REAL-TIME HUMAN INTERACTION WITH
SUPERVISED LEARNING ALGORITHMS FOR
MUSIC COMPOSITION AND PERFORMANCE

REBECCA ANNE FIEBRINK

A DISSERTATION
PRESENTED TO THE FACULTY
OF PRINCETON UNIVERSITY
IN CANDIDACY FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE
BY THE DEPARTMENT OF
COMPUTER SCIENCE
ADVISER: PERRY R. COOK

JANUARY 2011
Abstract

This thesis examines machine learning through the lens of human-computer interaction in order to address fundamental questions surrounding the application of machine learning to real-life problems, including: Can we make machine learning algorithms more usable? Can we better understand the real-world consequences of algorithm choices and user interface designs? How can we devise more effective human workflows for building machine learning systems, enable more successful application of algorithms by machine learning novices, and ultimately make it possible in practice to apply machine learning to new problems?

The scope of the research presented here is the application of supervised learning algorithms to contemporary computer music composition and performance. Computer music is a domain rich with computational problems requiring the modeling of complex phenomena, the construction of real-time interactive systems, and the support of human creativity. Though varied, many of these problems may be addressed using machine learning techniques, including supervised learning in particular. This work endeavors to gain a deeper knowledge of the human factors surrounding the application of supervised learning to these types of problems, to make supervised learning algorithms more usable by musicians, and to study how supervised learning can function as a creative tool.

This thesis presents a general-purpose software system for applying standard supervised learning algorithms in music and other real-time problem domains. This system, called the Wekinator, supports human interaction throughout the entire supervised learning process, including the generation of training examples and the application of trained models to real-time inputs. The Wekinator is published as a freely-available, open source software project, and several composers have already employed it in the creation of new musical instruments and compositions.
This thesis also presents work utilizing the Wekinator to study human-computer interaction with supervised learning in computer music. Research is presented which includes a participatory design process with practicing composers, pedagogical use with non-expert users in an undergraduate classroom, a study of the design of a gesture recognition system for a sensor-augmented cello bow, and case studies with three composers who have used the system in completed artistic works.

The primary contributions of this work include (1) a new software tool allowing real-time human interaction with supervised learning algorithms; (2) a better understanding of the consequences of the user interface and interaction affordances in the application of supervised learning in real-time and creative problem domains; (3) a greater insight into the roles that human-computer interaction—encompassing both human-computer control and computer-human feedback—can play in the development of supervised learning systems, (4) a clearer characterization of the human-computer interaction requirements of practicing composers and instrument designers; and (5) increased understanding of how supervised learning can function as a creativity support tool. This work both empowers musicians to create new forms of art and contributes to a broader HCI perspective on machine learning practice.
Contents

Abstract .................................................................................................................. iii
List of Tables .......................................................................................................... vii
List of Figures ......................................................................................................... viii

1 Case Studies: Compositions Completed by Wekinator Users 1

1.1 Introduction ....................................................................................................... 1

1.2 Clapping Machine Music Variations by Dan Trueman 2

1.2.1 Composing the Piece with the Wekinator .................................................. 6

1.2.2 The Wekinator’s Influence on the Composition ....................................... 9

1.2.3 Evaluation of the Wekinator .................................................................. 12

1.2.4 Suggested Improvements ..................................................................... 14

1.3 The Gentle Senses by Michelle Nagai .......................................................... 15

1.3.1 Composing the Piece with the Wekinator ............................................. 17

1.3.2 The Wekinator’s Influence on the Composition ................................... 20

1.3.3 Evaluation of the Wekinator .................................................................. 21

1.4 G by Raymond Weitekamp ......................................................................... 24

1.4.1 Composing the Piece with the Wekinator ............................................. 27

1.4.2 The Wekinator’s Influence on the Composition ................................... 31

1.4.3 Evaluation of the Wekinator .................................................................. 33

1.5 Discussion ......................................................................................................... 35

1.5.1 How the Wekinator was Used .................................................................. 35
<table>
<thead>
<tr>
<th>1.5.2</th>
<th>Useful Features of the Wekinator</th>
<th>37</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5.3</td>
<td>Influence of the Wekinator on Composition</td>
<td>38</td>
</tr>
<tr>
<td>1.6</td>
<td>Conclusions</td>
<td>38</td>
</tr>
</tbody>
</table>

**Bibliography**

40
List of Tables

1.1 Changing Compositional Goals in CMMV ........................................... 10
1.2 Rating of the Wekinator’s Usefulness in CMMV ................................. 12
1.3 Changing Compositional Goals in The Gentle Senses .............................. 21
1.4 Rating of the Wekinator’s Usefulness in The Gentle Senses .................. 22
1.5 Laptop Instruments in G ................................................................. 27
1.6 Changing Compositional Goals in G ................................................. 32
1.7 Rating of the Wekinator’s Usefulness in G ........................................ 33
List of Figures

1.1 A Performance of CMMV ........................................ 3
1.2 The Manta Controller used in CMMV .......................... 5
1.3 The Tethered-uBlotar in Use in a CMMV Performance .... 6
1.4 The MARtLET Instrument, Performed in The Gentle Senses 16
1.5 PLOrk Students Performing G .................................. 24
1.6 The “G” Score ..................................................... 26
Chapter 1

Case Studies:

Compositions Completed by Wekinator Users

1.1 Introduction

In this chapter, we discuss case studies of three composers—a faculty member, a graduate student, and an undergraduate student—who have employed the Wekinator in the composition and performance of new musical works. In each of the following sections, we draw on informal interviews with the composers to illustrate how the Wekinator was used in their compositions, present their reflections on how the software was most useful to their work, and discuss possible improvements to the software that are suggested by their experiences. Following our discussion of the three compositions, we present a brief summary of how composers used the Wekinator, which aspects of the software were most useful to them, and how it influenced the compositions they created. We further draw on the work presented in this chapter in
our discussions of interaction in supervised learning in Chapter ?? and of the use of interactive supervised learning in creative applications in Chapter ??.

Our own role in the compositions described below was minimal. We seldom discussed users’ compositions with them as they worked, and on a few occasions, we implemented new features or fixed bugs in response to users’ requests. The compositions presented here constitute users’ own work, and they illustrate the types of projects that end-users of the Wekinator may accomplish on their own, using the version of the software that was presented in Chapter ??.

Overall, these users found the Wekinator to be a tremendously useful tool that enabled them to approach their work in new and more effective ways. Each composer employed an interactive approach to successfully building supervised learning models for use in their instruments and compositions, and they used interaction with the Wekinator to find inspiration, to take advantage of new ideas, and to work around challenges they encountered. The composers valued that the Wekinator allowed them to prioritize expression and embodiment, both during the compositional process and in the compositions they created. They expressed that the Wekinator enabled them to approach the process of composition in a new way and to create new and more expressive types of instruments.

1.2 *Clapping Machine Music Variations* by Dan Trueman

Daniel Trueman is a faculty member in Music Composition at Princeton University. He is an active composer and performer of music for laptop, the Norwegian Hardanger fiddle, and a variety of new and folk music ensembles. Trueman was a co-founder of the Princeton Laptop Orchestra, and he has created, composed for, and performed with several original computer music interfaces, including the BoSSA (Trueman and
Figure 1.1: A performance of *CMMV* by PLOrk in April 2010. Drum Machinists sit in the center of the stage, surrounded by Silicon/Carbonists, Tethered-uBlotarists, and guest percussionists Cameron Britt and Anders Astrand.

Cook [2000]. His work at Princeton has also included teaching, advising, and research activities focused on the development of new instruments and interfaces for computer music performance.

Trueman employed the Wekinator in the composition and performance of the piece *Clapping Machine Music Variations*, or *CMMV*, which is written for a variable-sized chamber ensemble of acoustic and laptop musicians. The piece was performed at Princeton by members of PLOrk on 3 April 2010, at the International Computer Music Conference by the Sideband ensemble on 5 June 2010, and by participants in the So Percussion Summer Institute on 25 July 2010. Figure 1.1 shows PLOrk students performing the piece at the April concert. Audio and video from the PLOrk and So Percussion Institute performances may be viewed online at [http://music.princeton.edu/~dan/compositions.html#cmmv](http://music.princeton.edu/~dan/compositions.html#cmmv).

