The AAAI 2017 Spring Symposium on
Designing the User Experience of Machine Learning Systems
Technical Report SS-17-04

Interactive Machine Learning for End-User Innovation

Francisco Bernardo, Michael Zbyszyński,
Rebecca Fiebrink, Mick Grierson
Goldsmiths University of London
Department of Computing
London, UK

Abstract

User interaction with intelligent systems need not be limited to interaction where pre-trained software has intelligence "baked in." End-user training, including interactive machine learning (IML) approaches, can enable users to create and customise systems themselves. We propose that the user experience of these users is worth considering. Furthermore, the user experience of system developers—people who may train and configure both learning algorithms and their user interfaces—also deserves attention. We additionally propose that IML can improve user experiences by supporting user-centred design processes, and that there is a further role for user-centred design in improving interactive and classical machine learning systems. We are developing this approach and embodying it through the design of a new User Innovation Toolkit, in the context of the European Commission-funded project RAPID-MIX.

Introduction

When considering the user experience of machine learning systems, it is important to consider the experiences of users designing, training, evaluating, and refining such systems. One relevant group of users is the "end users" of software tools in which machine learning facilitates personalisation or adaptation. Tools that afford customisation or adaptation—for instance by learning on user-provided training examples—are useful in domains including the creation of new control interfaces for people with disabilities, data monitoring applications for "quantified self" or smart homes, new interactive interfaces for music and art, and the creation of diverse sensor-driven interactions by hackers and makers.

Interactive machine learning (IML) approaches can enable such users to create and customise machine learning applications. Fails and Olsen 2003 define IML as a new machine-learning paradigm with a workflow that features rapid, iterative cycles of the user training a model, evaluating its performance, and modifying the model to improve its performance. In the simplest case, users can interact with the machine learning process by iteratively providing new training examples (e.g., examples of a human action or activity, alongside the label that a classifier should apply to that action/activity). In this paper, we describe how IML can support new types of end-user customisation, which can be understood in terms of end-user development (Lieberman et al. 2006) or of user innovation (von Hippel 2005). We describe considerations and challenges for supporting user experience in this context. We also discuss a similar context, that of system developers employing machine learning to create intelligent systems for use by others. These developers are also "users" of machine learning tools, and the tools they use influence the type of systems that can be built and the types of experiences that are possible for their end users.

We also describe how IML and user-centred design processes may inform one another. User-centred design (UCD) (Norman and Draper 1986) is an approach to design based upon the understanding of users, their tasks, and environments; it involves the user at an early stage of project development and throughout the project lifetime. In UCD, the design is driven and refined through an iterative cycle of development and user-centred evaluation. IML provides a set of mechanisms to support UCD, by making it possible to translate users' demonstrations or observations of user actions into training examples that are used to build or refine a new technology. At the same time, UCD can and should play a role in the design of interactive and classical machine learning systems, whether aimed at developers or end users.

We frame our work in the context of innovation studies, where we introduce the concept of User Innovation Toolkits (von Hippel 2001) and explain how we are appropriating it to create IML tools to support end-user innovation.

The final section of our paper describes RAPID-MIX, a design project that places next-generation machine learning tools in the hands of end users in the form of a User Innovation Toolkit (von Hippel 2001). We describe the aims of this project and the methodologies employed to understand and improve the user experiences of both end users and professional developers using machine learning to build new interactive systems.

Claims

Interactive machine learning can enable end-user customisation

As described above, IML can make it possible for end users to create customised systems. Typically, IML users iteratively adjust training data or learning parameters to "steer" a machine-learned model toward a desired behaviour. A user interface for IML may enable users who have domain expertise (but possibly no machine learning expertise) to steer models by providing new training examples (Fiebrink 2011; Fails and Olsen Jr 2003), adjusting misclassification costs (Kapoor et al. 2010), adjusting weightings of component classifiers in an ensemble system (Talbot et al. 2009), or taking other actions. This approach can be used to create simple computer vision classifiers (Fails and Olsen Jr 2003), new sensor-based interactions (Hartmann et al. 2007), customised gestural controllers including new musical instruments (Fiebrink 2011), customised alerts (Amershi et al. 2011), and potentially many other bespoke systems.

