Machine Learning of Musical Gestures

Baptiste Caramiaux
Department of Computing
Goldsmiths, University of London
bc@goldsmithsdigital.com

Atau Tanaka
Department of Computing
Goldsmiths, University of London
atau@goldsmithsdigital.com

ABSTRACT

We present an overview of machine learning (ML) techniques and their application in interactive music and new digital instrument design. We first provide the non-specialist reader an introduction to two ML tasks, classification and regression, that are particularly relevant for gestural interaction. We then present a review of the literature in current NIME research that uses ML in musical gesture analysis and gestural sound control. We describe the ways in which machine learning is useful for creating expressive musical interaction, and in turn why live music performance presents a pertinent and challenging use case for machine learning.

Keywords

Machine Learning, Data mining, Musical Expression, Musical Gestures, Analysis, Control, Gesture, Sound

1. INTRODUCTION

Machine learning techniques have increasingly gained currency in NIME, holding significant potential to better understand performers' physical movements in controlling musical processes. This understanding can aid in the design of enhanced interaction between the performed gesture and sound synthesis, ultimately enhance musical expression. Machine learning can offer a number of solutions to confront fundamental challenges in NIME instrument design. It can provide statistical methods to extract information from digital representations of gestures created by interfaces and sensor systems. It can help focus the scope of control to make live performance mappings more cogent and meaningful.

Recent advances in machine learning bring real time performance, robustness and invariance in modeling data, making ML increasingly capable of analyzing live, expressive gestural input. We present ML techniques that are becoming increasingly popular in interaction design and look at their applicability to NIME.

The aim of this paper is two-fold. First it intends to give the non-specialist reader a brief introduction on machine learning principles, focusing on two common ML tasks, classification and regression. Second, it presents a review of the literature in current NIME–ML research to introduce the techniques and algorithms in use, and the kinds of musical tasks that are studied. We end by offering perspectives on the relevance of ML to NIME, pointing out the key challenges that the NIME context, one of live, realtime performance, poses for ML. We supplement this paper with a web tutorial[1] and hope that these resources will serve as a gathering point for further discussion in the application of machine learning techniques to expressive, musical performance.

The next section (Section 2) starts with a preamble on machine learning. Section 3 presents a review of NIME literature that incorporates ML. We then report on existing tools that allow musicians and composers to work with ML in preparing musical performances (Section 4). Finally, we conclude with a discussion offering perspectives for future work in Section 5.

2. MACHINE LEARNING

Machine Learning (ML) is a body of statistical analysis methods that achieve tasks by learning from examples. The field is intricately linked to domains such as Data Mining, techniques that discover unknown structures from data, and Pattern Recognition, techniques that identify patterns within given datasets based on a likelihood matching with preexisting patterns. Machine learning methods are distinguished by comprising a learning component that allows inference and generalization. They are particularly useful in contexts where an application is too complex to be described by analytical formulations or manual brute force design, and when an application is dependent to the environment in which it is deployed [22].

ML methods have successfully been implemented in a range of real-world audio and music applications. Speech recognition systems use ML to extract words and text from a stream of spoken audio. Early systems were speaker-dependent with advancements in signal representation and analysis leading to highly robust speaker independent systems. These generative models for speech recognition can in turn be used to synthesize speech. In Music Information Retrieval (MIR), a compelling example made famous by the Shazam app is the possibility of finding a piece of music within a large database using an excerpt of music as the query. In human movement studies, machine learning has successfully been applied to automatic sign language recognition to recognize discrete letters by comparing incoming human motion with an established previously recorded alphabet. This usually involves computer vision, with a key issue being to find suitable representations that are not merely the series of images from the video.

In order to "act by learning" rather than being explicitly programmed, ML is divided into two phases: *training* which learns the data's internal structure from given sam-

NIME'13, May 27 – 30, 2013, KAIST, Daejeon, Korea.

[1] `http://baptistecaramiaux.com/blog/machine-learning/`

ples (training data); and *testing* which takes new samples (testing data) and acts, or infers decisions based on the previously learned structure.

