Constructivist Design for Interactive Machine Learning

Advait Sarkar
Computer Laboratory
University of Cambridge
Cambridge CB3 0FD
United Kingdom
advait.sarkar@cl.cam.ac.uk

CHI'16 Extended Abstracts, May 07-12, 2016, San Jose, CA, USA
ACM 978-1-4503-4082-3/16/05.
http://dx.doi.org/10.1145/2851581.2892547

Abstract
Interactive machine learning systems allow end-users, often non-experts, to build and apply statistical models for their own uses. Constructivism is the view that learning occurs when ideas and experiences interact. I argue that the objectives of interactive machine learning can be interpreted as constructivist. By so characterising them, I show how constructivist learning environments pose critical questions for the design of interactive machine learning systems.

Author Keywords
Constructivist design; Interactive machine learning; Learning environments; End-user programming; Analytics

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

What is interactive machine learning?
As machine learning makes rapid advances, researchers are increasingly interested in enabling people to build and apply machine learning models for their own use in a variety of scenarios. These end-users are typically not experts in statistics or machine learning, so careful interaction design is applied, in order to reduce the expertise barriers imposed by the hard concepts of statistical modelling and model programming. Thus, the model construction process

At a glance:
types of IML systems

Model-building systems help the user build a reusable model with which to classify or predict new values. For example, Crayons [12].

Analytic systems help the user explore, understand, and analyse a dataset. For example, Behrisch et al.'s feedback-driven exploratory classifier [5].

Hybrid systems incorporate both the aims of model-building as well as analytic systems, and sometimes also aim to illustrate statistical concepts. For example, BrainCel [28].

typically involves a training loop wherein the user repeatedly chooses (or is given) example data, and then provides a judgement or label with which to train the system. As it is an interactive process, these systems are collectively known as *interactive machine learning* (IML) systems.

IML systems can largely be categorised into 3 types, based on the tasks they facilitate: model-building, analytics, and hybrid (facilitating both model-building and analytics). The *model-building* activity is not principally concerned with the *structure* of the model, but rather its ability to robustly predict real-world outcomes, and so it is usually acceptable for the model to have undecipherably large numbers of parameters, and convoluted structure. The outcome here is a reusable model. In contrast, the *analytics* activity is concerned with the interpretation of data through the model, typically requiring the model to have small numbers of readily-interpretable parameters. The outcome here is analytic insight. The difference between model-building and analytics can be loosely understood as Breiman's 'Two Cultures' of statistical modelling [8] manifested in task form. Hybrid systems incorporate elements of both.

Crayons [12] is perhaps the canonical example of interactive machine learning in the model-building tradition. In Crayons, the user builds a classifier to segment an image, for example, to detect a hand against a background. The user annotates a training image by drawing positive (hand) and negative (not-hand) labels by colouring in the relevant regions with a brush tool as one would in a graphics application. As the user draws, the system trains a model (in this case, fast decision trees) on the training data and displays the model prediction on the image itself, through a translucent overlay. This makes clear which parts of the image are currently 'thought' to be a hand by the model, enabling the user to focus further annotation on misclassified areas. This loop of continuous human intervention and system feedback, through which a machine-learned model is improved, is the essence of interactive machine learning.

CueT [2], another model-building system, helps users quickly triage alarms regarding the health of devices on a network. As network alarms stream in, the system suggests a number of options for triaging the alarm, based on the operator's past history of attending to alarms. It learns continuously, allowing it to quickly start producing useful suggestions for previously unencountered alarm situations.

An example of an analytics application is Behrisch et al.'s interactive system for building decision tree classifiers on large multidimensional data [5]. In contrast to Crayons and CueT, the objective here is not to build a reusable model, but to learn something about the structure of the data being modelled. Multiple visualisations are used to help the end-user develop a sense of the quality of the model. For instance, a histogram shows the number of uncertain decisions – when a large proportion of the dataset is classified with low certainty, that suggests that the model is currently of poor quality and more (or better) training data is required.