Amershi et al. (2014) describe the IML workflow as more rapid, focused and incremental than classic machine learning (CML), which can be laborious and sometimes inefficient. IML can also offer users without machine learning or programming expertise an effective way to customise systems. For instance, systems that learn to recognise human gestures or mimic human decision-making can be trained on users' own demonstrations of gestures or decisions. When systems make mistakes, users can often fix them providing corrective demonstrations to the system rather than changing the machine learning algorithm or program code. For instance, a user can provide additional examples of an action that was misclassified by a trained model, along with the correct label for this action, and reasonably hope that the next model trained on the augmented training set will improve its performance on that type of action. IML can thus be understood as a tool for end-user programming, as well as a way to "democratise" machine learning, making the benefits of learning algorithms realisable by a wider range of people.

Based on work by Fiebrink and collaborators (Fiebrink, Cook, and Trueman 2011; Fiebrink et al. 2010; Fiebrink 2011), we argue that using IML for end-user programming and customisation can provide the following advantages:

- IML can be used by people without programming or machine learning expertise.
- IML can facilitate rapid prototyping and iterative refinement activities which are known to be important to design of new systems.
- Providing training examples can be a direct and effective way for a user to communicate the desired behaviour of a system, particularly when designing a system to recognise or respond to human actions. Even expert programmers may have difficulty coding how an embodied activity is to be analysed, but may be able to easily demonstrate an example of the activity.

At the same time, supporting end-user customisation with IML presents several challenges:

- It may be hard for users to reason about how well a system will work in the future. Fiebrink et al. (2011) show that conventional metrics of quality computed from the training data (e.g., cross-validation) are inappropriate for IML systems in which the user employs training data to steer the model behaviour. It may not be clear how to assess whether a model is trustworthy, how to identify likely model mistakes, or how to do these efficiently (i.e., without experimentally feeding the model new data and observing its response).
- It may be difficult for users to select or construct appropriate features. Many real-world applications cannot be built easily from raw data, but rather require some processing of the data in order for learning to be possible. For instance, building new interactions with sensors may require smoothing, segmentation, or analysis of statistics over time windows. Such feature engineering can be difficult even for users with programming and signal processing expertise.
- It may be difficult to collect appropriate training data and understand this data, especially when the user's goal is to build a system that generalises well to new users or environments whose data he/she cannot easily replicate for training or testing.
- It may be difficult for end users to connect machine learning tools to other tools of interest (e.g., existing systems for home automation, music, activity sensors, etc.) Existing sensors, hardware, and software may not use interoperable or open communication protocols.
- Not all designs a user might imagine will be learnable with the available algorithms, features, and data. Users may struggle to understand what is learnable and what is not.

IML can also be useful for developers and designers

Even when the end user interacts with a pre-trained intelligent system, the user experience of the developer or designer deserves consideration. Both the set of supported learning algorithms and user interfaces for employing those algorithms affect the developer experience. Just as with end-user development, IML can allow developers to build new systems by demonstration, resulting in a faster development process. When the goal is to build a system that responds to complex phenomena the developer cannot easily describe in code (e.g., high-dimensional or noisy sensor data, or embodied interactions), IML can also result in systems that are more accurate in their labeling or response to new data.

The speed, directness, and iterative nature of interaction involved in IML makes it a good fit for supporting rapid prototyping for interaction design. IML-based prototyping tools like Wekinator (Fiebrink, Cook, and Trueman 2011) and Exemplar (Hartmann et al. 2007) can yield a smaller gulf of execution (Norman and Draper 1986) than other techniques for building functional interactive systems. Furthermore, reducing the time needed to instantiate and modify prototypes facilitates exploration of the design space.