Trueman’s program notes ([Trueman 2010b](http://music.princeton.edu/~dan/compositions.html#cmmv)) for the piece read: “At the core of *Clapping Machine Music Variations* is a pair of laptop-based Drum Machinists. Surrounding this duo is an assortment of other instruments, some clearly defined laptop-based instruments, others more variable and traditional in type. *CMMV* takes specific inspiration from works by Steve Reich, Györgi Ligeti and Björk. In particular,
the drum-machine algorithm was initially designed to mimic certain rhythmic processes in the Ligeti Études pour Piano, processes which also coincidentally generate the rhythmic pattern for Reich’s Clapping Music (this should come as no surprise, as both composers were deeply influenced by traditional African rhythms); this algorithm is then used to generate variations on the original Clapping Music pattern, variations that are explored over the course of CMMV. More generally inspiring are pieces like Riley’s In C, and Andriessen’s Worker’s Union, where some things are specified, other things are not, and anyone can join the party.”

CMMV is written for three sets of human performers: “Drum Machinists,” “Silicon/Carbonists,” and “Any Instrumentalists.” The two Drum Machinists employ a laptop GUI and a MIDI keyboard to control the delay-line-driven rhythmic patterns and drum sample instrumentation of a ChucK drum machine. The Silicon/Carbonists manipulate timbral and dynamic properties of sound produced by a granular sampling synthesis patch, also written in ChucK. The piece requires an even number of at least two Silicon/Carbonists. The Any Instrumentalists can play any instrument, and they are also matched in pairs.

The Wekinator was used in the creation of the instruments played by the Silicon/Carbonists during two of the performances, and of the “Tethered-uBlotar” instruments played by the Any Instrumentalists in all three performances to date. In the PLOrk performance, each Silicon/Carbonist employed the 3Dconnexion SpaceNavigator interface, pictured in Figure ?? on page ??, to drive six Wekinator neural networks controlling continuous parameters of the synthesis patch. These networks were trained by Trueman. In the ICMC performance, two performers familiar with the Wekinator (Rebecca Fiebrink and Jeff Snyder) used the Wekinator to train a new set of neural networks for controlling the same synthesis parameters using Snyder’s Manta interface, pictured in Figure 1.2. Trueman modified the Silicon/Carbon map-
Figure 1.2: The Manta touch controller, created by Jeff Snyder [2010] and used in the second performance of CMMV. Each hexagon senses the surface area under a performer’s touch, and programmable LEDs beneath the surface provide feedback to the performer.

ping again in the So Percussion Summer Institute performance, this time creating an explicit mapping between a Wacom tablet and the synthesis parameters.

The “Tethered-uBlotar” instrument was created by Trueman to be capable of controlling eleven synthesis parameters of the uBlotar algorithm using the GameTrak Real World Golf “tether” controller, pictured in Figure ?? on page ???. This USB controller, which was used frequently by composers in the participatory design process in Chapter ??, contains two strings that a performer can pull out of the base. The controller measures the x-, y-, and z-coordinates of the two string handles in 3D space, and the Wekinator’s built-in HID feature extractor (Section ?? on page ???) is able to use these six features as input into its supervised learning models. The uBlotar synthesis algorithm, described in Stiefel et al. [2004], employs a physical model capable of producing flute-like and electric guitar-like sounds, and it was used by composers in the participatory design process in Chapter ???. Here, Trueman
used the Wekinator to train a set of neural networks to control eleven continuous uBlotar parameters affecting the sound’s volume, timbre, vibrato, feedback, sustain, and distortion. Figure 1.3 shows a performer playing the Tethered-uBlotar in the PLOrk performance of *CMMV*. As the figure shows, the performance gestures for this instrument were capable of being quite dramatic and large in scale.

Our discussion of *CMMV* draws on e-mail correspondence with Trueman in October 2010, as well as on his published paper describing the piece and his compositional process ([Trueman](#)2010a).
1.2.1 Composing the Piece with the Wekinator

Background

Trueman was actively involved in the participatory design process described in Chapter ???. During that process, he experimented extensively with the Wekinator, and he built many models using the SpaceNavigator and tether controllers, and using variants of the Silicon/Carbon and uBlotar synthesis patches. His experiences during that process informed both many changes to the Wekinator software and the development of the instruments he created for CMMV.

Trueman’s interests include creating new sensor-based musical instruments, repurposing commercial hardware and built-in laptop capabilities as control interfaces for performing computer music, and composing and performing pieces for these new instruments. He has been building and performing with new musical controllers since the mid-1990s, and he has significant expertise in using ChucK and Max/MSP to create mappings for these instruments and write software for many other aspects of his compositions. Prior to beginning work on CMMV, Trueman characterized his familiarity with machine learning as a “2” and his familiarity with the Wekinator as a “4” on a scale from 1 (“Not at all familiar”) to 5 (“Extremely familiar”).

Asked about the motivation for using the Wekinator in CMMV, Trueman wrote, “I had two instruments in mind for the piece that posed problems for explicit mapping approaches (both had many control features and many synthesis parameters) and lent themselves nicely to the implicit [i.e., generative, not explicit mapping] approach of the Wekinator.”

Using the Wekinator

Trueman characterized his use of the Wekinator as initially focused on experimentally building many models to find out what he liked, then ultimately focusing more on
refining a few models over a longer period of time. He trained the Tether-uBlotar and SpaceNavigator Silicon/Carbon models late in the composition process, and he distributed final versions of the trained models to performers at the beginning of the rehearsal process. He did not change the models after rehearsals began. He did, however, continue to experiment with alternative controllers for the Silicon/Carbon instrument through the subsequent performances of the piece, finding that he was unsatisfied with the SpaceNavigator instrument.

Trueman used the Wekinator to craft expressive instruments that sonically fit into the overall piece, that lent themselves to playing using subtle gestures, and that provided challenges to the performer while maintaining a predictability of control: “For this piece, I wanted an expressive instrument within a narrow sound-range. I wanted to be able to make small, subtle moves, and have them reflected appropriately in the sound. I also wanted to have, at least in part, continuity, so that most of the time there wouldn’t be sudden changes in the sound world. However, I did find that I really liked having ‘tipping points,’ gestural areas where the model changed suddenly; these continue to be quite fun and interesting to negotiate. However, if there are too many of these, and not enough continuity, the instrument begins to feel random, and I prefer to have consistency and reproducibility in behavior.”

In performances of CMMV, the performers launched the software component of their instruments, which contained Trueman’s trained models, by running a shell script. This script launched the Wekinator using the command line options described in Section ?? on page ?? to automatically load the Feature Configuration and trained Learning System, begin running the Wekinator to produce output synthesis parameters from the extracted gestural features, and minimize its GUI. Performers therefore did not interact at all with the Wekinator GUI.
**Interactions During Model Creation**

Trueman used playalong recording almost exclusively to create the training data for the models, and he developed his own strategy for using the Wekinator to discover the sounds that he wanted to use in the piece and to sculpt the behavior of the trained models. He writes: “I had the most success with playalong situations, where I refined a playalong score to outline the sonic extremes of the instrument I was interested in playing, and where I then practiced playing along before creating a data set.” Trueman often deleted all the training data and recorded new training examples from scratch. As he played with the models and discovered more about the sounds he wanted to use, he performed subsequent iterations of of playalong recording using modified parameter values in order to narrow the instrument’s initially broad sonic range down to a “more specific sound-world.” Additionally, he sometimes used subsequent playalong iterations to provide better gestures in conjunction with the same parameter sets, “so I can playalong again, but playalong *better* than I had previously; I find I need to practice!”

He found that adding more data to existing training sets was an effective way to “refine or bring out a particular feature of the sound world that [he was] after,” but this action also sometimes changed the models in other ways that he liked: “The most memorable example is when I wanted to teach it to be quiet when the tethers were fully released. I added more training data to that effect, which worked well, but it also changed the way the model behaved when the tethers [were] extended, and in quite exciting ways. This was a nice, unintended result.”

