Real-time Interaction with Supervised Learning

Rebecca Fiebrink
Princeton University
Department of Computer Science
Princeton, NJ 08540 USA
fiebrink@princeton.edu

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.
ACM 978-1-60558-930-5/10/04.

Abstract

My work concerns the design of interfaces for effective interaction with machine learning algorithms in real-time application domains. I am interested in supporting human interaction throughout the entire supervised learning process, including the generation of training examples. In my dissertation research, I seek to better understand how new machine learning interfaces might improve accessibility and usefulness to non-technical users, to further explore how differences between machine learning in practice and machine learning in theory can inform both interface and algorithm design, and to employ new machine learning interfaces for novel applications in real-time music composition and performance.

Keywords

Machine learning, interface design, music performance.

ACM Classification Keywords

H.5.2 [User Interfaces]: Graphical User Interface; I.2.6 [Artificial Intelligence]: Learning.

General Terms

Human factors, Design, Experimentation.

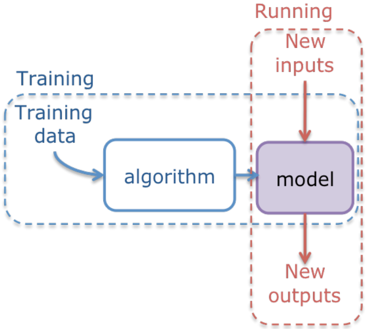

figure 1. The training and running phases of applying a machine learning algorithm to map from inputs to outputs

Introduction

Machine learning provides a powerful set of tools for “teaching” computers about relationships in data. In real-time systems, one application of supervised learning algorithms is to teach computers about the relationship between an input data signal and the desired corresponding output signal, by providing example pairs of inputs and outputs. For example, to create a gestural controller from a sensor glove, each training example pair might consist of the sensor data (the input) matched with either a continuous control signal or a discrete gesture label (the output). In a “training” step, the algorithm creates a model (or “mapping”) of the input/output relationship; this model can then predict output values for new inputs (Fig. 1). Supervised learning can be very useful when a problem requires the computer to understand a complex input/output relationship, or when it is impractical for a user to explicitly specify that relationship in code.

I am interested in building systems that allow users to interact with the entire supervised learning process, from the creation of the training dataset to the building, evaluating, and refining of the trained models. In particular, I focus on the application domains of music composition and performance, where users possess musical expertise but may lack knowledge of computer programming or machine learning. In building an interactive machine learning system and using it with performing musicians and composers, I have come to believe that there are important unanswered questions surrounding how to better facilitate the application of machine learning algorithms in real-life contexts, which are neither addressed by existing research in HCI nor in machine learning.

HCI and machine learning

My interest is informed by recent work illustrating the importance of interface design for applied machine learning systems and investigating practical benefits of human intervention beyond the mere selection of an algorithm and its parameters. Talbot et al. [5] have created an ensemble learning GUI that allows users to manipulate the learning process in response to visual feedback of component classifiers’ performance. Prior work by Fails and Olsen [1] describes an interface for interactively creating training data for a supervised learning-based computer vision system.

My work is also inspired by the impact of the Weka [6] machine learning software on domains from analysis of recorded music to bioinformatics (a Google scholar search reveals 8772 papers citing the Weka textbook). Weka’s GUI allows users to apply many standard machine learning techniques to their own data, without needing to write code or deeply understand the mathematics of the algorithms. However, the Weka GUI only allows interaction with some stages of the learning process, and it does not work with real-time data signals.

Completed and proposed research

Choosing and reviewing an application domain

The domain of real-time computer music performance is rich with potential and existing machine learning applications, such as building software for adaptively accompanying human musicians, or for generating mappings from input gestures to output sounds in the creation of new musical instruments [4]. In some problems such as pitch identification, the goal is high classification accuracy; in others, such as prototyping of gestural control systems, creative exploration may be more important. In any case, machine learning systems for music performance tend to be custom-built by researchers with engineering or computer science backgrounds. General-purpose graphical interfaces capable of handling real-time signals might greatly benefit composers and new instrument designers, just as Weka has made machine learning accessible to researchers in many other fields.

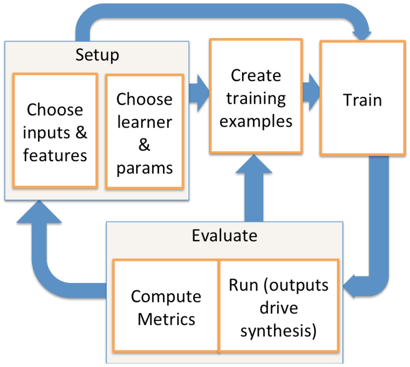

figure 2. Possible paths of interaction with Wekinator

Building and using an interface for real-time, interactive learning

I have built a software system for real-time interaction with standard supervised learning algorithms [3]. Though built on the Weka APIs, my "Wekinator" offers functionality beyond the Weka GUI: it allows a user to choose among input feature extractors, create training examples, and run a trained model on real-time inputs, within the same graphical interface as configuring and training a learning algorithm. The Wekinator thus enables a richer real-time interaction (Fig 2); for example, a user may create training examples by gesturing with an input controller, train an algorithm, then interact with the model in real-time to evaluate its performance. She may then modify the model by adding new training examples, or changing the learning algorithm or its parameters, using the same GUI.