The *BrainCel* system [27, 28] allows users to build and apply machine learning systems on data within spreadsheets. Users select rows of 'correct' data to train a model which can then predict values for empty cells. Using multiple coordinated views of the model, BrainCel helps end-users to critically assess the quality of their model, including ensuring adequate class representation, and avoiding reliance on predicted values which have low confidence. BrainCel is a hybrid, developed to encompass the aims of systems like Crayons as well as Behrisch et al.'s. That is, it is both an analytics tool and a model-building tool. A final, additional aim of BrainCel has been to introduce statistical concepts to non-expert end-users.

What is constructivism?

Constructivism is a theory describing the learning process; the manner in which human knowledge is generated. It posits the view that human knowledge is constructed as a result of the interaction between a person's mental models and their experiential perceptions. This stands in opposition to the naïve psychology made explicit by Heider [3] that knowledge is information, and consequently, learning is analogous to information delivery; in this (instructionist) view, one can optimise learning purely by training teachers to transfer information more effectively. Constructivism has had many influences but is largely attributed to Piaget [33]. Although evidence of the benefits of constructivism as a pedagogical tool has been mixed [20], the theory has been highly influential on learning theory and educational reform.

Computer science education has been the source of a few notable applications of constructivist theory. For example, Papert's Logo programming system [1]. Logo's emphasis on direct visual representation and feedback through turtle graphics allowed novice programmers to directly *experience* the effects of their code, increasing the surface area for interaction between their ideas and experiences, facilitating the construction of new or better mental models. The idea has been refined and combined with powerful notions such as blocks programming and multiple representations, to create sophisticated constructivist environments such as Alice [10], Scratch [25], and DrawBridge [30].

Implicit and explicit learning outcomes in IML

As previously noted, interactive machine learning systems have multiple objectives. Some, such as Crayons, allow you to build a reusable model, whereas others, such as Behrisch et al.'s, allow you to analyse a dataset through the lens of a statistical model. Still others, such as BrainCel, aim to expose the end-user to statistical concepts.

If examined carefully, each of these can be restated as *learning outcomes*, as follows:

1. Model-building: learning about the model instance, its strengths, weaknesses, coverage of training data, fitted parameters, etc.
2. Analysis: learning about the structure of the data, its statistical properties and features.
3. Exposition to statistical concepts: learning about a particular algorithm, or general concepts about training and testing such as class representation, noisy data, outliers, etc.

Systems such as Logo aim to maximise constructivist knowledge generation by maximising the opportunities for interaction between experience and ideas. Happily, the feedback loop between system and user which drives interactive machine learning is also a rich source of experience generation. As the user adds training examples or manipulates other parameters of the model, the system attempts to illustrate its current understanding through externalisations such as the translucent overlays in Crayons, an experience which causes the user to update their mental model of the system's intelligence – a form of metacognition [26].

Thus, it may be argued that IML systems are constructivist learning environments just like Logo and Scratch, but here the 'programs' being generated are not graphics or animations, but models, and the programming language is not a blocks language, but the stream of interactions (annotation, labelling, parameter adjustment). However, unlike Logo and Scratch, 'learning' in the sense of acquiring concepts and gaining real-world skills is not an explicit outcome in IML systems. Rather, in IML systems, production of the desired result, whether reusable model or analytic insight, is incumbent on a series of implicit intermediate learning outcomes.

Critical constructivist questions for IML

Many interesting challenges and opportunities arise from subscribing to the view that IML systems are really constructivist programming systems with the aim of generating certain types of knowledge through an experiential interaction loop. Many virtual learning environments grounded in constructivist design, so-called constructivist learning environments (CLEs), have been developed for domains outside programming. For instance, the *Lab Design Project* was a hypermedia system designed for researchers to practice sociological research skills and to learn about how lab design shapes scientific practice [17]. The *Jasper* series [24] developed at Vanderbilt immersed students in vivid stories to encourage situated response to mathematics problems. Consequently, much research has focused on the design of such CLEs. In this section I have engaged in the hermeneutic exercise of drawing upon a few core texts [18, 9, 32, 14, 16, 23, 19] in that area. In the following paragraphs, I will highlight the established design principles of CLEs which ask meaningful critical questions for the design of IML systems.