A number of different learning strategies exist to train a program. We focus on three main types of learning: *supervised*; *unsupervised*; and *semi-supervised* (an exhaustive list can be found in [14]). In a supervised learning approach, the training data consists of pairs of input with corresponding desired output (a case where the goal is known). In an unsupervised learning approach, the training data consists only of inputs (the goal is unknown and must be learned from the data). Finally, semi-supervised approaches consider examples of pairs of inputs and desired outputs as well as examples comprising only inputs (the goal is partially known and is refined by considering more unlabeled data).

ML techniques are configured to achieve specific tasks:

- *Regression* models a function of input data to create output
- *Classification* categorizes datasets
- *Segmentation* partitions incoming data into separate regions
- *Clustering* groups objects based on similarity
- *Prediction* analyzes historical data to forecast future events

As the majority of research in NIME–ML falls under the regression and classification categories, and in the interest of space, we limit our discussion specifically to these two task types[2].

3. TASKS

3.1 Regression

3.1.1 Definition

Regression is the task that consists of modeling discrete samplings of an unknown function by learning what the function creating the samplings may be. It is based on samples of continuous input variables paired with their target continuous variables. The input–output relationship is a function that is learned by the method. Since during training a known output is provided for each input, regression uses supervised learning.

During the testing phase, the method applies the learned regression function on each new input, and assigns an output to it. Figure 1 illustrates a basic example of linear regression. In the upper portion of the figure, the gray circles are samples that are used in the training phase. Each sample is a pair of input (x-axis) and output (y-axis) variables. The line is the regression function learned from the examples of input–output associations. This function has a given shape, here a straight line. The regression method learns the parameters governing the shape of the function. The bottom half of the figure illustrates the testing phase. A new input enters the system (x-axis) and the method assigns an output value through the learned function (y-axis).

In NIME, regression can be used to interrelate gesture and sound. Incoming gesture data would be the inputs (e.g. acceleration) and sound synthesis parameters the output (e.g. amplitude). The method infers a relationship between gesture and sound represented by a regression shape. The method learns the parameters of the shape, in the case of the line its slope and translation coefficient, to fit the data. The resulting learned function will assign to each new input gesture data (e.g. accelerometer sensor value) the corresponding audio output value (e.g. amplitude level).

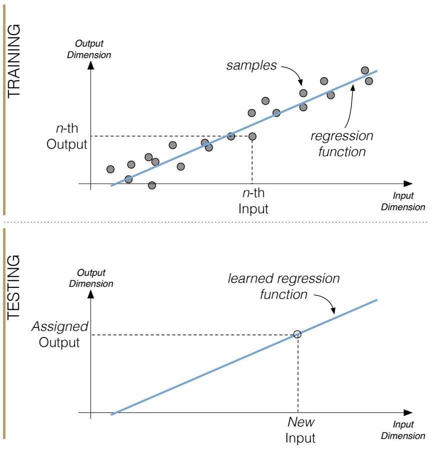

Figure 1: Regression: example of a simple linear regression. Input–Output relationship is modeled as a straight line whose parameters are learned during the training phase. During the decoding the line function is applied to a new input to produce an output.

The regression function could of course be more complex than the example depicted in Figure 1. In fact, the function is often non-linear requiring more complex learning methods. The function may also receive input in a high-dimensional space to create output values in a lower-dimensional space. In this case we apply dimensionality reduction, a method for representing high dimensional data with fewer though still meaningful dimensions. Regression techniques for dimensionality reduction that have been widely used for gesture and gesture–sound analysis include *Linear Discriminant Analysis* (LDA), which allows for finding a new data representations by maximizing distance between classes; and *Principal Component Analysis* (PCA), a statistical analysis technique that arrives at new representations by projecting data onto axes describing the data's variance (where variance is assimilated as conveying information).

3.1.2 Gesture representation

In this section we review research that uses regression as a method for gesture representation. The works described use regression techniques to reduce dimensionality of the data yet retain key information. They have been shown to be powerful tools for analysis and have been successfully implemented to analyze instrumentalist gesture as well as spontaneous movement in response to musical stimuli.