Trueman also experimented with adding meta-features to the models, though he states, “I feel like I need a lot more experience with the meta-features before I really understand them and can use them effectively.” He never edited the training data or deleted subsets of the training data.
Trueman agrees that he became better at using the Wekinator as he composed, “though I still feel like a beginner. I mostly got better at anticipating what kind of training sets would work well and give me something that I found interesting to play.” He indicates that he developed the strategy of using playalong recording, described above, to become best able to create those types of training sets.

1.2.2 The Wekinator’s Influence on the Composition

The Wekinator software influenced the composition of CMMV by enabling Trueman to discover new mappings and compositional ideas, challenging him to reflect on the nature and role of the Silicon/Carbon instrument when he could not create a suitable mapping for it, and enforcing particular constraints on the type of mappings he was able to easily create using playalong recording.

Table 1.1 shows the extent to which he had initial plans for the gestures, sounds, and mappings, and the extent to which he changed his mind about those aspects of the instruments as he worked. Although he began the process of composition with some clear ideas about the interfaces and sounds he wanted to use, the final realization of the instruments was continually informed by his hands-on experimentation with the Wekinator: “After creating an initial model, I spent a fair [amount] of time exploring the model, learning what was possible and interesting. I created new models and revised models many times, constantly re-evaluating based on how the instrument responded. The final instrument is not one I would have, or could have, envisaged before beginning.” He also writes that the instruments created with the Wekinator could “themselves inspire new compositional ideas.”

Trueman was satisfied with the Tether-uBlotar instrument that he created with the Wekinator, and he has not changed it since the first performance of CMMV. However, over the course of the three performances of the piece, he has experimented with a number of different physical controllers for the Silicon/Carbon instrument.
Table 1.1: Rated level of agreement with statements about compositional goals before and during work with the Wekinator in CMMV, rated on a 5-point Likert scale from 1 = “Strongly disagree” to 5 = “Strongly agree.”

<table>
<thead>
<tr>
<th>Statement</th>
<th>Agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Before I started working with the Wekinator...</strong></td>
<td></td>
</tr>
<tr>
<td>I had a specific gestural interface in mind that I wanted to use or build for the piece</td>
<td>5</td>
</tr>
<tr>
<td>I had specific physical gestures I wanted to use in the piece</td>
<td>2</td>
</tr>
<tr>
<td>I had a specific palette of sounds I wanted to use in the piece</td>
<td>4</td>
</tr>
<tr>
<td>I had specific ideas about how I wanted gestures to control sounds in the piece</td>
<td>3</td>
</tr>
<tr>
<td><strong>While working with the Wekinator...</strong></td>
<td></td>
</tr>
<tr>
<td>I changed my mind or had new ideas about the gestural interface(s) that I wanted to use, or that I wanted to build</td>
<td>3</td>
</tr>
<tr>
<td>I changed my mind or had new ideas about the physical gestures I wanted to use in the piece</td>
<td>5</td>
</tr>
<tr>
<td>I changed my mind or had new ideas about the palette of sounds I wanted to use in the piece</td>
<td>4</td>
</tr>
<tr>
<td>I changed my mind or had new ideas about how I wanted gestures to control sounds in the piece</td>
<td>5</td>
</tr>
</tbody>
</table>

Ultimately, he found that he was not satisfied with any of the mappings he created for Silicon/Carbon using the Wekinator, and the failure of the Wekinator to produce a satisfactory instrument led him to reflect on the nature of the instrument itself and to reconsider its role in the piece. He writes, “in the end I found that an explicit approach to creating a mapping for the CMMV granulator instrument (with [the] Wacom tablet) worked the best; somehow, the synthesis parameters suggested one-to-one mappings quite naturally, and the non-linearity of the Wekinator models were a bit frustrating. That said, in the end I think that that instrument is a bit of loser for CMMV, and I’m probably going to cut it completely; it may be that the fact that it wanted an explicit mapping was a sign that it wasn’t a very good instrument.”

Although the playalong recording process was integral to Trueman’s strategy for building his instruments, he was frustrated by the way that it privileged certain types of playalong gestures and training datasets that had characteristics he felt were
contrary to his compositional goals. As discussed in Section ?? on page ??, the parameter clipboard can be used as a playalong score that sequentially plays each set of parameters for a set duration of time. It does not currently support gradual transitions from one set of parameters to the next; though Trueman requested this at the culmination of the participatory design process, it has not yet been implemented. Reflecting on this problem, he writes, “It’s hard to make playalong scores that don’t feel ‘edgy’ and ‘pointy’ with the sudden changes, and so it’s hard to make an instrument where you want to move smoothly but the playalong is so jerky.” Because it was important to him to create instruments capable of subtle and smooth changes, and to be able to practice along with the playalong scores in a natural manner, he sometimes ended up creating new clipboard-based playalong scores to work around this problem: “…with the tether-blotar, I made a fairly narrow playalong score—the parameter changes were minimal—so the pointy-ness was minimized and the playalong more closely approximated what I wanted to ultimately do with the instrument after training.”

1.2.3 Evaluation of the Wekinator

As illustrated in Table 1.1, Trueman found the Wekinator to be extremely valuable as a compositional tool. He strongly agreed that it allowed him to create mappings more easily, to create more expressive mappings, to create new kinds of music, and to approach composition in a different—and more enjoyable—way. Addressing this last characteristic, he emphasizes the usefulness of the Wekinator for rapid experimentation and inspiration during the composition process: “I like to enjoy the process of composing, however difficult it may be. I have a variety of tools and approaches towards composing that I switch between, depending on the needs of the piece and where I am creatively. The Wekinator suits this approach nicely, allowing me to create new tools quickly and on the fly. These tools tend to have personality, which suggest
Table 1.2: Rated level of agreement with statements about the Wekinator’s usefulness, in *CMMV*; rated on a 5-point Likert scale from 1 = “Strongly disagree” to 5 = “Strongly agree.”

<table>
<thead>
<tr>
<th>Statement</th>
<th>Agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Wekinator allowed me to create mappings between gesture and sound more easily than other techniques.</td>
<td>5</td>
</tr>
<tr>
<td>The Wekinator allowed me to create mappings between gesture and sound that were more musically expressive than other techniques</td>
<td>5</td>
</tr>
<tr>
<td>The Wekinator allowed me to create a kind of music that isn’t possible or that is hard to create using other techniques</td>
<td>5</td>
</tr>
<tr>
<td>Using the Wekinator allowed me to approach the process of composition in a new way</td>
<td>5</td>
</tr>
</tbody>
</table>

different kinds of music; I have always found musical instruments tremendously inspiring compositionally, and the Wekinator enables the rapid and fun construction of new instruments that can themselves inspire new compositional ideas.”

Discussing the aspects of the Wekinator that were most useful to him, Trueman highlights its support for creating instruments with a broad capacity for expressiveness, that were customized to a particular composition, and that were both accessible and musically engaging to the performer: “The Wekinator enabled me to create a physically expressive instrument that was precisely tailored to the sound-world of the piece. In particular, I was able to prescribe a narrow pitch range around the primary pitches of the piece that the tether performer could subtly navigate, creating a range of timbres that are simply impossible with acoustic instruments. In performance, the physical aspects of the tether controller combine, through the Wekinator mapping, with the blotar physical model to create an instrument that is both compelling to play and watch. It requires practice to explore and master, and is constantly revealing new possibilities, but is fairly easy to get started on; the player doesn’t need to spend lots of hours learning the instrument before joining the piece—rather the instrument teaches the player how to play it, especially when within the complete sound-world of the piece.”
Trueman further writes that the Wekinator “enables certain kinds of instruments that simply wouldn’t exist otherwise,” and “I don’t think I would have attempted to create this instrument without the Wekinator, given the complexity of the mapping problem, and the piece would have turned out differently.” Elaborating on the type of instruments that may be created with the Wekinator, he emphasizes both the ease of creating complex mappings and enabling the composer to privilege the physical “feel” of the instrument during the design process: “Without [the Wekinator], it’s either impossible or practically impossible (just too hard and tedious to do by hand) to create certain kinds of instruments, so they never come into existence. . . But, maybe there is another way of thinking about it. With [the Wekinator], it’s possible to create physical sound spaces where the connections between body and sound are the driving force behind the instrument design, and they *feel* right. It’s very difficult to do this with explicit mapping for any situation greater than 2–3 features/parameters, and most of the time we want more than 2–3 features/parameters, otherwise it feels too obvious and predictable. So, it’s very difficult to create instruments that feel embodied with explicit mapping strategies, while the whole approach of [the Wekinator], especially with playalong, is precisely to create instruments that feel embodied. I like to think of digital instrument building as a kind of choreography. Choreographers are hands-on—they like to push, pull, hold their dancers, demonstrate how things should go, in order to get what they want, and the resistance and flow of their dancers in turn feeds back into their choreography. This is quite similar to the approach that [the Wekinator] engenders, and radically different than what explicit mapping strategies enable.”