Improving the interfaces used by developers can make the process of training machine learning systems more efficient and effective (e.g., Patel et al. 2010, Amershi et al. 2015). Yet developers still face many challenges in applying machine learning effectively, and not all these have obvious solutions. In addition to the challenges facing end users, described above, developers and designers must translate their

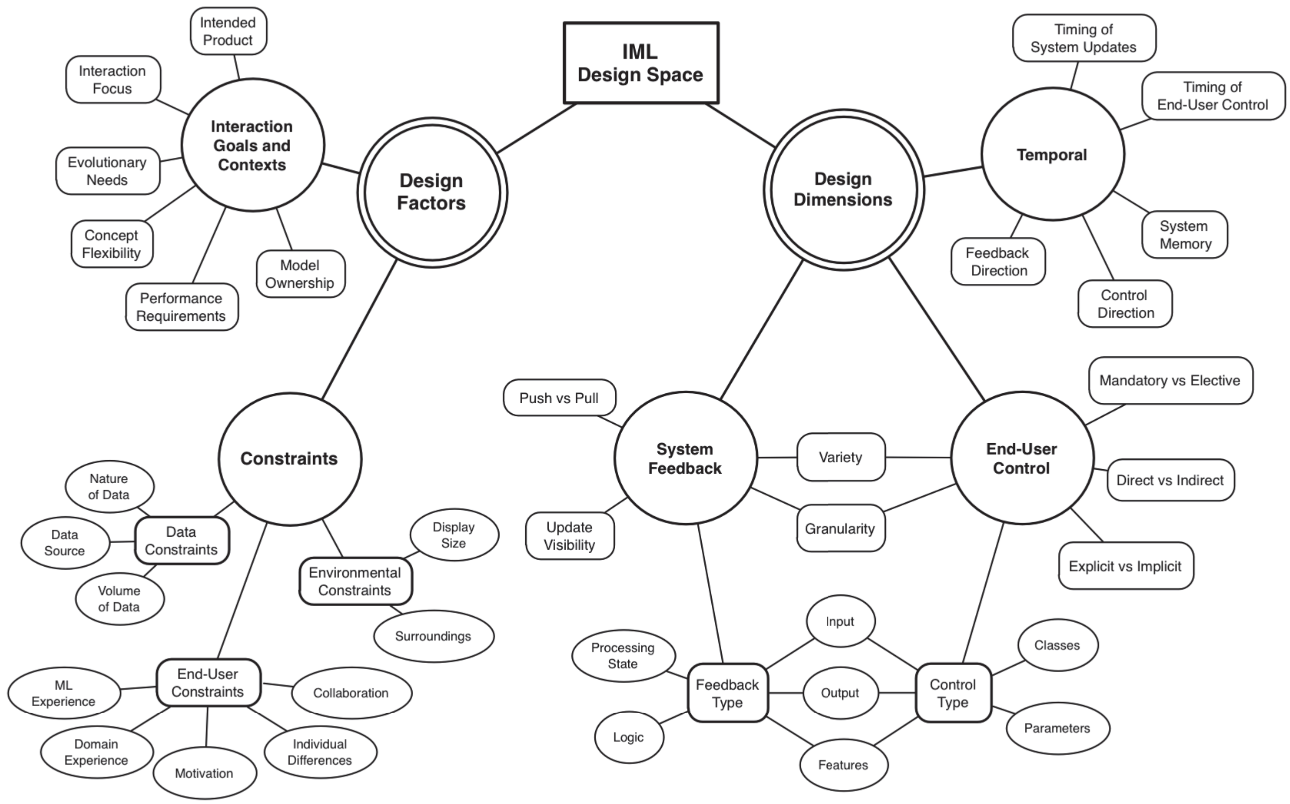

Figure 1: Design space for end-user interaction with machine learning, abstracted from Amershi's (2012) textual description of design factors and dimensions.

design ideas into choices about machine learning configurations. For instance, developers may may have to choose among the many potentially useful learning algorithms. A developer of a gestural control system may choose between classification, regression, and temporal modelling methods such as hidden Markov models. Variants of these algorithms have also been designed specifically to support interactive training or new types of user interaction (e.g., hierarchical HMMs enable gesture segmentation and tracking within segments (Françoise et al. 2014)). Each algorithm may afford a different type of interaction by end users, as well as different trade-offs in accuracy, training time, the number of training examples needed, and so on. Developers may also struggle with feature engineering, despite having more signal processing and programming expertise than end users. They may also encounter substantial challenges in understanding and cleaning training datasets collected from target users.