In [27], Rasamimanana et al. sought to recognize and distinguish three violin bow strokes (*Détaché*, *Martelé*, *Spiccato*) using accelerometer data of bow movement. A set of four values acceleration (min, max) and velocity (min, max) represented a bow stroke. They divided their work into two parts, 1.) representation (discussed here) and 2.) classification (discussed in Section 3.2.3). LDA to reduce the number of dimensions needed for bow stroke representation from 4 to 2. In doing, they showed that each bow stroke can in fact be efficiently represented by considering only two gestural parameters (a_{min} and a_{max}).

Young [31] extended this work to considered six bow strokes

[2] The interested reader is directed to our online resource for further details: `baptistecaramiaux.com/blog/machine-learning/`

classes. Each bow stroke was represented as the concatenation of the temporal evolutions of 8 gesture parameters (downward, lateral force; x, y, z acceleration; and 3-dimensional angular velocity). These parameters were analyzed by PCA, the concatenated data being reduced to three dimensions that could be visualized. Dimension reduction is thus achieved on both the gesture variables and their temporal profiles.

MacRitchie et al. [18] linked a pianist's arm movements to the temporal structure of a classical repertoire work. They used PCA on the gesture variables. Dimension reduction was performed, and they retained two components whose temporal evolution was shown to be linked to the metrical structure of the piece.

A similar application of PCA has been proposed by Toiviainen et al. [30]. They used PCA on motion capture data of a dancer's movements and propose a set of representations that can be visualized and that allow for relating limb trajectories to musical metrical levels.

3.1.3 Cross-modal analysis

Studies reported in the previous section involve the use of PCA or LDA for reducing the dimension of the incoming gesture data to facilitate analysis. There is a working assumption in these works to first analyze gesture (reducing its dimensionality) and then discuss its link with music. However, gestural and musical structures may be interdependent. One approach that takes this into account is to perform the analysis on both modalities, gesture and music, simultaneously.

In [3], the authors have applied machine learning in sound tracing scenarios, where gestures are made while listening to sound stimuli. They performed cross-modal regression based on Canonical Correlation Analysis (CCA) for deducing intrinsic mappings between a proposed sound and the resulting performed gesture. CCA considers two datasets and allows for extracting subsets of features from each that are the most correlated over time. They found that people favored gestures related to perceptual audio energy often using kinetic energy or acceleration as an intrinsic gestural response.

Nymoen et al. [25] extended this previous study by considering only short abstract sounds synthesized from feature profiles. CCA is applied between the sound and the participant's movements performed synchronously to the sounds. This study refined the notion of intrinsic mapping by allowing control of the sound representation through its audio features.

CCA, however, has several drawbacks. There is a fundamental assumption in correlating temporal signature of gesture and audio features: that the temporal evolution of gesture and sound are synchronous and that the relationship between gestural and sonic features remains linear over time. While previous studies revealed that these two simplifications apply for short abstract sounds, this may not be the case for all sounds and all gestures.

Ohkushi et al. [26] use Kernel-CCA linking human motion features and music features to create a music recommendation system based on human motion. A kernel can be seen as a function that sends the input variables to a higher-dimensional space where their analysis is easier (for instance in the space where their dependency is linear). In this study the problem of synchronicity is overcome by using special kernel functions.

3.1.4 Cross-modal control

Cross-modal approaches have also been proposed for the gestural control of synthesized sound. Fiebrink et al. [7] implement a high-level interface (the Wekinator) for the exploration of different regression analysis methods and parameters between a performer's movements and sound synthesis parameters. Algorithms are chosen by the user, after which regression functions are then learned by the system while the user "plays along" to the music. This shows the potential of high level toolkits, discussed in Section 4, for the application level use of ML by NIME practitioners.