Task ownership Users learn by working towards a problem which they see to be relevant and reflective of real-world situations. Such a stimulus for authentic activity causes users to be goal-oriented and intrinsically motivated to complete the task. It is typical for IML systems to satisfy this criterion, as both model-building and analytics applications facilitate an end-user task – it is an assumption of these systems that users will only engage with the system precisely because they need to solve a real-world problem of which they take ownership.

Ill-defined problem Users are able to better engage with problems which are ill-defined. This allows aspects of the problem to be emergent, requiring users to make defensible judgements. Both model-building and analytics are ill-defined. With typical software engineering problems, the definition of ‘bug’ is typically uncontroversial, with the programmer able to conclusively decide whether some behaviour of their program is desirable or undesirable. However, in model-building and analytics, the user does not really know what the ‘right’ answer is. Instead, these activities are dominated by ill-defined questions, for instance: does the model accurately capture the domain? Is it overfitting? Are these variables interrelated? Is this analysis leading to sound conclusions?

IML’s techno-pragmatic roots have led to an odd juxtaposition between the objectivist requirements of machine learning algorithms, and the open-ended nature of the tasks being facilitated. To build a robust machine learning model requires high-quality training data with clearly discriminable classes, free of training label noise. The user’s task is to recognise and label objects, organise them coherently, and integrate them with existing knowledge – a decidedly instructionist approach. By contrast, in the constructivist view, objects do not have absolute meaning; meaning is constructed by the individual as a result of experiences and situated beliefs. This type of activity requires a rich context where meaning can be negotiated and understanding can emerge and evolve.

When IML systems hit the limits of the objectivist approach, new interaction design techniques coupled with powerful inference algorithms have been shown to provide a unique middle ground. Some examples of this include *structured labelling* [21] and *setwise comparison* [29], where training labels are assumed to be ill-defined, and the system assists the user in gradually forming stable notions of labelled concepts, fit for use in machine learning systems.

Perturbation The concept of perturbation (or disequilibration, in Piagetian terms) is the engine driving the learning process. It refers to a stimulus which does not conform, or gently subverts, the expectations and mental model of the user, forcing them to construct new knowledge in order to 'accomodate' this experience. The introduction of perturbations, and encouraging the strategic exploration of errors, is already a central issue in IML systems, since building effective and interpretable models revolves around the activity of addressing errors made by the machine-learned model [22]. Unlike IML, CLEs acknowledge that errors have deeply embedded negative connotations in our sociocultural environment. As the intended learning outcomes of IML become more explicit, designers need to be sensitive to the impact of errors on learners' motivations, and the potential for the misattribution of poor instructional outcomes.

I have begun by introducing three matters (*task ownership*, *ill-defined problem*, and *perturbation*) with which IML systems already engage to some extent, but for which I have elaborated a new theoretical grounding based in constructivist design. Next, I will discuss how four further issues core to CLEs shed new light on IML systems.

Reflexivity A critical self-awareness of one's learning, beliefs and knowledge is central to constructivist environments. Reflective users take control over and responsibility for their thoughts, and create a defensible catalogue of provenance for their knowledge. IML systems currently do not promote critical reflection, but a promising solution is to capture the user's interaction history in detail, and facilitate simple querying and browsing. This is not technically straightforward, but exemplary design solutions exist in the domains of sketching [36] and source code change history [35, 34] which demonstrate how such affordances support reasoning about knowledge provenance as well as direct manipulation of the knowledge construction history space.

Collaboration Learning takes place in a social context. The construction of meaning, like so many other activities, seldom occurs individually. The ability of a user to perform is predicated on group contexts, unlike the (typically) artificially individualistic settings of school and classrooms. To this end, CLEs often incorporate collaborative activities intended to promote dialogue and encourage the social exposure of ideas. IML systems are typically not designed with collaboration in mind, but may import lessons from collaborative analytics [15] in order to do so.