Trueman ties his extensive use of playalong recording to the importance he places on physicality, and he relates this to his experiences as a fiddler: “The score creation and practicing aspects are really important for me. It reminds me of learning to play fiddle tunes; takes a long time, and I have to really work on the feel of the bowings,
beyond just getting the notes. I play along with other fiddlers a lot, live in jam
sessions and with recordings, so the playalong notion is very familiar. I’m used to
putting in a lot of deliberate work to get these tunes in my body, and I feel like I had
the most success with [the Wekinator] when I could set up a situation where I could
practice like that to a good score, and then train from that data.”

1.2.4 Suggested Improvements

Asked what new features of the Wekinator might be most helpful to him in the future,
Trueman reiterated his wish for parameter smoothing in the playalong parameter
clipboard, to give him a greater freedom to practice smoothly changing gestures along
with playalong scores, and to create training datasets that more effectively captured
his intentions for the instrument.

He also expressed interest in adding interfaces within the Wekinator to support
explicit mappings. Such support would enable more efficient experimentation with
explicit alternatives to the neural network models; Trueman performed such experimen-
tation when designing mappings for the Silicon/Carbon instrument, but he had
to design the explicit mappings outside the Wekinator environment. Additionally, he
saw this as being potentially quite helpful in applying the Wekinator to controlling in-
struments that used explicit mappings for some parameters and generative mappings
for others.

1.3 The Gentle Senses by Michelle Nagai

Michelle Nagai is a graduate student in Music Composition at Princeton University.
In her work, she “creates site-specific performances, compositions, installations, radio
broadcasts, dances and other interactions that address the human state in relation-
ship to its setting” (Nagai 2010a). A composer of both acoustic and digital/computer
music, Nagai used the Wekinator throughout the conception, development, and performance of a new musical instrument, the MARtLET, and in a composition for the MARtLET called *The Gentle Senses*. *The Gentle Senses* was publicly performed on 27 April 2010 at Princeton University.

MARtLET is an acronym that stands for “Material Artifact, Responding to Light, Emitting Tones.” The MARtLET interface, shown in Figure 1.4, is a wearable piece of tree bark containing 28 light sensors. Nagai designed the MARtLET to be controlled by “[t]racing inflections of light and shadow as [she moves her] hands and arms across the surface” during performance (Nagai 2010b). She describes the relationship between the MARtLET instrument and *The Gentle Senses* as follows: “The composition was developed in tandem with the construction of the instrument itself. Although I am composing new work for the MARtLET, many of the sounds and techniques that I created for *The Gentle Senses* have persisted. In this way, the first work made for the instrument has also become the voice of the instrument (in part).”

Describing her performance technique, Nagai writes, “In performance on the MARtLET, I work with very slow, subtle gestures, sometimes shifting only a few inches away or towards the light. I use my hands to seek out the possibilities for sound control across the surface of the instrument. Depending on the performance environment, I am able to get a range of sound results from this type of movement.” Sonically, *The Gentle Senses* is “a quiet, slowly changing piece with a lot of delicate and subtle gradations of sound.” An audio excerpt of the piece may be heard at [http://michellenagai.com/Site/MARtLET.html](http://michellenagai.com/Site/MARtLET.html).

In the instrument and piece, the Wekinator is used in conjunction with several Max/MSP patches created by Nagai. The light sensor values are communicated via USB to a laptop, where they are first sent to a Max/MSP patch that applies filtering to the values. This patch is used by the Wekinator as an OSC feature extractor (see Section ?? on page ??), which sends the processed feature values to the Wekinator to
drive four multilayer perceptron neural networks. The Wekinator communicates the outputs of these neural networks via OSC to a Max/MSP synthesis patch. In that patch, they control the register (i.e., pitch range) of the generated sounds, as well as the pulse period, amplitude, and filter Q parameters of a vocoder.

The following discussion of *The Gentle Senses* draws on an in-person interview conducted with Nagai on 27 May 2010 and on e-mail correspondence conducted in October 2010.
1.3.1 Composing the Piece with the Wekinator

Background

Nagai took part in the participatory design group described in Chapter ???. During that process, she experimented with using the Wekinator for building gestural control interfaces for controlling the blotar synthesis algorithm (Stiefel et al. 2004) using interfaces including the GameTrak Real World Golf controller (Figure ?? on page ??) and the laptop’s internal motion sensor. In parallel, she worked to develop her ideas and the sensor hardware for the instrument that would become the MARtLET, and she used the Wekinator to experiment with a light sensor prototype at the very end of the participatory design process. From January 2010 until the performance in April 2010, she worked independently on the composition of her instrument and piece, using a version of the Wekinator software that incorporated the majority of the improvements resulting from the participatory design process.

Nagai had considerable prior experience working in Max/MSP before composing this piece, and she estimates that she had previously created “maybe a half dozen” pieces or instruments that involved some sort of gestural mapping before her participation in the participatory design work. She had not previously used machine learning, however, and she rated her familiarity with both machine learning and with the Wekinator as low (“2” on a scale from 1 = “Not at all familiar” to 5 = “Extremely familiar”) before beginning her work on the MARtLET.

The inspiration of the physical form of the MARtLET came to Nagai one day on a walk through Princeton, when she passed a fallen tree. She immediately knew she wanted to turn the wearable bark into an instrument, and she took it home. Nagai’s plans for using the Wekinator in this new instrument took shape during her experiences with the software in the participatory design workshop sessions: “After getting familiar with the software, I began to understand the real value of the Wekinator
in working with the MARtLET—namely, the possibility for gestural control that is non-explicit, changeable with each different performance environment and sensitive to subtle differences in gesture and interpretation.”

**Using the Wekinator**

Nagai developed the trained Wekinator models in conjunction with the development of the Max/MSP synthesis patches. She describes the composition process as entailing “lots of back and forth” between adjusting how the Max/MSP patch produced sound from the parameters and retraining the Wekinator models to adjust the types of parameters it output for different gestures. She continued to refine the Wekinator’s models “up until fairly late in the compositional process. At a certain point, just a few days before performance, I settled on a set of models and saved that and worked with it. I felt I needed time to learn the model before going on stage.” After practicing with the final set of models, she continued to experiment with modifications to the synthesis code “right up until dress rehearsal time. I was getting [Wekinator outputs] that I liked, in terms of a steady stream of suitable numbers, and then I was switching this stream of numbers into other parameters in my algorithm, or scaling it in different ways, sort of on the fly, to see if I liked the results.”

Her goal for the Wekinator was to produce models that provided her with a large expressive range while also allowing her to use a gestural vocabulary that was appropriate to the piece: “I was always searching for [models] that would provide enough sound variation, not just one or two different sounds types, but many, with interesting, unpredictable transitions in between them. I did also prefer models that responded to the most subtle movements and gestures. I didn’t like models that only responded with a few big movements.”

To build models with these characteristics, she developed a strategy of using the training data to map out the range of effects that she hoped to control, and to
associate different areas of the instrument with the different effects: “I worked with blocking light to specific areas of the instrument and sending in different processing effects (all sound sources were sine tones, with various processing added). In this way I was trying set up zones for different textural, timbral and registral effects, hoping that I could work with them somewhat independently and also in combination with one another.”