3.2 Classification

3.2.1 Definition

Classification is the task of deciding to which category a dataset belongs. It pairs samples of input variables (*instances*) with labels in order to constitute categories (*classes*). Whereas regression produces continuous output, the classes reported by classification are discrete and typically comprised of integers or string labels. A function between continuous input variables and discrete labels is learned. Learning methods based on a known training set used to achieve a classification task are an example of supervised learning. (Grouping data into categories based on similarity alone is the unsupervised task of clustering.)

During the testing phase, the classification method takes a new input sample that has not been seen before and assigns it an output label.

Figure 2 illustrates the classification process. The circles are input samples. During training (above), each circle has a label (blue, grey) and classes are built based on the available samples. The testing phase takes unlabeled data (white circles, bottom-left). These unlabeled data points are assigned to classes using the classification function (bottom-right).

By way of example, imagine that the circles are data representing gestures. A vocabulary of two gestures is considered (class 1, class 2). The incoming unclassified gestures (white) are assigned to one class or the other. Note that with this simple example we can see that a gesture always belongs to either one class or the other, constituting discrete output.

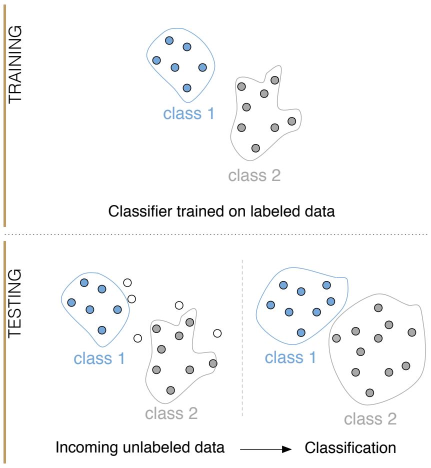

Figure 2: Classification: example of a typical classification task. A classifier is trained on the dataset comprising examples of input data (circle) paired with labels (circle background color). For each unlabeled new input (white circles), the algorithm assigns this input to a class.

A large number of classification methods exist in the ML literature. The methods differ amongst each other according to their performance (classification rate) and their characteristics (e.g. reporting at end of input or continuous classification). Classification offers a high-level symbolic representation of an input rather than an opaque continuous data stream from a given interface. In gesture based sound control, this allows characterizing qualities of gesture in ways that blind mapping does not.

3.2.2 *Multiparametric control*

The use of sensor interfaces often produces multidimensional data that contain noise and are partially redundant (i.e. data dimensions that do not add more information). In cases where information redundancy in the data is nonlinear, a common method for classification are Artificial Neural Networks (ANN).

Artificial Neural Networks connect input variables to output variables via hidden layers. Input variables are the data used for training and testing (for example gesture data examples). The output variable is the classification output. The hidden layer is an intermediary representation. There may be one or more hidden layers depending on the network topology. The main advantage of ANN is its capacity to learn non-linear relationships between input and the output.

Modler [23] used ANN for classifying hand postures based on data streamed from a glove. Classification output (i.e the hand posture recognized) is used to control a sound synthesis engine (as well as being visualized in a 3-D graphical environment). Mitchell et al. [21] used a similar model for hand posture recognition where the input data come from a glove measuring finger adduction/abduction. Classification output is used for mapping, controlling a set of performance processes.

ANN are not able to take into account temporal aspects of input (a sample at a certain instant does not dependent on previous samples). An extension of the ANN that takes into account temporal dependency by introducing short-term memory is the Recurrent Neural Network (RNN).

Early work by Lee et al. [17] made use of RNN for recognizing gestures from a a consumer dataglove, Nintendo's Power Glove and the Matthews Radio Baton. The RNN performed regression and classification with outputs from the classification process used to control a MIDI instrument.

Like ANN, RNN allows learning of the non-linear relationship between given inputs and their corresponding output. One drawback is that a complex relationship requires a large number of examples to train the network and results in a complex topology between inputs, hidden layers and outputs. Kiefer [15] proposed the use of a particular RNN topology called Echo State Network (ESN) which allows for the learning of complex non-linear dynamical systems all while maintaining a computationally efficient training phase. The model was used to learn manipulations of a squeezable interface for the control of sound.

