Gestural Electronic Music using Machine Learning as Generative Device

Jan C. Schacher
Zurich University of the Arts
Institute for Computer Music
and Sound Technology
Pfingstweidstrasse 96
CH-8031 Zürich, Switzerland
jan.schacher@zhdk.ch

Chikashi Miyama
ZKM | Institute for Music and
Acoustics
Lorenzstr. 19
D-76135, Karlsuhe, Germany
miyama@zkm.de

Daniel Bisig
Zurich University of the Arts
Institute for Computer Music
and Sound Technology
Pfingstweidstrasse 96
CH-8031 Zürich, Switzerland
daniel.bisig@zhdk.ch

ABSTRACT

When performing with gestural devices in combination with machine learning techniques, a mode of high-level interaction can be achieved. The methods of machine learning and pattern recognition can be re-appropriated to serve as a generative principle. The goal is not classification but reaction from the system in an interactive and autonomous manner. This investigation looks at how machine learning algorithms fit generative purposes and what independent behaviours they enable. To this end we describe artistic and technical developments made to leverage existing machine learning algorithms as generative devices and discuss their relevance to the field of gestural interaction.

Author Keywords

gestural performance, machine learning, generative behaviour, interaction

ACM Classification

J.J.5 [Arts and Humanities] Performing arts, H.5.5 [Information Interfaces and Presentation] Sound and Music Computing, I.2.8 [Artificial Intelligence] Problem Solving, Control Methods, and Search—Control theory

1. INTRODUCTION

The use of machine learning (ML) techniques in live performance of electronic music poses a number of interesting challenges and opens opportunities for different types of interaction. In this article we investigate the application of such algorithms for gesture recognition and explore some concepts for their application. First we will delineate the starting premises, before discussing the tool development phases and technical implementations. We show how in a concrete musical piece we explore these techniques using gestures acquired by motion sensing and discover a specific solution for generating interactivity through machine learning techniques. As such, this article will focus less on the technical foundation and characteristics of these models but rather attempt to address the question of the value and the impact of using them for musical performance.

NIME'15, May 31-June 3, 2015, Louisiana State Univ., Baton Rouge, LA.

At the beginning of a development for a piece that uses ML techniques, the first question to be answered should be what the role of these algorithms in the overall composition and what the expected outcome of their application can be. The answers to that will vary greatly, and further down we show one concrete solution that we found. The roles can range from that of an intelligent cue system that recognises key moments or phrases [3], to an adaptive mapping for the control of single or grouped parameters, to observation and classification of higher-level features for autonomous generative behaviour. It is this latter model that we investigate in a piece for trombone and live-electronics that provides the exemplary use-case for this article.

2. CONCEPTS

Machine learning algorithms represent a class of analysis tools that are widely used for data-mining and pattern-recognition tasks. They become increasingly relevant in the musical context both for sound and motion analysis, in particular in a realtime setting during performance. The algorithms provide powerful means for identifying salient features and classifying complex and compound patterns in the spectral and gestural domains. See [2] for a comprehensive overview of the field of machine learning for musical gesture. In the context of this project, we are dealing with capturing an instrumentalists movement, therefore relating to a field that is described with the problematic term 'musical gesture' [11].

Generative algorithms are categorised as rule-based systems that exhibit emergent qualities and that follow larger contextual constraints in order to generate their specific output [9, 4]. In their simplest form, a generative algorithm can take the shape of a random number generator that merely controls certain parameters of the sound production. More sophisticated algorithms are frequently derived from simulation-based approaches that deal with the modelling of complex natural phenomena. Such phenomena are characterised by a structural organisation, which emerges from processes of self-organisation and combines regular and chaotic properties [21]. When adapted and employed as artistic and musical tools, these algorithms might also exhibit completely artificial relationships and structures that find no correspondence in the natural world. In the context of electronic music performance, additional aspects of interactivity and autonomy come to the forefront. Here, the behaviour of the algorithm becomes related to the actions of one or several musicians and both supports and opposes the musicians' intentions and activities. Accordingly, the generative system exhibits levels of agency and thereby establishes an interaction paradigm in which two subjects enter into dialogic relationships [13]. A large

number of conditional inter-dependencies form a network of relationships that generate complicated behaviours within a field of possible actions. For true interaction to occur, the rule-based system needs to be capable of perceiving the musician's action and based on this alter its decision making processes and behavioural responses. An example of such a system is the seminal and historical case of George Lewis's Voyager system, an improvisation tool that is based on machine observation and intended as an improvisation partner for open form performance [15].