**Interactions During Model Creation**

Nagai primarily created training data by specifying sets of parameter values in the “Collect Data” subview of the Wekinator’s “Use it!” tab (see Section ?? on page ??) and recording the corresponding gestural inputs, stating that she “never was comfortable with play-along learning. I would have liked to work more with the spreadsheet and graphical editor, but time didn’t allow for that.” She frequently added more data to the training set and retrained when she wanted to change a model without starting over, but she was often unsatisfied with the outcomes: “I don’t think [adding data] was all that useful a strategy…It seems to just muddy what I was working with. I almost always was forced to delete and start from scratch at some point after adding more training data, when I lost my sense of where things were headed.” She frequently deleted the entire training data set and started over, and she only occasionally edited the training set.

Nagai experimented extensively with using feature selection to limit which light sensors influenced each sound parameter, and to reduce the sometimes significant training time required by the neural networks. As mentioned above, she also frequently experimented with how the neural network outputs were used in the synthesis patch. She did not edit the neural network architecture or change its learning parameters.
Nagai remarked that, through her interactions with the Wekinator, she learned to use it more effectively. She also learned to take measures to reduce the training time of the neural networks; training could take up to 30 minutes, presenting an obstacle to efficient work: “I learned to limit features in my training sessions so that it would take less time to train. I also learned how to retrain only a part of model. This was very helpful. And I learned how to distinguish what part of the model wasn’t working and could better focus on changing just that part.”

1.3.2 The Wekinator’s Influence on the Composition

As shown in Table 1.3, Nagai approached her work with the Wekinator without clear initial ideas about the gestures or sounds she wanted to use; instead, she developed these ideas during and through her work with the software. Commenting on the slow and subtle gestural vocabulary she used in the piece, she writes, “I’ve developed this kind of gesture, I believe, as a direct result of the responses I got from training the Wekinator. More specifically, I wasn’t getting entirely predictable results from the Wekinator using gestures that were fairly consistent and specific. So instead I developed a kind of searching, sensing performance gesture that seems to work very well with the visual element of the MARtLET and allows the Wekinator to trigger sounds with more subtlety and in a more aesthetically satisfying way.” Additionally, she wrote about the sound of the piece: The Gentle Senses turned out to be a quiet, slowly changing piece with a lot of delicate and subtle gradations of sound. I think, again, this was largely due to the kind of responses I was getting from my trainings with the Wekinator.”
Table 1.3: Rated level of agreement with statements about compositional goals before and during work with the Wekinator in *The Gentle Senses*, rated on a 5-point Likert scale from 1 = “Strongly disagree” to 5 = “Strongly agree.”

<table>
<thead>
<tr>
<th>Statement</th>
<th>Agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Before I started working with the Wekinator...</strong></td>
<td></td>
</tr>
<tr>
<td>I had a specific gestural interface in mind that I wanted to use or build for the piece</td>
<td>1</td>
</tr>
<tr>
<td>I had specific physical gestures I wanted to use in the piece</td>
<td>1</td>
</tr>
<tr>
<td>I had a specific palette of sounds I wanted to use in the piece</td>
<td>1</td>
</tr>
<tr>
<td>I had specific ideas about how I wanted gestures to control sounds in the piece</td>
<td>1</td>
</tr>
<tr>
<td><strong>While working with the Wekinator...</strong></td>
<td></td>
</tr>
<tr>
<td>I changed my mind or had new ideas about the gestural interface(s) that I wanted to use, or that I wanted to build</td>
<td>5</td>
</tr>
<tr>
<td>I changed my mind or had new ideas about the physical gestures I wanted to use in the piece</td>
<td>5</td>
</tr>
<tr>
<td>I changed my mind or had new ideas about the palette of sounds I wanted to use in the piece</td>
<td>5</td>
</tr>
<tr>
<td>I changed my mind or had new ideas about how I wanted gestures to control sounds in the piece</td>
<td>5</td>
</tr>
</tbody>
</table>

1.3.3 Evaluation of the Wekinator

As illustrated in Table 1.4, the Wekinator offered Nagai a valuable tool for creating mappings more easily, creating more expressive mappings, and creating a new kind of music and taking a new approach to composition.

Compared to the music she might have created with other techniques, Nagai valued how she was able to use the Wekinator to create music where, in her experience as a performer, “[t]here was a stronger connection between the physicality of specific gestures and the resulting sounds, more like playing a violin or some other acoustic instrument.” It was important to her that this relationship was both direct and complex; because the logical, mathematical relationship between a gesture and the perceived sound was less obvious, her attention (and that of the audience members)
Table 1.4: Rated level of agreement with statements about the Wekinator’s usefulness, in *The Gentle Senses*; rated on a 5-point Likert scale from 1 = “Strongly disagree” to 5 = “Strongly agree.”

<table>
<thead>
<tr>
<th>Statement</th>
<th>Agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Wekinator allowed me to create mappings between gesture and sound more easily than other techniques.</td>
<td>4</td>
</tr>
<tr>
<td>The Wekinator allowed me to create mappings between gesture and sound that were more musically expressive than other techniques</td>
<td>5</td>
</tr>
<tr>
<td>The Wekinator allowed me to create a kind of music that isn’t possible or that is hard to create using other techniques</td>
<td>5</td>
</tr>
<tr>
<td>Using the Wekinator allowed me to approach the process of composition in a new way</td>
<td>5</td>
</tr>
</tbody>
</table>

was more focused on the experience of the piece and not on attempting an intellectual deconstruction of the mapping function.

Nagai discusses how the Wekinator enabled a new approach to composition: “I have never before been able to work with a musical interface (i.e. the MARtLET) that allowed me to really ‘feel’ the music as I was playing it and developing it. The Wekinator allowed me to approach composing with electronics and the computer more in the way I might if I was writing a piece for cello, where I would actually sit down with a cello and try things out.”

Asked what aspects of the Wekinator were most useful in composing the instrument and piece, Nagai responds, “…[T]he ability for sound-gesture mappings that are flexible and somewhat unpredictable (within a larger pre-determined framework). This not only contributed to the composing of the piece, but helped to influence the construction of the instrument, the gestures used for performance and subsequent revisions of both of these elements.”

The most significant difficulty Nagai encountered during her use of the Wekinator was the sometimes high training time of the neural networks. Throughout the composition process, she at times used up to 15 neural networks, each with up to 28 features and several thousand training examples. In the worst case, training the
entire set of neural networks took up to half an hour or more, and she found this frustrating and disruptive. Additionally, she was distressed by not having a clear idea of how long to expect the training to take, as the progress bar displayed by the Wekinator only indicates the percentage of models that have been trained, not the progress of the currently training model. (This aspect of the Wekinator’s implementation is constrained by the fact that finer-grained information about training progress is not easily accessed from Weka). She did not feel like she had adequate information to choose whether to cancel the training or wait for it to complete. A significant amount of her work with the system therefore concentrated on figuring out how to reduce training time, in particular by using feature selection and only training a subset of models at any given time.

Nagai expressed a desire to edit the training data more often in the future, using the spreadsheet viewer or graphical editor. She found that deleting all the training data and starting over sometimes involved more overhead than she was willing to expend, while simply adding more training examples muddied up the models’ behavior in an unsatisfying way. She saw editing the data as a viable alternative that she was more comfortable trying after gaining more experience using the Wekinator: “As I got more comfortable with the software, I attempted it more. In future work with the Wekinator, this will certainly be something I do more of.”

1.4 \( G \) by Raymond Weitekamp

Raymond Weitekamp was a senior undergraduate student at Princeton in Spring 2010 when he composed the piece \( G \) for the Princeton Laptop Orchestra (PLOrk). Though a Chemistry major at Princeton, and now a graduate student in Chemistry, Weitekamp is an active electronic musician and performs under the name “Al-
Figure 1.5: PLOrk students performing $G$. Performers are following the Processing score on their laptop screens and playing the piece by tilting and hitting their laptops.