In addition to ANN and RNN, other classification techniques have been used in music applications. Kiefer et al. [16] used Support Vector Machines (SVM, a binary linear classifier) to analyze hand positions captured by a video camera to allow the user to control sound. Gillian et al. [13] used an Adaptive Naïve Bayes Classifier (an independent feature probabilistic classifier) for recognizing control gestures to allow a pianist to change presets and scenes using free space gestures at the piano picked up by a Microsoft Kinect.

3.2.3 *Gesture analysis*

Some approaches combine classification with regression. Regression can be used to pre-process the gesture data to reduce dimensionality and to arrive at a suitable intermediary representation that can then be classified. This kind of approach divides effort across representation and recognition tasks, potentially achieving robust performance while maintaining simplicity within each sub-task.

After using LDA regression to identify the two salient features from four dimensions of data presenting violin bow strokes (above) Rasamimanana [27] then used a classifier based on the k-Nearest Neighbors (k-NN) algorithm to recognize and distinguish three different bow strokes. The regression phase reduces data dimensionality from four (vmin, vmax, amin, amax) to two and produces an intermediary representation based on minimum and maximum acceleration values. k-NN based classification assigns class neighbors around a centroid efficiently exploiting on the separation provided by the upstream LDA.

Young [31] followed up this study with six bow stroke classes and used also a k-NN taking as input the three values per stroke that resulted from a PCA (see Section 3.1.2). Similarly, the authors showed that PCA generates new representations that facilitate classification by the subsequent k-NN.

Maestre et al. [19] aimed at defining prototypical bowing movements that could be used to automate musical performance. They proposed a robust representation of bowing parameters of each note that feeds a statistical classification process based on Gaussian Mixture Models (GMM). GMM is a *generative model* meaning that it can be used for the classification of bowing movements as well as their synthesis.

Finally, Hadjakos et al. [13] proposed a probabilistic model of pianists' arm touch movements. Movement in an arm joint is described by its mean and standard deviation. A thresholding strategy is used for classification.

3.2.4 *Gesture within time*

Gestures are time-based processes. Differences in model gesture execution can reflect expressive nuance. To be able to assess temporal evolution of, and variations across, gestures leads to a greater potential for expressive interaction.

While RNN adds temporality to ANN by providing short-term memory to the network, it is not capable of taking into account longer time dependencies (long-term memory). One solution is to create a model of the gesture temporal evolution. Several statistical models exist in the ML literature (such as the generic Dynamic Bayesian Networks [24]) and often require multiple examples to efficiently train the model. Some authors argue for the design of methods that require fewer examples.

This has led to template-based techniques such as Dynamic Time Warping (DTW) that allows temporal alignment between an input time series with a template. DTW requires only a small number of examples to establish a template, and permits vectors of different lengths. It has been used by Merrill et al. [20] to allow user initiated gesture programming and subsequent recognition and sound triggering. Fujimoto et al. [10] also applied DTW to dance gesture recognition and sound mapping. The motivations are similar: designing a system allowing personalization and temporal flexibility.

DTW, however, has two major drawbacks. The method does not provide an explicit noise model, hence does not prevent errors arising from unexpected or lost observations in the incoming sequence. And although DTW gives access to the temporal evolution of the gesture, the classification decision is made at the end of input. For NIME applications, this means that the algorithm recognizes gesture only after it is over.

Some techniques allow continuous classification output during data input. Bevilacqua et al. [2] use Hidden Markov

Models (HMM) to propose a template-based method that continuously recognizes gesture in realtime as a multivariate time series. Continuous classification means that, for each new sample, the algorithm evaluates and returns a stream of likelihood weights. We can think of this as a kind of realtime DTW that is able to report the current temporal alignment (normalized position within the gesture) alongside with recognition weights at each time step.

A recent study [4] has proposed a new algorithm based on Particle Filtering inference that goes beyond the previous method with an ability to adapt classification to variation in input gesture such as continuous changes in size, speed, orientation, and offset.