Machine learning algorithms principal purpose is to find, recognise or describe patters in large sets of data. However, and since they have the capability of observation and can therefore be applied to generative, decision-taking purposes as defined in the third category above. In this sense, the algorithm can be set to listeningto the performance of the musician in the same manner than a human co-performer, even if the 'sense' i.e. the technical channel it uses provides a much narrower view and much lower-dimensional data than the human perceptual apparatus. Derived from this 'machinic' perception, the ML algorithm's generative potential can be leveraged for creative purposes.

Fiebrink [7] et al. have worked extensively with ML software suite called Weka[1] and integrated it into a combined real-time tool called the Wekinator [6]. They explore the uses of machine learning as a creative tool, especially in the context of designing gesturally controlled musical instruments [5] and investigate machine learning as a means of generating surprising and complex machine behaviours, the differences between using machine learning in creative vs. conventional contexts, and the mechanisms for refinement and comparison of alternative models.

3. SOFTWARE TOOLS

The music technology developments in this project are embedded in a larger research context, which is investigating the meaning and affective impact of musical actions and gestures [20]. In order to bridge the gap between the disciplines of artistic practice and systematic quantifiable work, we chose to implement a software toolset that could be applied in gestural electronic music performance as well as serve as testbed for collecting experiences and knowledge about the handling and evaluation of different ML algorithms. In conjunction with this application but also for the use in other related software task within the project we implemented a simple *dataset* format definition and software library.

In order to examine different machine learning algorithms for gesture classification, a dedicated software, called the Machine Learning Workbench (in short MLWorkbench), was developed. The requirements for this workbench were specified in an iterative cycle and resulted in the data-flow scheme depicted in Figure 1.

The software receives multidimensional sensor data over the network in OSC packets [22] or as bytes through serial ports. The incoming data gets recorded and the trajectories of gestures can simultaneously be plotted for immediate inspection, at the same time specific segments of the recorded data can be extracted as samples necessary for the training of the machine learning models in supervised modes. This can be done in real-time on the fly or from a previously recorded stream of data. The gesture segments as well as the recorded raw or cooked data from the sensor devices can be stored as dataset files for later re-use and analysis.

Once a sufficient number of samples is provided, the software automatically commences the training of an ML model

[1] http://www.cs.waikato.ac.nz/ml/weka/

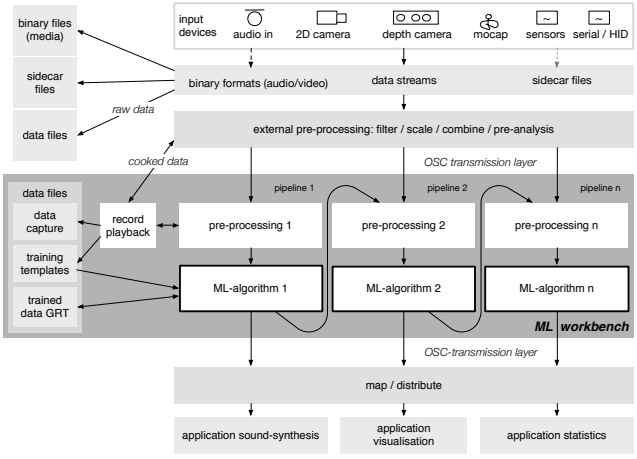

Figure 1: The MLWorkbench: data flow schema.

employing the chosen algorithm. After the training, incoming sensor data can be classified in realtime. The classification results and the likelihoods between the incoming sensor data and each trained model are visualised above and below the plotted data (see Figure 2). In order to compare and examine the differences between machine learning algorithms, the software is capable of processing incoming data in parallel through several machine learning pipelines that each can execute a distinct algorithm.

For the implementation of core machine learning routines, the Gesture Recognition Toolkit (GRT) is used [10]. In addition to the supervised and unsupervised classification algorithms provided by the GRT, we have implemented the Gesture Variation Follower (GVF) as an additional classifier [1], and we intend to include another group of ML algorithms [8] capable of real-time gesture classification, thus making the MLworkbench a versatile tool for the parallel evaluation of ML algorithms.