Weitekamp was a member of the PLOrk ensemble for several years, and he wrote $G$ to be performed by nine members of the group during his final semester. He describes the motivation and concept of the piece as follows: “Compositionally, my motivation for $G$ was to step outside of my mode of working with sample-based music; every instrument in the piece was synthesized in Chuck (in realtime). Conceptually, the original idea for the piece was to have each performer map their own physical gestures on the laptop to specific musical events. The idea was to use the play along learning algorithm at the beginning of the piece to classify which gestures (as read by the internal accelerometer) correspond to which sounds. The only controller for each of the instruments was the keyboard and [motion-sensor-driven] Wekinator output. The score for the piece was delivered in realtime using a 3-D Processing sketch, delivering notes in a ‘Guitar Hero’ fashion.” A screenshot of this score as it

1Altitude Sickness’s homepage and example tracks can be found at [http://altitudesickness.bandcamp.com/](http://altitudesickness.bandcamp.com/)
Figure 1.6: The Processing-driven score for Weitekamp’s G. Objects in the score float through the three-dimensional grid towards the performer, and they indicate the timing and location of laptop hits as well as the timing and position of laptop tilts.

was displayed on performers’ laptop screens appears in Figure 1.6. Figure 1.5 shows students performing G during a concert on 3 April 2010 at Princeton University; a second performance was given at Princeton on 15 May 2010. Audio and video of the 3 April performance can be accessed online at [http://www.music.princeton.edu/~nbritt/PLOrk-spring2010/](http://www.music.princeton.edu/~nbritt/PLOrk-spring2010/).

Compositionally, G is beat-oriented and tonal, and the use of the Processing score enables a style of performance that is—compared to many PLOrk pieces—less improvisatory and more rigidly structured with regard to both the temporal evolution of the piece and the roles played by each performer. G employs four virtual instruments created by Weitekamp, which are characterized by distinct synthesis
algorithms and distinct roles in the piece. These instruments were each played by two or three performers, and they are described in Table 1.5.

As the table shows, each instrument is played using the same set of laptop gestures: horizontal and vertical tilt, and physical hits to two different locations on the laptop. The composition uses the Wekinator to detect these gestures using three support vector machine classifiers. The first classifier classified performers’ hits or taps on the laptop into two discrete hit locations at the side and top of the machine, as well as into “no hit” and “tilting” states. All performers used the same set of hit classifiers. The second and third classifiers quantized horizontal and vertical tilt, respectively, into 3 or 5 discrete classes, depending on the instrument. The three classifiers’ outputs were used to control different ChucK synthesis classes depending on the performer’s instrument, as described in Table 1.5.

To perform classification, the Wekinator used the built-in motion feature extractor alongside a custom ChucK feature extractor written by Weitekamp. This custom extractor computed additional features from the laptop’s accelerometers, including multiple time-averaged derivatives and maxima. The tilt classifiers used only the built-in motion feature extractor, and the hit classifier used only the custom features.

The following discussion of \( G \) draws on an in-person interview conducted with Weitekamp on 23 April 2010 and e-mail correspondence in October 2010.

1.4.1 Composing the Piece with the Wekinator

Background

Prior to creating \( G \), Weitekamp had considerable expertise developing music performance software in ChucK and Processing. He had developed software for use by both himself and other musicians; notably, Weitekamp had previously written and publicly distributed “SmackTop,” a software package for translating a performers’ laptop “smacks” into MIDI messages \cite{weitekamp2009}. The SmackTop package was im-
Table 1.5: The four laptop instruments played in G. Columns indicate the instrument name, the ChucK classes used for synthesis, and the mapping from vertical tilt, horizontal tilt, and hits at two locations on the laptop to the parameters of the sound.

<table>
<thead>
<tr>
<th>Name</th>
<th>Synthesis Algorithms</th>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wobble Bass</td>
<td>PulseOsc</td>
<td>Vert. tilt: “Wobble frequency”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Horiz. tilt: Pulse width</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hit 1: Sidechain compressor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hit 2: “Frequency jump effect”</td>
</tr>
<tr>
<td>Lead Synth</td>
<td>PulseOsc TriOsc Saxofony</td>
<td>Vert. tilt: High-pass cutoff frequency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Horiz. tilt: Pulse width</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hit 1: Start “howl effect”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hit 2: Stop “howl effect”</td>
</tr>
<tr>
<td>Bells</td>
<td>BandedWG</td>
<td>Vert. tilt: Chorus effect mix level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Horiz. tilt: Rhythm patterns of “Fast forward” effect</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hit 1: “Fast forward” effect</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hit 2: Variation of “Fast forward” effect</td>
</tr>
<tr>
<td>Drums</td>
<td>SinOsc ResonZ Shakers Noise ModalBar</td>
<td>Vert. tilt: Select drum sounds used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Horiz. tilt: “Bitcrush effect”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hit 1: Play drum 1 of selected set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hit 2: Play drum 2 of selected set</td>
</tr>
</tbody>
</table>

implemented without using machine learning or the Wekinator, and prior to beginning work on G, Weitekamp characterized his familiarity with the Wekinator as “1” and his familiarity with machine learning as “2” on a scale from 1 (Not at all familiar) to 5 (Extremely familiar). Weitekamp did not take part in the participatory design seminar discussed in Chapter ??, and did not receive instruction on the Wekinator in the PLOrk class discussed in Chapter ?? (he was enrolled in a different section of the course, COS/MUS 316).

The motivation for using the Wekinator in G stemmed from Weitekamp’s desire for a larger gestural vocabulary than that offered by SmackTop, and a potential to customize the vocabulary in a way that was appropriate for the performers: “I was interested in the challenge of classifying different types of ‘hits’ using Wekinator. For
example, using the algorithm to classify between hitting the right and left sides of the laptop, or soft vs. hard hits. I really wanted the performers to be able to choose their own gestures to perform a fixed musical composition.” As we discuss below, he did not ultimately find it feasible to allow performers to design customized gestures, but he writes, “Though this wasn’t realized, I feel that with more preparation time it would have been possible to do play along learning as the prelude to the piece. Some of the motivation for allowing the performers gestural freedom comes from my experience performing with PLOrk. Many of the hard-coded, linear mappings between controller and instrument lead to awkward physical gestures that were either uncomfortable or unrealizable for some of the performers. As well, there were certain pieces where I felt my gestures were more of a gimmick than an expression. I wanted to develop a piece which would remain fixed musically, but allow each performer to create an individual gestural mapping that was both comfortable and expressive.”

Using the Wekinator

Weitekamp’s work with the Wekinator entailed some research to determine which custom features he might compute from the raw built-in accelerometer outputs in order to best discriminate among different hit locations. Over the semester, he developed his own set of Wekinator-trained models for hit classification in conjunction with refinements to his custom feature extractor.

As mentioned above, Weitekamp’s original plan for the piece was to have performers define their own gestures to use in performance. Though Weitekamp himself was able to use his final set of features to train a hit classifier capable of reliably discriminating among four different hit locations, he was disappointed to find that, after several weeks of rehearsal in which performers used playalong learning to train their own models, not all of his performers were consistently successful in creating classifiers that were accurate enough for use in performance. Therefore, for the final
performance, he chose to remove the playalong component of the piece. Instead, he
distributed pre-trained models to the performers a few weeks before the performance.
These models were trained by him to recognize just two hit locations, and performers
could practice with them outside of the performance to ensure they could consistently
perform hits that were distinguishable by the classifiers.

Early in the process of developing the piece, Weitekamp contacted us and ex-
pressed that he was having difficulty creating training data for the different hits. In
particular, it was hard to synchronize his hitting of the laptop precisely with a play-
along score or with manually clicking on the GUI button to start and stop recording,
so that training features extracted just before, during, and just after a laptop hit
were labeled cleanly. To help solve this problem, we implemented the OSC control
mechanism for starting and stopping recording, described in Section ?? . Weitekamp
then wrote OSC “gating” code in ChucK, which he used during training of the differ-
ent hit locations. The gating mechanism enabled recording of new training examples
only when the internal accelerometer values exceeded a hard-coded threshold (i.e.,
immediately after a laptop was hit). With this mechanism in place, Weitekamp was
able to successfully train accurate classifiers.