Another significant challenge is that designers of systems that allow end-user customisation must choose which interactions to expose and how to expose them in a user interface. Amershi (2012) argues that there are a lack of established principles and proven techniques for designing interaction with machine learning, and that knowledge should advance from ad-hoc solutions into generalized understanding about the IML process.

IML can support user-centred design

UCD typically involves iterative cycles of working with users to identify design directions, implementing prototype designs, and getting feedback from users. IML can accelerate this iteration for the design of systems that respond to human gestures or actions, or to other events that can be easily simulated or demonstrated by end users. IML can be used to collect training examples directly from user demonstrations and to instantiate new prototype designs on the spot, which is useful to elicit feedback from users. Or, designers or developers can translate observations of users in UCD activities into training examples for the system being developed. We have previously used both approaches to develop new gestural musical instruments for people with disabilities (Katan, Grierson, and Fiebrink 2015).

UCD should be applied to IML & CML

UCD can and should be applied to understanding and improving usability of interactive and classical machine learning, by both end users and developers. Talbot et al. (2009) observe that "a critical human-computer interaction challenge is to provide adequate tools to allow non-experts to wield [machine learning] techniques effectively. In order to design these tools, we must start at current best practices in applied machine learning and identify tasks that can be supported or augmented by an effective combination of human intuition and input with machine processing" (p. 1283). This challenge and the previously enumerated ones in the sections above cannot be met by algorithmic or technical innovation alone. Rather, meeting these challenges requires understanding of users' practices and goals.

Amershi (2012) highlights the importance of studying users of interactive machine learning applications based on the tighter link that is established between learning systems and their users. She found that users' goals have a significant impact on the success of system design; for instance, design requirements can be influenced by whether end users want to train an accurate or a reusable model; or, whether they are interacting directly with machine learning or using it in the background. These factors can lead to interface designs that differ, for instance, in terms of provisions for alternative mechanisms for end user control and system feedback, and varying degrees of guidance about the quality of the model. These are just a few examples of many factors that impact and constrain how people interact with machine learning systems, which Amershi has distilled into a description of the design space for interactive machine learning (Figure 1).

Prior work contains a number of examples of how UCD processes can improve the user experiences of developers and end-users applying machine learning. For instance, Patel et al. (2010) worked with software developers to create improved development environments for applying conventional machine learning. Fiebrink et al. (2010) used participatory design processes with composers both to design more usable interfaces for applying interactive machine learning to musical instrument design, and to better understand the utility of machine learning in this application domain.

The goals of understanding and improving user experiences with machine learning should also apply when the "interface" employed by the user is a software API. This is the case for many developers who currently use a variety of machine learning libraries to create new software. Some existing research considers the user experience of programming languages, IDEs, and APIs (e.g., (Ko and Riche 2011)), but very little research has considered the challenges that are unique to developers working with machine learning. Understanding the needs of developers building real-time sensor applications and using that understanding to build better APIs are current goals of the RAPID-MIX project.

An innovation studies perspective on IML

Fagerberg (2013) defines innovation studies as "the study of how innovations emerge and diffuse, what factors influence these processes..., and what the social and economic consequences are." (Fagerberg 2013)

One branch of research within innovation studies considers end users' involvement in innovation activities (Von Hippel 1986; Flowers et al. 2010). This work provides a complementary perspective to human-computer interaction research on end-user programming, development, and customisation. Whereas HCI research tends to focus on understanding and supporting end-user engagement, innovation studies research is more concerned with the economic and social contexts that motivate end-user development or customisation, as well as the factors that contribute to the success of end-user innovation.

