparameters optimization

Feature selection and the tuning of hyperparameters are key elements for the recognition model performance. In addition to the choice of audio descriptors, three main parameters can vary in the model:

- The **number of hidden states** of the Markov model.

TABLE I
A CONFUSION MATRIX EXCERPT OUTPUT AFTER THE TEST PROCEDURE OF *Zāmyād*'S GESTURE RECOGNITION MODEL.

GROUND TRUTH CLASS \ PREDICTED CLASS	A	C	G	H	K	L	...
A	**0.89**	0.10					
C		**1**					
G		0.09	**0.81**	0.05	0.05		
H	0.10		0.16	**0.74**			
K					**0.54**	0.45	
L		0.05		0.19	0.05	**0.71**	

- The **two regularization coefficients** (offset added to the covariance matrices of the Gaussian distributions at each re-estimation in the algorithm):
  - *Relative*: offset relative to data-variance.
  - *Absolute*: minimum offset value.

A simple format has been defined to describe the model features and hyperparameters in the validation process. For instance: *<mfcc,descr> (0 2 15) 15 (0.1 0.05)* means:

- Audio descriptors are extracted using the *mfcc* and *descr* analysis modules (which respectively generate 12 MFCC coefficients and the vector of 9 descriptors).
- The subset of selected descriptors are the descriptors

number 0, 2 and 15 (among 21 in total).[2]
- 15 hidden states are used in the Markov model.
- The relative and absolute regularization values in the model are 0.1 and 0.05.

Because results can be highly dependent on the dataset used for validation, it is important to have an easy and automated way for users to optimize their model and fine-tune these hyperparameters. Different techniques have been applied to choose optimum values for these different parameters.

1) Brute force algorithm: This algorithm consists in an exhaustive test (10-fold cross-validation) of a whole list of hyperparameters sets. It gives a clear view of which parameters set works and which doesn't, but requires a pre-definition of all hyperparameters configurations to be tested beforehand (testing all possible combinations would take too much time).

2) Genetic algorithm: To avoid the test of an exhaustive set of hyperparameter configurations, a genetic approach has been developed: starting from a small number of hyperparameter configurations (in principle, already identified for fairly good performances), and applying small variations to them at each iteration.

The algorithm searches for optimal number of states and regularization values, and selects audio descriptors to use in the model. The range of random variations for each hyperparameter is set as follows (but adjustable for different cases and data-sets):

- Number of states: [-4 , +4];
- Relative regularization: [-0.05 , +0.05];
- Absolute regularization: [-0.005 , +0.005];

At each iteration a new audio descriptor is also randomly chosen and added or removed in the descriptors list.

After applying these random variations to the hyperparameters configurations, the algorithm evaluates the newly formed configurations and keeps the 4 best-performing (in terms of test-set accuracy) among the old and new ones.

The main limitation of this algorithm is that the search is likely to get stuck in local optima: it is efficient in fine-tuning hyperparameter configurations already known for good performances, but is not reliable in a global search context.

E. Computation time

These search and evaluation procedures for hyperparameters take time to compute. Although efforts were made to optimize them, in average it takes approximately 5 minutes on a standard personal computer to train and cross-validate a model with a given configuration of hyperparameters on the 187 elements of our dataset.

III. Results

In this section, we report elements of the model performance depending on selected hyperparameters, focusing on the selection of audio descriptors used for training and classifying gestures. Additional results and data can be found at http://repmus.ircam.fr/paco/cbmi18.

Our experiments have shown that the more descriptors are used in the model, the more hidden states are required for the HMM to perform well. We used first-order Hierarchical HMMs, with only one gaussian per state. In the following sections III-A, III-B, and III-C, the number of hidden state has been fixed manually. In section III-D, this number is searched using the genetic algorithm. In all reported results, relative and absolute regularization values were set to 0.1 and 0.05 respectively.

A. One descriptor

Table II reports the model performance when trained and run with a single audio descriptor. The model is tested with 10 hidden states.

TABLE II
ACCURACY OF THE MODEL WITH SINGLE DESCRIPTORS.

Descriptor	Test set accuracy	Training set accuracy
MFCC 1	0.21	0.30
MFCC2	**0.24**	0.29
MFCC 3	**0.24**	0.31
MFCC 4	0.17	0.27
MFCC 5	0.11	0.24
MFCC 6	0.12	0.23
MFCC 7	0.19	0.30
MFCC 8	0.12	0.12
[...]	[...]	[...]
Frequency	0.23	0.29
Energy	0.04	0.05
Periodicity	0.06	0.12
AC1	0.04	0.06
Loudness	**0.24**	0.30
Centroid	0.21	0.35
Spread	0.22	0.34
Skewness	0.14	0.22
Kurtosis	0.11	0.18

These initial results show that none of the descriptors is reliable enough by itself to accurately analyze and recognize gestures in the audio samples. We study the model performances using combined descriptors in the following sections.

B. Combination of two descriptors

Table III presents performances of models trained with selected combinations of two descriptors. Those models were tested with 15 hidden states.

Performances are visibly improved. We also note that the best combinations of two do not necessarily correspond to the combination of best-performing descriptors when tested alone (section III-A), which emphasize emerging, non-predictable aspects of the descriptors selection task. For example, *Periodicity* alone gave a 0.06 test-set accuracy, but performs a 0.44 test-set accuracy when combined with *Spread*.

[2]These numbers correspond to the 1st and 3rd MFCCs, and to the first-order autocorrelation coefficient descriptor (AC1).

TABLE III
ACCURACY OF THE MODEL WITH COMBINATIONS OF 2 DESCRIPTORS.