Task in context Knowledge construction is context-dependent. Within a particular setting, it is historically developed, evolved over time within a culture. Moreover, beliefs and opinions are constantly adjusted by socially-mediated expectations. This process tends towards increasing common ground, resulting in an increased robustness of mental models. Professional customs, skills, workflows, and institutional expectations all filter and sculpt learning. The design *process* for IML systems is sensitive towards these issues, but what this might mean for their *design* is a critical question.

Tool mediation The process of creating knowledge is mediated by tools and symbols. Just as carpentry is not merely teleology for the hammer, but is also actively shaped by the invention thereof, so do many aspects of modern technology influence the practice from which they emerge. For example, the invention of email hasn't merely made us more efficient communicators – the nature of that technology has radically changed our paradigms for communication. The invention of tools may be a cultural necessity, but the tools, in turn, transform the culture.

At a glance: critical constructivist issues for IML

Task ownership: a relevant, owned problem is intrinsic motivation for learning.

Ill-defined problem: requires users to tackle emergent issues and make defensible judgements.

Perturbation: nonconforming experiences necessitate mental 'accomodation'.

Reflexivity: self-critical learners understand their thoughts and knowledge better.

Collaboration: the construction of meaning is a social activity.

Task in context: historical and socio-cultural influences shape learning.

Tool mediation: systems are responsible for changing the culture which creates them.

IML systems, especially those with an emphasis on analytical outcomes, need to be aware of their role as cultural mediators. Boyd and Crawford [7] and Blackwell [6] have highlighted as a concern the fact that knowledge is shaped by the constraints, assumptions, and contexts of statistical algorithms and the data on which they operate. For instance, are Twitter posts representative of global sentiment, simply because they constitute a large sample? IML systems embody epistemological and ontological assertions, which are currently implicit. This is made further problematic because statistical inference is the subject of much dispute. Clearly we are not ready to settle into ideologies and epistemologies when there is warfare on several fronts, e.g., frequentist versus Bayesian approaches, which differ on such fundamental axioms as the interpretation of 'truth' [4, 11]. Since these analytical systems constitute an indispensable *umwelt* through which we collectively experience data, it is well worth these assumptions being made explicit.

Implications for design practice in IML

This brief analysis cannot profess to directly support design practice, not least because IML, still in its nascency, has yet to form a clear identity as a design practice. As Stolterman [31] notes, interaction design research often fails due to inaccurate characterisations of design as a practiced activity. However, over the course of this exercise, two themes with apparent implications for IML have emerged from the documented design practice for CLEs. The first is that design must be grounded in a defensible theoretical framework, and validated iteratively through successive implementations. Secondly, design must be informed through situated techniques such as ethnography and activity-theoretic analysis. Many current IML systems are indubitably products of similar underlying processes; however, the thick description which typically evidences such a process is conspicuously lacking from our current literature.

Conclusions and future work

The constructivist view of IML expands our design considerations and shows how we might frame concerns such as reflexivity, task in context, and tool mediation. The discussion has raised several practical and theoretical issues. For instance, what are the appropriate design and interaction metaphors for knowledge provenance? How can model-building be collaborative – is it more like collaborative analytics, or collaborative problem solving [13]? In what ways can IML systems promote reflexivity? How could IML systems explicate epistemological assertions? Is there a fundamental difference between learning and analytics, and if so, how is that meaningfully articulated in design? Currently, learning is an implicit side-effect of IML systems – what would a pure, explicit learning environment look like?

In this paper, I have argued that the interaction loop of interactive machine learning systems facilitates constructivist learning, as it maximises the interaction between the end-user's experience of the model, and their ideas regarding the model status. I have drawn parallels between interactive machine learning systems and constructivist learning environments. While interactive machine learning systems have so far had a certain set of pragmatic design influences, this constructivist interpretation opens up new avenues and implications for design.

Acknowledgements

I thank the reviewers, whose feedback has greatly improved this paper, and Alan Blackwell for discussions. I am supported by an EPSRC and BT iCASE award, and a Cambridge Computer Laboratory Robert Sansom Scholarship.