For instance, von Hippel (1986) notes that in markets with highly heterogeneous consumer needs, certain end users—"lead users"—are highly motivated to develop products for their own needs, and well-positioned to anticipate general demand and identify specific market needs for new products. In many of these cases, both end users and product developers can benefit from the existence of user-innovation toolkits (UITs), which von Hippel (2001) defines as "integrated sets of product-design, prototyping, and design-testing tools intended for use by end users" (p. 163). The fundamental principle behind UITs is to repartition product development tasks by shifting design activities to end users, enabling them to design high-quality custom products, or to customize essential parts of products, and meet their needs in a more accurate and complete way. Integrated circuits (ICs) are the seminal early example a UIT that efficiently and flexibly enabled end-user innovation (von Hippel 2001). Other examples include game level editors such as Final Alert 2 (Jeppesen and Molin 2003) and music synthesis environments such as Propellerhead's Reason (Jeppesen and Frederiksen 2006).

Von Hippel (2001) identified the conditions in which user-innovation toolkits are able to provide the highest value:

i) for types of products and services in which users require high degrees of customisation;

ii) when users have "sticky" information, i.e., need-related, context-of-use or product usage information that is costly to acquire, transfer and apply, and which might grow outdated quickly;

iii) when users "must engage in learning by doing to clarify what they want" (p. 22).

IML offers a set of techniques for addressing these needs in many contexts: As we have discussed earlier in this paper, IML can support efficient customisation and prototyping. General-purpose machine learning algorithms enable learning of diverse concepts, making them suitable for heterogeneous user needs. IML provides users with natural ways to embed specialist or changing knowledge about the intended design into a new system.

Work in innovation studies can therefore inform the creation of IML tools that aim to support end-user innovation, and it can also suggest new uses of IML in creating or improving UITs across a variety of application domains. For instance, Von Hippel (2001) discusses the importance of UITs that enable user learning and problem solving through trial-and-error. This suggests that IML tools for innovation should gracefully accommodate user mistakes and backtracking (e.g., by supporting multiple levels of "undo"-ing, such as rolling back user changes to the training examples). Von Hippel also notes the importance of UITs facilitating the translation of users' designs into production. This suggests it may be beneficial to provide IML users with easy ways to use trained models outside of prototyping environments, for instance embedding them into users' own software or hardware projects.

RAPID-MIX

We are currently participating in a research project involving the application of UCD methodologies to improve tools for applying machine learning and, by extension, the interactive systems built with these tools. *Real-time Adaptive Prototyping for Industrial Design of Multimodal, Interactive and eXpressive technologies* (RAPID-MIX) is an Innovation Action funded by the European Commission under the Horizon 2020 program. The RAPID-MIX consortium aims to accelerate the production of the next generation of multimodal, interactive, and expressive technologies, such as gesturally-controlled musical instruments and games and immersive audiovisual experiences. Specifically, we aim to accelerate this production by creating hardware and software tools for rapid prototyping and product development, and placing these tools in the hands of developers and designers in the form of a UIT (von Hippel 2001).

The members of the RAPID-MIX consortium, which include three European research labs and five small and medium—sized creative technology enterprises, have established research portfolios in the design and evaluation of embodied and wearable human-computer interfaces for creative and music technology. We have also developed and accumulated a significant portfolio of technologies for multimodal and expressive interaction, including music information retrieval and digital signal processing libraries, cloud-based repositories for storing and visualizing audiovisual and multimodal data, and the following tools for applying IML:

- Wekinator (Fiebrink, Cook, and Trueman 2011): a general-purpose, standalone application for applying machine learning. It provides a high-level interface to supervised learning algorithms and their parameters, and it enables users to rapidly create and edit datasets, train and run models in real time.
- XMM (Françoise, Schnell, and Bevilacqua 2013): libraries for using Hierarchical Hidden Markov Models for classification and regression to model gesture and sound parameters, and for creating mappings between gesture and sound in interactive music systems.
- Gesture Variation Follower (Caramiaux et al. 2015): a library for real-time gesture recognition and analysis that employs a template-based method using Sequential Monte Carlo inference.

We are integrating these technologies into a UIT for applying IML to sensors, audio, and other realtime data. This UIT includes developer-facing libraries with a modular and multi-layered architecture for cross-device and cross-platform deployment. The UIT aims to make IML efficient and accessible to developers who are new to machine learning by providing high-level APIs with default settings—such as the choice of learning algorithm and its parameters—that are appropriate for many interaction prototyping applications. For instance, developers prototyping interactions will often provide very small datasets, because they are creating training data by demonstrating interactions at the moment of design rather than curating large datasets in advance. Learning algorithms such as nearest-neighbour may be more appropriate in this context than in more conventional machine learning applications.