4. ML TOOLKITS

In addition to the research reviewed above, there are a number of software toolkits for the application of ML in musical performance.

Wekinator[3] [6] is a toolkit for interactive machine learning based on Weka, the well known suite of ML software written in Java. Wekinator implements the following supervised methods: multilayer perceptron neural networks, k-nearest neighbors, decision tree, adaboost, support vector machines. The toolbox has been used in the preparation of a number of musical performances as well as in user-centered design workshops [8].

The SARC EyesWeb Catalog (SEC)[4] [11] proposes a suite of classification, clustering and regression methods for the EyesWeb cross-platform graphical programming environment. SEC has also been used in musical performance [12] and workshops (part of the NIME 2012 workshop session).

IRCAM's MnM toolbox[5] [1] is a library of FTM-based objects and abstractions running in Max/MSP that facilitates handling matrices for gesture–sound mapping tasks. Even if not a exclusively a ML toolkit, it contains regression methods such as PCA, CCA and classification methods such as HMM and GMM. Another Max/MSP toolbox has recently been proposed by Smith et al. [29] and includes unsupervised techniques such as: Self-Organizing Maps (SOM), Adaptive Resonance Theory (ART). The toolbox is meant to be used by non-expert artists working with music and video.

Finally, OpenCV (Open Source Computer Vision) contains ML tools for the OpenFrameworks that can also be used in musical performance[6].

5. DISCUSSION

Modern ML techniques are highly relevant, comprise useful ways to analyze musical gesture, and hold significant potential to be applied in gesture-based control of sound synthesis. They offer palpable advantages compared to direct gesture/sound mapping by providing the means to create new representations of the complex data generated by sensor systems and interactive interfaces. Despite recent advances in the field, important work remains to be done for machine learning to fulfill its promise in facilitating real time, expressive musical interaction.

Classification A number of different ML methods have been used to perform classification of musical gesture. There is a need to create evaluation models to assess their suitability for real world music performance situations. Static and dynamic neural networks are widely used in applications classifying musical gestures for multi-parametric control of sound synthesis. Through the use of adaptive basis functions, neural networks offer powerful means to create intermediary representations of complex data. This functionality can be encapsulated and used as black boxes, making these models available to composers and musicians in the form of end-user toolkits. Other methods such as linear SVM, HMM, and Particle Filtering each offer specific advantages like continuous classification and adaptation to variation. These features are particularly relevant to NIME where we would like to identify a gesture while it is being performed, and where subtle variations in the way that a gesture is executed is the key to understanding expressive nuance. However, the literature lacks robust evaluation models for these specific contexts in musical gesture classification. The continuous recognition systems reported in [2] and [4] propose quantitative evaluation: the former on synthetic data, and the latter on a standard HCI gesture vocabulary. Future research in classification of musical gestures would benefit from the use of benchmarks for performance analysis. A significant challenge remains in assessing the ability of classification systems to perform against potentially subjective criteria involved in expressive performance.

Regression Most regression methods are linear, and for the most part use Principal Component Analysis or its cross-modal extension, Canonical Correlation Analysis. These methods produce new representations of data where variables are re-projected according to their level of variation (or co-variation depending on two datasets for CCA). These approaches are a robust manner to perform dimension reduction of full body motion capture dimension and are therefore useful in the analysis of instrumentalist gesture. However, there are certain limitations to the information provided by such new representations. Constraints imposed by linearity do not allow for retrieving more subtle relationships between dependent and independent variables. For instance, if the dataset contains velocities along each axis along with the norm of the velocity, the non-linear relationship between these variables (given by the addition of squared values) will not be able to be found by linear combinations of the original variables (as PCA does). Another limitation is the use of correlation as a measure of similarity between variables. This measure is sensitive to similar variations of the data and consequently requires synchronous variables. Extension of these models are of interest for expressive musical gesture representation. In addition, it becomes mandatory when considering multimodal data (from a range of different sensors) where synchrony is not assured and linearity is an unrealistic simplification.