Descriptors	Test set accuracy	Training set accuracy
MFCC 1, 4	0.44	0.66
MFCC 2, Frequency	0.47	0.64
MFCC 2, Loudness	0.47	0.63
MFCC 2, Spread	0.46	0.65
MFCC 3, Spread	0.45	0.65
Frequency, Loudness	0.46	0.66
Frequency, Spread	0.45	0.63
Periodicity, Spread	0.44	0.56
Centroid, Spread	**0.52**	0.69

C. Combination of three descriptors

Table IV presents a selection of performance results from models trained and run with combinations of 3 descriptors. Those models were tested with 18 hidden states. As we can see, results are again sensibly better, and relevant combinations of descriptors become more salient (e.g. Spread, Centroid, and the first few MFCC coefficient).

TABLE IV
ACCURACY OF THE MODEL WITH COMBINATIONS OF 3 DESCRIPTORS.

Descriptors	Test set accuracy	Training set accuracy
MFCC 2, Centroid, Spread	**0.57**	0.79
Periodicity, Centroid, Spread	**0.57**	0.77
MFCC 3, 4, Spread	0.56	0.79
MFCC 2, Frequency, Spread	0.56	0.75
MFCC 2, 4, Centroid	0.56	0.76

D. Genetic algorithm results

Using the directed search of the genetic algorithm allowed to fine-tune relevant hyperparameter configurations found previously. Table V shows the four best hyperparameter sets obtained with the algorithm. For all of them, the relative and absolute regularization values converged to 0.1 and 0.05 respectively. The resulting selected sets of descriptors are far more complex than the previous combinations, and the performance in recognition is slightly improved.

TABLE V
MODEL ACCURACIES WITH HYPERPARAMETER SETS AND BEST COMBINATIONS OF DESCRIPTOR AS FOUND BY THE GENETIC ALGORITHM.

Descriptors	Hidden states	Test set accuracy	Training set accuracy
MFCC 2, 3, 4, 5, 6, 7, 12, Frequency, Energy, Periodicity, AC1	29	**0.61**	0.91
MFCC 2, 3, 4, 5, 12, Frequency, Energy, Periodicity, AC1, Loudness	21	0.60	0.87
MFCC 2, 3, 4, 5, 12, Frequency, Energy, Periodicity, AC1	22	0.60	0.86
MFCC 2, 3, 4 5, 12, Frequency, Energy, AC1	22	0.60	0.86

IV. CONCLUSION

We tested machine learning techniques designed for motion processing on audio signal descriptors, in order to recognize abstract musical gestures in audio extracts: elements of a temporal form which carry a perceptual identity, but can not be easily described using standard score descriptions.

We also developed an integrated framework including the IAE and XMM libraries in the OpenMusic computer-aided music composition and visual programming environment, which allows users (composers) to prepare data-sets, train and test HMM models, and select relevant hyperparameters.[3]

The choice of hyperparameters still requires a significant involvement from the user, and is highly dependent on the datasets and types of gestures to be recognized. The work accomplished so far facilitates a workflow for such hyperparameter optimization, thanks to built-in tools such as cross-validation and genetic algorithms.

We can not be fully satisfied yet with our model performance on the tested dataset (test-set accuracy of 0.65 at maximum) but no conclusion can be drawn from one single dataset: further work will focus on applying and improving these tools and workflow on other examples and datasets.

ACKNOWLEDGMENT

This work is carried out within the PEPS I3A support framework of the French National Center for Scientific Research (CNRS). The authors thank composer Alireza Farhang for providing the annotated gesture dataset.

REFERENCES

[1] A. Farhang, "Modelling a gesture: *Tak-Sīm* for string quartet and live electronics," in The OM Composer's Book. vol. 3, J. Bresson, C. Agon, and G. Assayag, Eds., Editions Delatour / Ircam-Centre Pompidou, 2016.

[2] R. I. Godøy and M. Leman, Eds., Musical Gestures: Sound, Movement, and Meaning. Routledge, 2010.

[3] J. Françoise, N. Schnell, and F. Bevilacqua, "A Multimodal Probabilistic Model for Gesture-based Control of Sound Synthesis," in MM'13 Proceedings of the ACM international conference on Multimedia, Barcelona, Spain, 2013.

[4] H.-K. Lee and J. H. Kim, "An HMM-based threshold model approach for gesture recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 10, 1999.

[5] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana, "Continuous realtime gesture following and recognition," in Gesture in Embodied Communication and Human-Computer Interaction, 8th International Gesture Workshop, Bielefeld, Germany, 2010.

[6] J. Bresson, D. Bouche, T. Carpentier, D. Schwarz, and J. Garcia, "Next-generation Computer-aided Composition Environment: A New Implementation of OpenMusic," in Proceedings of the International Computer Music Conference, Shanghai, China, 2017.

[7] J. Bresson, P. Best, D. Schwarz, and A. Farhang, "From Motion to Musical Gesture: Experiments with Machine Learning in Computer-Aided Composition," in MUME 2018: International Workshop on Musical Metacreation, Salamanca, Spain, 2018.

[8] N. Schnell, D. Schwarz, R. Cahen, and V. Zappi, "IAE & IAEOU: The IMTR Audio Engine," in Topophonie research project: Audiographic cluster navigation (2009-2012), R. Cahen, Ed., ENSCI – Les Ateliers / Paris Design Lab, pp. 50-51, 2012.

[9] N. Schnell, D. Schwarz, J. Larralde, and R. Borghesi, "PiPo, A Plugin Interface for Afferent Data Stream Processing Modules," in International Symposium on Music Information Retrieval, Suzhou, China, 2017.

[3]The library is open-source and available for download at: https://github.com/openmusic-project/om-xmm.