**Interactions During Model Creation**

When creating the training data for the models he built himself, Weitekamp used
playalong recording as well as the standard training data recording mechanism. He
focused his effort on building classifiers that “classified the different hits correctly and
accurately, and [were] not sensitive to laptop orientation.” To improve models whose
performance Weitekamp characterized as “loose” (meaning that hits that seemed
similar to the training examples did not result in accurate classifications), Weitekamp
added more training examples. Before getting the gating mechanism to work well,
he often used manual editing of the training data in the spreadsheet editor, in order
to delete all but the training data produced immediately after a hit was performed. He very frequently deleted the entire training set and started building a model from scratch, “Especially when I had to update the models or change input features, or if classification was wrong.” As mentioned above, Weitekamp experimented extensively with using feature selection to discover which of his custom features were most useful for classifying different types of hits.

Weitekamp states that he never changed the classifier algorithm or its parameters in an attempt to improve the models. He started by using a support vector machine algorithm for classification, and it just “seemed to work... and I had too many other things to optimize.”

When asked if he became better at using the Wekinator at he composed, Weitekamp replied, “Certainly. Primarily, I became better at providing consistent examples to the models. This was a hard skill to try to teach my performers... We spent a lot of time learning how to ‘teach’ the computer. I also feel that I became better at thinking ‘like a computer’; once I was able to understand which input features were important, I felt I could better prepare the input data to optimize classification.”

Weitekamp describes his approach to training the models as conscientious: “In order to train my model successfully, I needed to sit very still and give the machine consistent inputs during the play along learning.” He seemed confident that, given more rehearsal time and experience with the Wekinator, his performers could have also learned to use the Wekinator more effectively to create classifiers for their own gestures. In practice, the need to teach performers to use the Wekinator effectively ended up conflicting with the need to simultaneously teach performers how to play the piece; during the rehearsal process, performers communicated to Weitekamp that it was important to them to have access to a set of classifiers that they could use to practice their part of the piece and play effectively in rehearsal, even before they possessed the expertise to successfully create customized models for themselves.
1.4.2 The Wekinator’s Influence on the Composition

As illustrated in Table 1.6, although Weitekamp began the composition of G with specific ideas about the gestures and sounds he wanted to use in the piece, he changed his mind about these aspects of the composition during his work with the Wekinator. The gestural vocabulary he ultimately used in the piece was highly informed by the gestures that he found to be easy to classify, and that rehearsals revealed to be robust across different performers and laptops. He wrote, “I quickly found that certain types of hits were not easy to classify. For example, classifying a hit on the right vs. left side of the laptop is not trivial—most likely because it only involves one accelerometer axis. For the final performance, I used one side hit and one top hit as the training examples.”

As discussed above, the failure of his original plans to teach performers to successfully train their own models led to a significant change in the character and performance technique involved in the piece, when Weitekamp decided to have the performers use only classifiers pre-trained by him.

Additionally, Weitekamp believes that the sonic character of the piece was influenced by his experiences developing the gesture classifiers, and by testing out many different mappings to find one that was suitable: “I developed the Wekinator code well before designing the instruments or composing the final score. As a result, most of the ideas for percussive events or tilt-controlled FX were either directly or subconsciously influenced by Wekinator. Because the gesture-sound mappings had to be both robust and interesting, it took a number of iterations to develop a good combination.”

Finally, Weitekamp describes the Wekinator as an essential force motivating the constraints he chose to shape the composition. “The Wekinator provided a new tool for employing the laptop as an expressive controller. For this composition, I chose to use no additional hardware because I wanted to explore the limits of this idea.”
Table 1.6: Rated level of agreement with statements about compositional goals before and during work with the Wekinator in G, rated on a 5-point Likert scale from 1 = “Strongly disagree” to 5 = “Strongly agree.”

<table>
<thead>
<tr>
<th>Statement</th>
<th>Agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Before I started working with the Wekinator...</strong></td>
<td></td>
</tr>
<tr>
<td>I had a specific gestural interface in mind that I wanted to use or build for the piece</td>
<td>5</td>
</tr>
<tr>
<td>I had specific physical gestures I wanted to use in the piece</td>
<td>3</td>
</tr>
<tr>
<td>I had a specific palette of sounds I wanted to use in the piece</td>
<td>4</td>
</tr>
<tr>
<td>I had specific ideas about how I wanted gestures to control sounds in the piece</td>
<td>5</td>
</tr>
<tr>
<td><strong>While working with the Wekinator...</strong></td>
<td></td>
</tr>
<tr>
<td>I changed my mind or had new ideas about the gestural interface(s) that I wanted to use, or that I wanted to build</td>
<td>5</td>
</tr>
<tr>
<td>I changed my mind or had new ideas about the physical gestures I wanted to use in the piece</td>
<td>5</td>
</tr>
<tr>
<td>I changed my mind or had new ideas about the palette of sounds I wanted to use in the piece</td>
<td>4</td>
</tr>
<tr>
<td>I changed my mind or had new ideas about how I wanted gestures to control sounds in the piece</td>
<td>4</td>
</tr>
</tbody>
</table>

1.4.3 Evaluation of the Wekinator

As illustrated in Table 1.7, Weitekamp evaluated the Wekinator as being highly valuable in creating mappings more easily, creating more expressive mappings, and creating a new type of composition using a new type of compositional process. Asked to specifically describe how the Wekinator enabled a new approach to composition, Weitekamp responded, “Wekinator allowed me to turn the laptop into both a percussive and graduated controller (simultaneously). Because the gestural mapping between the inputs and outputs need not be linear, the process of sonifying a gesture was easier and ultimately lead to a more natural mapping.”

When asked what aspects of the Wekinator were most helpful in composing and performing the piece, his first response was “Every aspect, because it was integral
Table 1.7: Rated level of agreement with statements about the Wekinator’s usefulness, in G; rated on a 5-point Likert scale from 1 = “Strongly disagree” to 5 = “Strongly agree.”

<table>
<thead>
<tr>
<th>Statement</th>
<th>Agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Wekinator allowed me to create mappings between gesture and sound more easily than other techniques.</td>
<td>5</td>
</tr>
<tr>
<td>The Wekinator allowed me to create mappings between gesture and sound that were more musically expressive than other techniques</td>
<td>5</td>
</tr>
<tr>
<td>The Wekinator allowed me to create a kind of music that isn’t possible or that is hard to create using other techniques</td>
<td>5</td>
</tr>
<tr>
<td>Using the Wekinator allowed me to approach the process of composition in a new way</td>
<td>5</td>
</tr>
</tbody>
</table>

to the concept of the piece.” He then added that the OSC gating mechanism was particularly useful for recording clean training data for different laptop hits.

Even though Weitekamp was a proficient software developer, he indicates that he would not have been able to create this piece by hard-coding the analysis of the features and their mapping to sound. In his opinion, the accelerometer analysis and classification would have been exceedingly difficult, awkward, and time consuming, if not impossible, to perform by writing code.

Weitekamp continues to be excited by the Wekinator’s potential to allow computer music performers to customize their performance interfaces. Having participated in a performance of CMMV (Section 1.2), he appreciated that supervised learning could be used to create new instruments with greater expressive possibilities than simple, linear, and one-to-one mappings. At the same time, he desires to create expressive instruments that do not lose all of the transparency that such simpler interfaces offer to a performer learning how to play them, and he is interested in how the Wekinator could be used by performers to create their own mappings or to customize mappings created by a composer (e.g., by adding new data or changing the model in minor ways). One potential application of the Wekinator that he proposed involves having performers use playalong recording to construct their own training sets, while
watching a video of the composer playing along to the same sounds: this would allow
performers insight into the composer’s training data creation process, and it would
allow them the freedom to deviate from the composer’s training process for aesthetic
or practical reason (e.g., having shorter limbs).