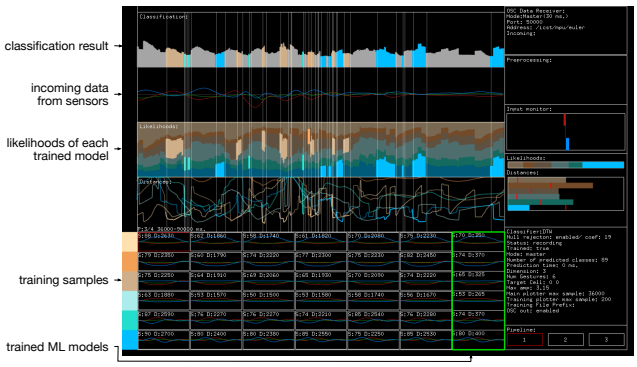

Figure 2: The MLWorkbench: application view.

The control of most of the functionalities as well as data interfacing with the software is done via OSC. Recording, playback and parametrisation of the algorithms can be controlled remotely, thus giving the MLWorkbench the ability to function as a standalone service in the background. Thus, for the use in real-time performance scenarios, the software is able to send the all the classification results to the other software, such as Max, PureData or SuperCollider, via OSC in real-time. The software is entirely programmed in C++ and utilises openFrameworks[2] [17] as its foundation, binding the GRT, GVF and all the dataset definitions as external libraries.

[2] www.openframeworks.cc

For the purposes of storing and handling data we defined a light-weight format for a data structure and file-format that doesn't have to fulfil the complex needs and requirements of all kinds of movement-related tasks [12]. The storage and retrieval of multimodal sensor data is managed by a group of container and data conversion classes that are provided by the external library written in pure C++. Apart from its role as generic container, the *dataset* class also provides functionality for converting data values. By embedding these functions in the host environment, which provides file-writing and file-reading capabilities, i.e. our workbench application, the library deals only with the *formatting* of a variety of files such as the training sample matrices and the captured data-streams in the raw or cooked formats.

4. AN ARTISTIC USE-CASE

The development of the gestural piece "Double Vortex" for trombone and live-electronics serves as the test-case about the operation and the potential of machine learning techniques in this context.

For the gestural movement-based interaction, an existing motion-sensor-pack was adapted to the trombone. It contains everything needed to measure and wirelessly transmit the movement of the instrument and provides two additional channels of analog input, which are used to connect pressure-sensors as buttons (see Figure 3, right half).

The piece is developed using a compositional framework that balances between the elements of open-form exploratory playing of sound materials, a categorisation of expressive qualities of the different playing techniques, an equal number of movement and sound instructions, and a composed block-wise model for the overall form. In this modular system a number different interaction patterns between sound and movement are explored. The clear side-by-side juxtaposition of movement and sound leads to a peculiar section where playing of sound and moving the body alternate, producing a perceptual shift between eye and ear. The addition to the sound performance instructions of impulsive movements supposed to affect sound, results in a section where body impulses add reverberation effects to the sound of the trombone (this a clear case of a simple and direct mapping and was achieved with another movement sensor strapped to the knee that is moving up and down). In other sections more complex movement patterns are overlaid to the playing and begin to influence sound treatment in various ways.

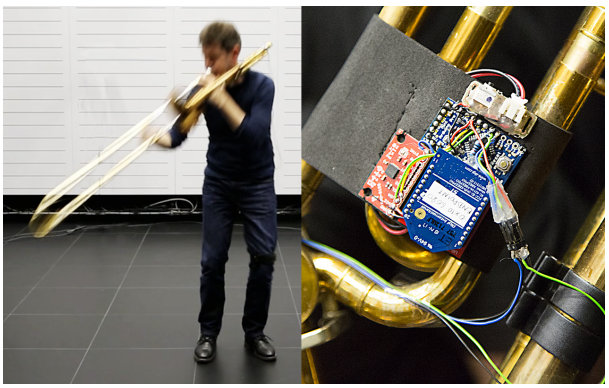

Figure 3: The trombone player, mounted sensors.