References

[1] Harold Abelson and Andrea DiSessa. 1986. *Turtle geometry: The computer as a medium for exploring mathematics*. MIT press.

[2] Saleema Amershi, Bongshin Lee, Ashish Kapoor, Ratul Mahajan, and Blaine Christian. 2011. CueT: Human-Guided Fast and Accurate Network Alarm Triage. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. ACM Press, New York, New York, USA, 157. DOI: http://dx.doi.org/10.1145/1978942.1978966

[3] Alfred Lee Baldwin. 1967. Theories of child development. (1967).

[4] M Jésus Bayarri and James O Berger. 2004. The interplay of Bayesian and frequentist analysis. *Statist. Sci.* (2004), 58–80.

[5] Michael Behrisch, Fatih Korkmaz, Lin Shao, and Tobias Schreck. 2014. Feedback-driven interactive exploration of large multidimensional data supported by visual classifier. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*. 43–52. DOI: http://dx.doi.org/10.1109/VAST.2014.7042480

[6] Alan Blackwell. 2015. Interacting with an Inferred World: The Challenge of Machine Learning for Humane Computer Interaction. *Aarhus Series on Human Centered Computing* 1, 1 (2015), 12.

[7] Danah Boyd and Kate Crawford. 2012. Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon. *Information, communication & society* 15, 5 (2012), 662–679.

[8] Leo Breiman and others. 2001. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statist. Sci.* 16, 3 (2001), 199–231.

[9] D Cunningham and T Duffy. 1996. Constructivism: Implications for the design and delivery of instruction. *Handbook of research for educational communications and technology* (1996), 170–198.

[10] Wanda P Dann, Stephen Cooper, and Randy Pausch. 2011. *Learning to Program with Alice (w/CD ROM).* Prentice Hall Press.

[11] Bradley Efron. 2005. Bayesians, frequentists, and scientists. *J. Amer. Statist. Assoc.* 100, 469 (2005), 1–5.

[12] Jerry Alan Fails and Dan R. Olsen. 2003. Interactive machine learning. *Proceedings of the 8th international conference on Intelligent user interfaces - IUI '03* (2003), 39. DOI: http://dx.doi.org/10.1145/604050.604056

[13] Sandra B Fan, Tyler Robison, and Steven L Tanimoto. 2012. CoSolve: A system for engaging users in computer-supported collaborative problem solving. In *Visual Languages and Human-Centric Computing (VL/HCC), 2012 IEEE Symposium on*. IEEE, 205–212.

[14] Michael J Hannafin, Kathleen M Hannafin, Susan M Land, and Kevin Oliver. 1997. Grounded practice and the design of constructivist learning environments. *Educational Technology Research and Development* 45, 3 (1997), 101–117.

[15] Jeffrey Heer and Maneesh Agrawala. 2008. Design considerations for collaborative visual analytics. *Information visualization* 7, 1 (2008), 49–62.

[16] Peter C Honebein. 1996. Seven goals for the design of constructivist learning environments. *Constructivist learning environments: Case studies in instructional design* (1996), 11–24.

[17] Peter C Honebein, Thomas M Duffy, and Barry J Fishman. 1993. Constructivism and the design of learning environments: Context and authentic activities for learning. In *Designing environments for constructive learning*. Springer, 87–108.

[18] David H Jonassen. 1999. Designing constructivist learning environments. *Instructional design theories and models: A new paradigm of instructional theory* 2 (1999), 215–239.

[19] David H Jonassen and Lucia Rohrer-Murphy. 1999. Activity theory as a framework for designing constructivist learning environments. *Educational Technology Research and Development* 47, 1 (1999), 61–79.

[20] Paul A Kirschner, John Sweller, and Richard E Clark. 2006. Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational psychologist* 41, 2 (2006), 75–86.