Our UIT also includes access to lower-level parameters of the learning algorithms and data processing pipeline, along with numerous examples, to help intermediate and expert users engage in more complex projects. Our UIT also includes mechanisms for exporting trained models and data processing chains from rapid prototyping environments, and running them in production environments (e.g., within other C++ or JavaScript applications).

To facilitate the design of these tools, we have developed a UCD framework that provides a set of guidelines for UCD research actions, which will be used to answer and refine key research questions within the RAPID-MIX project. We are performing multiple UCD actions, including co-design workshops with project partners, hack-a-thons, public workshops, and interviews with professional developers as well as students and hobbyist developers. These actions contribute to define the design space of the UIT based on insights that were obtained from:

i) Assessing and aligning the consortium partners' thinking through ideation, identification of scenarios and design challenges.

ii) Deploying technology probes based on early and background technologies.

iii) Prototyping the integration of component technologies into new products.

iv) Defining an initial set of target users and contexts of use, and evolving gradually to a more scoped definition based on user engagement.

v) Gathering of direct insights from the future users of the toolkit.

vi) Evaluating early products from industry partners, and their integration of the alpha version of the UIT.

Early UCD actions focused on refining and understanding the primary users of the UIT. These include web developers, mobile developers, game developers, creative coders, interaction designers, data analysts/scientists, hackers/makers, researchers and students, teachers, professional audio developers, music creators, and composers.

Deployment of technology probes and testing of early prototypes and UID implementations has led to some better understanding of challenges that exist for these users. For instance, we have found that many participants in our UCD actions struggle to grasp fundamental concepts of machine learning (e.g. classifier, labels, numerical data comes in and labels come out), even when users are professional developers when they are provided with documentation. This has informed our approach to documenting our UIT, which goes beyond code documentation to include interactive code examples as well as video tutorials of machine learning fundamentals.

The effort of intertwining several rounds of rapid prototyping with user engagement uncovered design problems and specific technical challenges for the RAPID-MIX consortium. For instance, some of the IML technologies such as XMM were favoured over classifiers of current market products for classifying dynamic body gestures, but seen as more difficult to use; Wekinator has been perceived as very user-friendly for beginners and designers with few programming skills and was used to build, but for a few participants it was appointed as limited for temporal data and recognition of continuous gestures.

We also observed in a hack-a-thon how different teams of developers were able to successfully integrate our library with their projects for basic machine learning features. Some of the feedback demonstrated that our API design approach could be facilitating understanding with respect to how machine learning can aid design - for example, one participant commented that "(I) could have programmed it in a more traditional manner but I realised how ML makes it easier. You worry about it in some specific ways, here's a couple of things, get to it and it seems to work! What RAPID-MIX brought to me is that, it's just quite usable". Another participant mentioned that it was "great to have a toolkit just ready to use and to plug in". We also uncovered how, for instance, API design decisions regarding data types and containing data structures could have an impact on the conceptual model of professional developers and mislead which machine learning tasks to use in slightly more sophisticated projects.

Reports on our UCD methodologies and our design guidelines can be found on our website at: http://rapidmix.goldsmithsdigital.com/downloads/

Conclusion

IML can enable end users and developers to customize interactive systems and can support UCD. Furthermore, UCD can and should be applied to the process of creating machine learning applications of all kinds. These positions are informed by, and inform the RAPID-MIX project, which is ongoing. By engaging with members of the user experience design, service design, HCI, HRI and AI communities at this symposium, we hope to connect with complementary perspectives and potential users of our toolkit and use that feedback in our next cycle of design and prototyping.

Acknowledgements

The work reported here was supported in part by RAPID-MIX, an EU Horizon 2020 Innovation Action: H2020-ICT-2014-1 Project ID: 644862.