Design Issues Regression and classification are both examples of supervised tasks. Supervised learning is well suited for interaction design scenarios where exemplars can be provided [6]. In NIME applications this means that an interactive music system can be trained based on gesture input along with corresponding labels in order to define the make up of a digital musical instrument. In this paradigm, an instrument can be re-trained, allowing personalization and fine tuning to specific users. A performer could use a system defined by an instrument designer and customize it by giving her way of articulating the gestures specified in a composition along with the event or section labels from the piece. This instrument/composition/performance paradigm is a good conceptual fit for supervised machine learning paradigm of training/label/testing. In other musical contexts such as improvisation, however, the constraint of having to provide *a priori* labels that correspond to input data may be difficult, time-consuming, or expensive.

[3] http://wekinator.cs.princeton.edu/

[4] http://www.somasa.qub.ac.uk/~ngillian/SEC.html

[5] Included in the FTM&Co release http://ftm.ircam.fr/index.php/Download

[6] http://opencv.org/

On the other hand, a musician who has an idiosyncratic gesture vocabulary could exploit supervised ML to make a challenging performance more natural.

Other aspects of interaction design that come to light given domain specific constraints imposed by NIME applications include: understandable representations; early classification (as a realtime decision process); light training (involving few examples); and adaptability (to gesture variation). This highlights ways in which musical performance constitutes an interesting and challenging use case, and the potential contribution that NIME could make to advancing research in machine learning as a whole.

Parting Thoughts Our review deliberately did not deal with other common ML tasks such as segmentation and clustering. An extended version of this review is available in the form of an online primer (see footnote 1).

While these other task types are standard techniques in content-based Music Information Retrieval, there is currently less work in the NIME literature using segmentation and clustering. They hold potential to be applied in NIME and offer compelling challenges to be adapted to the real time and noisy data contexts typical of live musical performance. For example a segmentation method that defines running segments in continuous gesture data input could be used to control musical processes at a higher temporal level [5, 9]. Clustering could be used to create on-the-fly classes of gestures from a given input [28]. Interestingly, such an approach could foster the development of interaction design paradigms based on unsupervised learning, finding interesting applications such as for improvisation, as previously mentioned.

Finally, our article did not deal with ML from a Human-Computer Interaction point of view. Challenges in this field involve dealing with errors, approximations or convergence time made by the algorithm. In the specific context of NIME, this aspect could be critical in terms of composition and performance and must be investigated from both an artistic and a scientific perspective.

6. REFERENCES

[1] F. Bevilacqua, R. Muller, and N. Schnell. MnM: a Max/MSP mapping toolbox. In *Proceedings of the 2005 conference on New interfaces for musical expression*, Vancouver, Canada, 2005.

[2] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana. Continuous realtime gesture following and recognition. In *Embodied Communication and Human-Computer Interaction, vol. 5934 of LNCS*, pages 73–84. Springer Verlag, 2010.

[3] B. Caramiaux, F. Bevilacqua, and N. Schnell. Towards a gesture-sound cross-modal analysis. In *Embodied Communication and Human-Computer Interaction, volume 5934 of LNCS*, pages 158—-170. Springer Verlag, 2010.

[4] B. Caramiaux, N. Montecchio, and F. Bevilacqua. Adaptive Recognition of Continuous Gesture. *Pattern Recognition Letters (in review)*, 2013.

[5] B. Caramiaux, M. M. Wanderley, and F. Bevilacqua. Segmenting and Parsing Instrumentalists' Gestures. *Journal of New Music Research*, 41(1):13–29, 2012.

[6] R. Fiebrink. *Real-time human interaction with supervised learning algorithms for music composition and performance.* PhD thesis, Faculty of Princeton University, 2011.

[7] R. Fiebrink, P. R. Cook, and D. Trueman. Play-along mapping of musical controllers. In *In Proceedings of the ICMC*, 2009.

[8] R. Fiebrink, P. R. Cook, and D. Trueman. Human model evaluation in interactive supervised learning. *Proceedings of CHI 2011*, page 147, 2011.