Finally, two sections are created in which three ML pipelines simultaneously observe the trombone player's movements. The top of figure 4 shows the six movement primitives that were taught to the different ML algorithms, applying in parallel the Dynamic Time Warping (DTW) twice and once the Gesture Variation Follower (GFV). Through different parametrisations, varying degrees of sensitivity to the trombone player's actions are achieved. This is visualised in the plots in figure 4 that depicts the sensor values and classification results, and the trigger points and durations as produced by the classifiers. The output of the algorithms is used in a pure live-electronic mode to record and playback sound materials performed by the trombone player himself.

In a majority of musical applications, in particular in Music Information Retrieval, ML is applied with the intention of getting a precise and repeatable result. However, since we are more interested in using the mechanism as a generative device, in line with the earlier categorisations, we decided on a different approach. To achieve this we need to explore the breaking points of the algorithms or rather the state where the output of the algorithm doesn't reproduce the original templates in a recognisable form. In addition, we also leverage the differences in reaction time between the two types of algorithm used. DTW is quite robust, however it comes too late because it can only provide an answer once a segment is finished. The GFV on the contrary, in particular in streaming mode, is relatively fragile but produces results in a more continuous manner. We use these different behaviours to operate two function of the autonomous behaviour: the first algorithm is used to trigger the response of the system via prerecorded sounds; while the the decision for recording is given to the second less predictable algorithm.

This technique results in the appearance of something akin to a 'second voice', where the 'machinic' response is clearly decoupled from the immediate action of the trombonist and gains the status of an independent dialogue partner. The structure operates as an autonomous sampling agent, providing the musician with unforeseeable musical elements that have the characteristic of calls rather than responses. The trombone player's movement remain the source for this additional musical material and activity but the direct correlation is no longer perceivable, neither for the instrumentalist nor the audience.

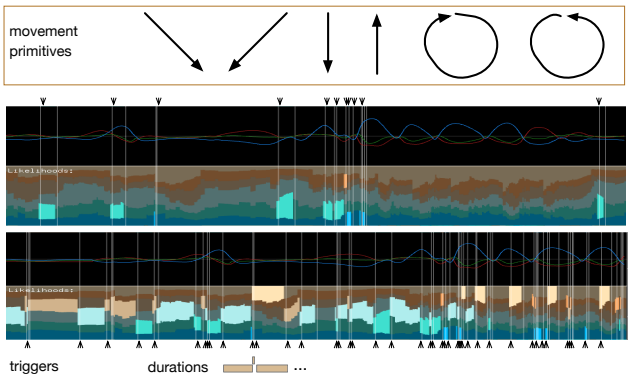

Figure 4: Six movements and two sensitivities affecting an identical movement segment.

In both cases the limitations of the algorithms are appropriated to produce an emergent quality not given by the method itself but rather by the context and linkage with which it is applied. Two areas of using real-time ML [7, 6] that is, gesture recognition and music generation, are approached by this configuration. On the level of gesture recognition and mapping, breaking the direct linkage from a gesture to a sounding result provides the perceptual independence that is necessary to ascribed agency. The use of parallel and multiply scoped observation modes [16] and the

establishment of a relationship through a 'reflexion' to the musician's playing by selectively mirroring sound-elements. On the level of gesture recognition, we follow the design rule of reduction [19] by selecting movement primitives that naturally occur in trombone playing, thus enabling the leveraging the observation of both the fine-grained as well as large scale movement patterns to enable decision-taking. This could be extended by deriving salient *dynamic features* from the data rather than spatial shapes. In this use-case a very limited model of inter-relating pattern-recognition is applied, thus building a behaviour space that can be understood with practice by the musician. This is a crucial aspect in the sense that musically it is the seamless inter-play and the feedback loop between the player and the system that generates tension, surprise and musically satisfying forms, in what Pachet calls an 'interactive reflexive musical systems' [18].

5. CONCLUSION

Applying machine learning techniques in a real-time, interactive, and gestural live-electronics performance is capable of producing artistically as well as conceptually relevant results. The challenge is to find the boundaries of behaviours and by transgressing them to establish a generative principle. The insights gained in the process of creating the tools and the specific musical piece shown here seem to pertain more to the domain of composition methods for live-electronics than engineering or HCI. They are nonetheless relevant for the fields of gestural interaction, applied machine learning and generative interactive music. It is evident that the notion of using ML techniques for music interaction is not new (cf. the liner notes in Lewis' Voyager Album [14]). But we believe that the tools and methods available today considerably alter and extend their impact on musical practice and think our application shows an example of that.[3]

6. ACKNOWLEDGMENTS

This investigation is carried out within the 'Motion Gesture Music' Project at the Institute for Computer Music and Sound Technology, Zurich University of the Arts, and is funded by the Swiss National Science Foundation Grant 100016_149345.