[21] Todd Kulesza, Saleema Amershi, Rich Caruana, Danyel Fisher, and Denis Charles. 2014. Structured labeling for facilitating concept evolution in machine learning. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*. ACM Press, New York, New York, USA, 3075–3084. DOI: http://dx.doi.org/10.1145/2556288.2557238

[22] Todd Kulesza, Margaret Burnett, Weng-keen Wong, and Simone Stumpf. 2015. Principles of Explanatory Debugging to Personalize Interactive Machine Learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces - IUI '15*. 126–137. DOI: http://dx.doi.org/10.1145/2678025.2701399

[23] David Lebow. 1993. Constructivist values for instructional systems design: Five principles toward a new mindset. *Educational technology research and development* 41, 3 (1993), 4–16.

[24] JW Pellegrino, D Hickey, A Heath, K Rewey, NJ Vye, and CGTV Vanderbilt. 1992. Assessing the outcomes of an innovative instructional program: The 1990-1991 implementation of the "Adventures of Jasper Woodbury.". *Nashville, TN: Leaming Technology Center, Vanderbilt University* (1992).

[25] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and others. 2009. Scratch: programming for all. *Commun. ACM* 52, 11 (2009), 60–67.

[26] Advait Sarkar. 2015. Confidence, command, complexity: metamodels for structured interaction with machine intelligence. In *Proceedings of the 26th Annual Conference of the Psychology of Programming Interest Group (PPIG 2015)*. 23–36.

[27] Advait Sarkar, Alan F Blackwell, Mateja Jamnik, and Martin Spott. 2014. Teach and try: A simple interaction technique for exploratory data modelling by end users. In *Visual Languages and Human-Centric Computing (VL/HCC), 2014 IEEE Symposium on*. IEEE, 53–56. DOI: http://dx.doi.org/10.1109/VLHCC.2014.6883022

[28] Advait Sarkar, Mateja Jamnik, Alan F. Blackwell, and Martin Spott. 2015. Interactive visual machine learning in spreadsheets. In *Visual Languages and Human-Centric Computing (VL/HCC), 2015 IEEE Symposium on*. IEEE, 159–163. DOI: http://dx.doi.org/10.1109/VLHCC.2015.7357211

[29] Advait Sarkar, Cecily Morrison, Jonas F. Dorn, Rishi Bedi, Saskia Steinheimer, Jacques Boisvert, Jessica Burggraaff, Marcus D'Souza, Peter Kontschieder, Samuel Rota Bulò, Lorcan Walsh, Christian P. Kamm, Yordan Zaykov, Abigail Sellen, and Siân E. Lindley. 2016. Setwise Comparison: Consistent, Scalable, Continuum Labels for Computer Vision. In *Proceedings of the 34th annual ACM conference on Human factors in computing systems - CHI '16*. ACM Press. DOI: http://dx.doi.org/10.1145/2858036.2858199

[30] Alistair Stead and Alan F Blackwell. Learning Syntax as Notational Expertise when using DrawBridge. In *Psychology of Programming Interest Group Annual Conference 2014*. 41.

[31] Erik Stolterman. 2008. The nature of design practice and implications for interaction design research. *International Journal of Design* 2, 1 (2008).

[32] Peter C Taylor, Barry J Fraser, and Darrell L Fisher. 1997. Monitoring constructivist classroom learning environments. *International journal of educational research* 27, 4 (1997), 293–302.

[33] Barry J Wadsworth. 1996. *Piaget's theory of cognitive and affective development: Foundations of constructivism .* Longman Publishing.

[34] YoungSeok Yoon and Brad A Myers. 2015. Semantic zooming of code change history. In *Visual Languages and Human-Centric Computing (VL/HCC), 2015 IEEE Symposium on*. IEEE, 95–99.

[35] YoungSeok Yoon, Brad A Myers, and Sebon Koo. 2013. Visualization of fine-grained code change history. In *Visual Languages and Human-Centric Computing (VL/HCC), 2013 IEEE Symposium on*. IEEE, 119–126.

[36] Zhenpeng Zhao, William Benjamin, Niklas Elmqvist, and K. Ramani. 2015. Sketcholution: Interaction Histories for Sketching. *International Journal of Human-Computer Studies* (2015).