[9] J. Françoise, B. Caramiaux, and F. Bevilacqua. A Hierarchical Approach for the Design of Gesture-to-Sound Mappings. In *Proceedings of SMC*, Copenhagen, Denmark, 2012.

[10] M. Fujimoto, N. Fujita, Y. Takegawa, T. Terada, and M. Tsukamoto. A Motion Recognition Method for a Wearable Dancing Musical Instrument. In *2009 International Symposium on Wearable Computers*, pages 11–18. Ieee, Sept. 2009.

[11] N. Gillian, R. B. Knapp, and S. O'Modhrain. A Machine Learning Toolbox For Musician Computer Interaction. In *Proceedings of NIME*, June, pages 343–348, 2011.

[12] N. Gillian and S. Nicolls. A gesturally controlled improvisation system for piano. In *1st International Conference on Live Interfaces: Performance, Art, Music*, number 3, Leeds, UK, 2012.

[13] A. Hadjakos, E. Aitenbichler, and M. Mühlhäuser. Probabilistic model of pianists' arm touch movements. In *Proceedings of NIME*, 2009.

[14] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction.* Springer Series in Statistics, second edition, 2001.

[15] C. Kiefer. A malleable interface for sonic exploration. In *Proceedings of NIME*, pages 291–296, 2010.

[16] C. Kiefer, N. Collins, and G. Fitzpatrick. Phalanger: Controlling music software with hand movement using a computer vision and machine learning approach. In *Proceedings of NIME*, Pittsburgh, PA, United States, 2009.

[17] M. A. Lee, A. Freed, and D. Wessel. Neural networks for simultaneous classification and parameter estimation in musical instrument control. In *Adaptive and Learning Systems*, pages 244–255, 1992.

[18] J. MacRitchie, B. Buck, and N. Bailey. Visualising Musical Structure through Performance Gesture. In *Proceedings of ISMIR*, 2009.

[19] E. Maestre, M. Blaauw, J. Bonada, E. Guaus, and A. Pérez. Statistical modeling of bowing control applied to violin sound synthesis. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(4):855–871, 2010.

[20] D. J. Merrill and J. A. Paradiso. Personalization, expressivity, and learnability of an implicit mapping strategy for physical interfaces. In *Proceedings of CHI*, page 2152. 2005.

[21] T. Mitchell and I. Heap. SoundGrasp : A Gestural Interface for the Performance of Live Music. In *Proceedings of NIME*, number June, pages 465–468, 2011.

[22] T. M. Mitchell. The discipline of machine learning. Technical report, Carnegie Mellon University, School of Computer Science, Machine Learning Department, 2006.

[23] P. Modler. Neural networks for mapping gestures to sound synthesis. *Trends in Gestural Control of Music*, pages 301–314, 2000.

[24] K. P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning.* PhD thesis, UC Berkeley, 2002.

[25] K. Nymoen, B. Caramiaux, M. Kozak, and J. Tø rresen. Analyzing Sound Tracings - A Multimodal Approach to Music Information Retrieval. In *ACM Multimedia – MIRUM 2011*, 2011.

[26] H. Ohkushi, T. Ogawa, and M. Haseyama. Music recommendation according to human motion based on kernel CCA-based relationship. *EURASIP Journal on Advances in Signal Processing*, 2011(1):121, 2011.

[27] N. Rasamimanana, E. Fléty, and F. Bevilacqua. Gesture analysis of violin bow strokes. *Gesture in Human-Computer Interaction and Simulation*, pages 145–155, 2006.

[28] B. Smith and G. Garnett. The Self-Supervising Machine, 2011.

[29] B. D. Smith and G. E. Garnett. Unsupervised Play: Machine Learning Toolkit for Max. In *Proceedings of NIME*, 2012.

[30] P. Toiviainen, G. Luck, and M. Thompson. Embodied meter: hierarchical eigenmodes in music-induced movement. *Music Perception*, 28(1):59–70, 2010.

[31] D. Young. Classification of common violin bowing techniques using gesture data from a playable measurement system. In *Proceedings of NIME*, 2008.