7. REFERENCES

[1] B. Caramiaux, F. Bevilacqua, and A. Tanaka. Beyond recognition: using gesture variation for continuous interaction. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 2109–2118. ACM, 2013.

[2] B. Caramiaux and A. Tanaka. Machine learning of musical gestures. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME 2013)*, Seoul, South Korea, 2013.

[3] A. Cont. Antescofo: Anticipatory synchronization and control of interactive parameters in computer music. In *Porc. of International Computer Music Conference*, pages 33–40, 2008.

[4] A. Dorin, J. McCabe, J. McCormack, G. Monro, and M. Whitelaw. A framework for understanding generative art. *Digital Creativity*, 23(3-4):239–259, 2012.

[5] R. Fiebrink, Trueman, and P. C. et al. Toward understanding human-computer interaction in composing the instrument. In *Proc. International Computer Music Conference*, 2010.

[6] R. Fiebrink, D. Trueman, P. R. Cook, et al. The wekinator: Software for using machine learning to build real-time interactive systems, 2011.

[7] R. A. Fiebrink. *Real-time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance*. PhD thesis, Princeton University, 2011.

[8] J. Françoise, N. Schnell, R. B. Riccardo, and F. Bevilacqua. Probabilistic Models for Designing Motion and Sound Relationships. In U. o. L. Goldsmiths, editor, *Proc. of the International Conference on New Interfaces for Musical Expression*, pages 287–292, 2014.

[9] P. Galanter. What is generative art? complexity theory as a context for art theory. In *In Proceedings GA2003 – 6th Generative Art Conference*, 2003.

[10] N. Gillian and J. A. Paradiso. The Gesture Recognition Toolkit. *The Journal of Machine Learning Research*, 15(1):3483–3487, 2014.

[11] A. R. Jensenius. To Gesture or Not? An Analysis of Terminology in NIME Proceedings 2001-2013. In U. o. L. Goldsmiths, editor, *Proceedings of the International Conference on New Interfaces for Musical Expression NIME14*, London, UK, June 30–July 03 2014.

[12] A. R. Jensenius, N. Castagné, A. Camurri, E. Maestre, J. Malloch, and D. McGilvray. A Summary of Formats for Streaming and Storing Music-Related Movement and Gesture Data. In *Proceedings of ENACTIVE/07 4th International Conference on Enactive Interfaces*, Grenoble, France, November 19th–22th 2007.

[13] S. Kozel. *Closer, Performance, Technology, Phenomenology*. The MIT Press, Cambridge, Massachusetts, London, England, 2007.

[14] G. Lewis. Voyager. compact disc, Album, Avant – Avan 014, 1993.

[15] G. E. Lewis. Too many notes: Computers, Complexity and Culture in Voyager. *Leonardo Music Journal*, 10:33–39, 2000.

[16] M. C. Mozer. Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multi-scale Processing. *Connection Science*, 6(2-3):247–280, 1994.

[17] J. Noble. *Programming Interactivity: A Designer's Guide to Processing, Arduino, and Openframeworks*. O'Reilly Media, Inc., 2009.

[18] F. Pachet. Enhancing individual creativity with interactive musical reflexive systems. In *Musical Creativity*, pages 359–375. Psychology Press, 2006.

[19] J. R. Quinlan. *C4.5: programs for machine learning*. Elsevier, 2014.

[20] J. C. Schacher. Corporeality, Actions and Perceptions in Gestural Performance of Digital Music. In *Proceedings of the joint International Computer Music and Sound and Music Computing Conference (ICMC-SMC)*, pages 629–636, Athens, Greece, 2014.

[21] J. C. Schacher, P. Kocher, and D. Bisig. The Map and the Flock – Emergence in Mapping with Swarm Algorithms. *Computer Music Journal*, 38(3), 2014.

[22] M. Wright. Open sound control: an enabling technology for musical networking. *Organised Sound*, 10(03):193–200, 2005.

[3] Link to the MLWorkbench software, additional documentation and supporting files: `http://mgm.zhdk.ch/`