A user may train a model for working with standard audio, video, and human interface device (HID) inputs without writing additional code or leaving the interface. Feature extractors for other inputs may be "plugged in" via UDP. The Wekinator also communicates the trained model's real-time outputs over UDP so that they may control real-time systems for sound synthesis, video processing, or other tasks.

Over the last year, I have informally observed nearly 30 musicians and composers using the Wekinator for live performance. In Fall 2009, I engaged with 7 composers in a participatory design process to improve the Wekinator and better understand how they employed it to create new music performance interfaces. I have personally used the system for experimental musical performance incorporating "on-the-fly" training of models by on-stage performers, and I am presently collaborating on an interactive gesture recognition system with the makers of a sensor-equipped string instrument bow. These experiences have brought my attention to several areas in which a better understanding of human interaction might enable more effective interfaces and systems.

Three areas where interaction matters

Interactive editing of training data

The clearest way for a user to improve a model's performance is often *not* to change the learning algorithm or its parameters, but to modify the set of training examples. For example, if an instrument identification algorithm mistakes a trumpet for a flute, the user's best recourse may be to simply add more representative trumpet training examples. Real-time interaction with training data has been explored in [1], but no general interface exists for this task. What should such an interface look like, in general or for a specific domain? I have proposed one alternative interface in [2]: assuming the model output can be sonified or visualized (easily accomplished for a music performance system), recording a user's actions while he *pretends* to control a pre-determined "output" signal can offer a painless training data-collection process.

Supporting use by non-experts

The choices of learning algorithm, parameters, and features may have significant effects on the accuracy of a trained model, but there does not exist a sufficient body of work regarding how to represent tradeoffs among these choices to non-technical users. Conveying additional practical information about algorithms (e.g., the time likely required to train) or parameterizations (e.g., the fact that boosting for more rounds may improve accuracy at the cost of training time) seems reasonable; in some cases, reformulations of an algorithm's standard parameters into human-interpretable parameters might be possible, thereby making an algorithm more "user-friendly."

Real user priorities

The priorities of users interacting with the Wekinator are often unrelated to the traditional dimensions of algorithmic analysis (bounds on time and error). Users may be concerned with strict timing guarantees (e.g., that the training phase completes in 10 seconds) or understanding their best means for improving performance. Furthermore, in our observations, users' priorities and goals often change as they experiment with a trained model. A better understanding of users' priorities may suggest both better interfaces and new avenues for algorithm analysis and design.

Further user studies

In Spring 2010, I will further investigate some of the above issues through a more formal user study with approximately 20 musicians. My work until then will consist primarily of formulating testable hypotheses from my above thoughts on interactive and real-time machine learning.

Contribution of the Work

I am interested in bringing HCI perspectives on accessibility and usability to bear on design of user interfaces for applying machine learning to real problems. These perspectives may enable broader access to and more effective use of machine learning tools, thereby making it possible for domain experts to employ these tools more successfully and to new ends. In a broader sense, this work is about recognizing that HCI perspectives are relevant even to theoretical computer science research that is far removed from human concerns: so long as that research hopes to produce algorithms that may be applied to real problems, its impact will be mediated by the extent to which it may be made understandable and usable to the broadest spectrum of potential users.

References

[1] Fails, J., and Olsen, D. Interactive machine learning. In *Proc. IUI 2003*, ACM Press (2003), 39–45.

[2] Fiebrink, R., Cook, P.R., and Trueman, D. Play-along mapping of musical controllers. *Proc. International Computer Music Conference* (2009).

[3] Fiebrink, R., Trueman, D., and Cook, P.R. A Meta-instrument for interactive, on-the-fly machine learning. In *Proc. New Interfaces for Musical Expression* (2009).

[4] Hunt, A., and Wanderley, M. M. Mapping performer parameters to synthesis engines. *Organised Sound 7* (2002), 97-108.

[5] Talbot, J., Lee, B., Kapoor, A., and Tan, D.S. EnsembleMatrix: Interactive visualization to support machine learning with multiple classifiers. In *Proc. CHI 2009,* ACM Press (2009), 1283–1292.

[6] Witten, I. H, and E. Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. 2nd ed. San Francisco: Morgan Kaufmann.