Weitekamp composed $G$ using a version of the Wekinator that did not yet have the
graphical editor, and when he was shown the graphical editor he saw it as potentially
greatly useful to informing the user about the nature of the training data and speeding
up the model-building process. He is interested in having additional capabilities
to visualize or otherwise understand how the models were working and how a user
might most meaningfully change the learning algorithms or parameters to achieve
different results. Weitekamp has also expressed interest in making the Wekinator
more available to a wider variety of electronic musicians, through integrating it with
a future version of the SmackTop software, and by enabling it to output and receive
MIDI for easier use by musicians who do not use OSC.

1.5 Discussion

1.5.1 How the Wekinator was Used

The composers discussed in this chapter applied the Wekinator to the creation of
diverse and distinctive new musical instruments and compositions. In the works
by Trueman, Nagai, and Weitekamp, the Wekinator was used to create instruments
from commodity computer input devices, from a piece of tree bark containing custom
light-sensor circuitry, and from the laptop’s built-in sudden motion sensors. Nagai
and Trueman created instruments with continuous gesture-to-sound mappings, and
Weitekamp created instruments incorporating discrete gesture classifiers. Nagai’s in-
strument was performed solo, and Trueman’s were performed within a heterogenous
ensemble of acoustic and laptop performers; Weitekamp used the same gesture clas-
sifiers within all four different laptop instruments used in his composition for laptop ensemble. The compositional styles of these pieces also spanned a wide range, from avant-garde to electronica-inspired and rhythmic.

Each composer developed distinct strategies for interacting with the Wekinator and for integrating it into his or her larger compositional process. Nagai experimented with many different neural networks for creating mappings for her instrument, and she developed the sound synthesis component of her composition in conjunction with the development of the mappings. She only finalized her instrument’s mapping a few days before the performance of her piece. Trueman, on the other hand, gradually refined his mappings over time, and he used the Wekinator to create mappings for hardware controllers and synthesis algorithms that were relatively fixed. Trueman completed his mappings before rehearsals of his piece so that performers could learn how to play them, and through three performances of his composition, he has kept the mapping of one instrument fixed while continuing to change both the mapping and the hardware controller for the other. Weitekamp used the Wekinator to experiment with different gesture classifiers while at the same time refining and improving his feature extractor, and throughout the rehearsal period for his piece, he experimented with both allowing performers to train their own classifiers and with training classifiers himself for them to use. Both Trueman and Nagai developed strategies for creating training data that would produce mappings that balanced predictability and control with surprise and complexity, while Weitekamp developed a strategy for creating training data that resulted in classifiers whose behavior was as predictable and robust as possible.

Despite these differences, each of the instruments created by these three composers shares a common focus on the expressive use of the human body, and each composer saw the Wekinator as a valuable tool in enabling him or her to prioritize expression and physicality throughout the activities of composition, instrument-building, and performance. Nagai and Trueman both expressed that, in the process of composing
and instrument building, it was important to establish a mode of performance in which the relationship between performer gesture and sound was tightly coupled, not simple or linear, and offered a range of expressive possibilities that a performer could learn—with practice—to control in a musically sensitive manner. When the instruments they created were successful in meeting these criteria, the composers likened them to acoustic musical instruments. In other words, they used the Wekinator to create new performance interfaces that possessed certain musically essential qualities of conventional, existing instruments, but which—through their use of new controller interfaces and synthesis algorithms, and the ability to tailor their design to a particular performer or composition—transcended certain physical or compositional limitations of such instruments. For Weitekamp, the Wekinator was the inspiring force behind his attempt to create a composition wherein each performer had the freedom to design his own expressive gestural vocabulary. While this vision was not realized in the performance, his ultimate use of the software still enabled him to efficiently build a set of laptop instruments whose performance technique demanded performer-laptop interactions that were physically engaging to the performers and visually engaging to the audience.

We further discuss how the Wekinator supported this type of work, and the consequences for future use of machine learning in musical and non-musical applications, in Chapters ?? and ??.

1.5.2 Useful Features of the Wekinator

The working process of all three composers took advantage of the Wekinator’s functionality for iteratively creating and editing training data, building models, experimentally running the models in real-time, and modifying the supervised learning problem, data, and/or other aspects of their instruments or compositions in order to improve the models. Some of the aspects of the Wekinator that allowed them to
perform this work successfully included its capabilities for users to switch between these different tasks with a low amount of overhead, to efficiently provide training data, to frequently evaluate models by interacting with them in a real-time manner similar to the way in which they would be used in performance, and to perform all these tasks without the need to write code or become machine learning experts.

The Wekinator’s ability to use OSC to receive features and control messages and to send synthesis parameters was also particularly useful to all composers, and this ability enabled composers to apply the Wekinator alongside the compositional software tools they already knew how to use and preferred.

The particular learning algorithms built into the Wekinator were also useful to composers. In Chapter ??, we discuss how neural networks functioned as a useful compositional tool for composers discussed here and in Chapter ??. Support vector machines were useful to Weitekamp because they were capable of accurately classifying the input gestures of interest to him.

1.5.3 Influence of the Wekinator on Composition

Each of the composers agreed or strongly agreed that, while they were working with the Wekinator, they changed their minds about the gestures, sounds, and gesture-sound mappings that they planned to use in their instrument or composition. Occasionally, these changes of course were a result of working around barriers that the software presented to their achieving their initial goals: for example, Nagai took many measures to reduce the training time of her neural networks, including limiting the number of light sensors that influenced each sonic parameter, and Trueman ended up crafting his Tethered-uBlotar playalong score to reside within a narrow sonic range to avoid the “pointy” mappings that were most easily produced from playalong recording. Sometimes, composers discovered through their use of the Wekinator that their initial goals were simply infeasible: Weitekamp discovered, for instance, that it was
too hard for him to train all of his performers to produce consistently good classifiers given the practical and compositional constraints of his piece.

At the same time, composers often relied on their experimentation with the Wekinator to discover new sonic and gestural possibilities that they had not previously imagined, and they adapted aspects of their instruments or compositions to take advantage of these possibilities. The Wekinator also inspired composers to consider new composition and performance paradigms, such as the composition of complex mappings or the engagement of performers in the instrument building process, which would not have been possible—or which just seemed too impractical to be considered—using other tools.

1.6 Conclusions

In this chapter, we have described three new musical compositions that have been completed by Wekinator users. The success with which these users have been able to apply the Wekinator to their work demonstrates that, while various improvements could still be made to the software, the Wekinator in its current form is a useful tool for real-world applications. Furthermore, users having low familiarity with machine learning were able to apply it successfully to their work.

The diversity of ways that these users applied the Wekinator in their work also attests to its flexibility as a compositional tool. The Wekinator’s OSC capabilities, which allow it to be used in conjunction with arbitrary feature extractors, synthesis algorithms, and visualizations, were crucial to enabling these users to employ the software in conjunction with other software and hardware systems of their choosing. Different users also took advantage of different mechanisms for interactively applying supervised learning to their problems, including different mechanisms for recording and editing the training data (e.g., playalong recording, OSC-gated recording, and
editing using the spreadsheet editor), and modifying the supervised learning problem (e.g., changing the classifiers, changing feature selection, and adding meta-features).

Furthermore, the Wekinator invited users to imagine vastly different applications, from performer-customizable mappings to turning a piece of tree bark into an expressive instrument, with vastly different musical aesthetics and technical challenges. While each of these applications were different from what we had imagined when designing the software, the Wekinator was able to inspire these projects and function as a useful tool for tasks that would have been impossible or difficult for users to accomplish otherwise. Additionally, the Wekinator supported each of the composers in creating music that privileged the expressive use of the human body, during both composition and performance.

The composers discussed in this chapter have been extremely helpful in informing us about the features of the Wekinator that are important to users, the features that they would like to see added to the software, and the potential future applications they envision may be possible with the Wekinator. We intend to continue to collaborate closely with them and other Wekinator users to drive improvements to the software and cultivate ideas for future research directions.
Bibliography


NAGAI, M. 2010b. MARtLET. http://michellenagai.com/Site/MARtLET.html