References

Amershi, S.; Lee, B.; Kapoor, A.; Mahajan, R.; and Christian, B. 2011. Cuet: human-guided fast and accurate network alarm triage. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 157–166. ACM.

Amershi, S.; Cakmak, M.; Knox, W. B.; and Kulesza, T. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* 35(4):105–120.

Amershi, S.; Chickering, M.; Drucker, S. M.; Lee, B.; Simard, P.; and Suh, J. 2015. ModelTracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 337–346.

Amershi, S. 2012. *Designing for Effective End-User Interaction with Machine Learning*. Ph.D. Dissertation, University of Washington.

Caramiaux, B.; Montecchio, N.; Tanaka, A.; and Bevilacqua, F. 2015. Adaptive gesture recognition with variation estimation for interactive systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 4(4):18.

Fagerberg, J. 2013. Innovation a New Guide.

Fails, J. A., and Olsen Jr, D. R. 2003. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, 39–45.

Fiebrink, R.; Trueman, D.; Britt, C.; Nagai, M.; Kaczmarek, K.; Early, M.; Daniel, M.; Hege, A.; and Cook, P. 2010. Toward understanding human-computer interaction in composing the instrument. In *Proc. of the International Computer Music Conference*.

Fiebrink, R.; Cook, P. R.; and Trueman, D. 2011. Human model evaluation in interactive supervised learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 147–156.

Fiebrink, R. 2011. *Real-time human interaction with supervised learning algorithms for music composition and performance*. Ph.D. Dissertation, Princeton University.

Flowers, S.; von Hippel, E.; de Jong, J.; and Sinozic, T. 2010. Measuring user innovation in the uk: The importance of product creation by users.

Françoise, J.; Schnell, N.; Borghesi, R.; and Bevilacqua, F. 2014. Probabilistic models for designing motion and sound relationships. In *Proceedings of the 2014 International Conference on New Interfaces for Musical Expression*, 287–292.

Françoise, J.; Schnell, N.; and Bevilacqua, F. 2013. A multimodal probabilistic model for gesture–based control of sound synthesis. In *Proceedings of the 21st ACM international conference on Multimedia*, 705–708. ACM.

Hartmann, B.; Abdulla, L.; Mittal, M.; and Klemmer, S. R. 2007. Authoring sensor-based interactions by demonstration with direct manipulation and pattern recognition. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 145–154.

Jeppesen, L. B., and Frederiksen, L. 2006. Why do users contribute to firm-hosted user communities? the case of computer-controlled music instruments. *Organization science* 17(1):45–63.

Jeppesen, L. B., and Molin, M. J. 2003. Consumers as co-developers: Learning and innovation outside the firm. *Technology Analysis & Strategic Management* 15(3):363–383.

Kapoor, A.; Lee, B.; Tan, D.; and Horvitz, E. 2010. Interactive optimization for steering machine classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1343–1352.

Katan, S.; Grierson, M.; and Fiebrink, R. 2015. Using interactive machine learning to support interface development through workshops with disabled people. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 251–254.

Ko, A. J., and Riche, Y. 2011. The role of conceptual knowledge in API usability. In *VL/HCC*, 173–176.

Lieberman, H.; Paternò, F.; Klann, M.; and Wulf, V. 2006. End-user development: An emerging paradigm. In *End user development*. Springer. 1–8.

Norman, D. A., and Draper, S. W. 1986. User centered design. *New Perspectives on Human-Computer Interaction*.

Patel, K.; Bancroft, N.; Drucker, S. M.; Fogarty, J.; Ko, A. J.; and Landay, J. 2010. Gestalt: Integrated support for implementation and analysis in machine learning. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, 37–46.

Talbot, J.; Lee, B.; Kapoor, A.; and Tan, D. S. 2009. EnsembleMatrix: interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1283–1292.

Von Hippel, E. 1986. Lead users: A source of novel product concepts. *Management science* 32(7):791–805.

von Hippel, E. 2001. User toolkits for innovation. *Journal of product innovation management* 18(4):247–257.

von Hippel, E. 2005. Democratizing innovation: The evolving phenomenon of user innovation. *Journal für Betriebswirtschaft* 55(1):63–78.