A Review of User Interface Design for Interactive Machine Learning

JOHN J. DUDLEY and PER OLA KRISTENSSON, University of Cambridge, United Kingdom

8

Interactive Machine Learning (IML) seeks to complement human perception and intelligence by tightly integrating these strengths with the computational power and speed of computers. The interactive process is designed to involve input from the user but does not require the background knowledge or experience that might be necessary to work with more traditional machine learning techniques. Under the IML process, non-experts can apply their domain knowledge and insight over otherwise unwieldy datasets to find patterns of interest or develop complex data-driven applications. This process is co-adaptive in nature and relies on careful management of the interaction between human and machine. User interface design is fundamental to the success of this approach, yet there is a lack of consolidated principles on how such an interface should be implemented. This article presents a detailed review and characterisation of Interactive Machine Learning from an interactive systems perspective. We propose and describe a structural and behavioural model of a generalised IML system and identify solution principles for building effective interfaces for IML. Where possible, these emergent solution principles are contextualised by reference to the broader human-computer interaction literature. Finally, we identify strands of user interface research key to unlocking more efficient and productive non-expert interactive machine learning applications.

CCS Concepts: • **Human-centered computing** → *HCI theory, concepts and models*;

Additional Key Words and Phrases: Interactive machine learning, interface design

ACM Reference format:
John J. Dudley and Per Ola Kristensson. 2018. A Review of User Interface Design for Interactive Machine Learning. *ACM Trans. Interact. Intell. Syst.* 8, 2, Article 8 (June 2018), 37 pages.
https://doi.org/10.1145/3185517

1 INTRODUCTION

Machine Learning (ML) supports the construction of mathematical models that can describe and exhibit complex behaviour without the need for explicit programming. These techniques have demonstrated advanced functionality in a wide variety of tasks such as recognising speech, classifying images, and playing games. This functionality is built through a process of training that is not dissimilar from the human process of learning. In general terms, the algorithm learns to recognise or represent a concept through repeated exposure to samples of that concept. A mathematical

Per Ola Kristensson was supported in part by a Google Faculty research award and EPSRC grants EP/N010558/1 and EP/N014278/1. John Dudley was supported by the Trimble Fund.
Authors' address: J. J. Dudley and P. O. Kristensson, University of Cambridge, Department of Engineering, Trumpington St, Cambridge CB2 1PZ, United Kingdom; emails: {jjd50, pok21}@cam.ac.uk.

https://doi.org/10.1145/3185517

model that accurately represents a concept for some functional purpose is the goal product of a machine learning algorithm.

Despite the success of these techniques, their use as a development tool remains largely confined to expert practitioners. However, the ability of these approaches to deliver functionality without explicit programming makes them attractive to a broad group of potential users. The recent explosion in broad interest in machine learning further exacerbates the shortfall in intuitive systems that allow non-experts to apply these techniques. Interactive Machine Learning (IML) has attracted interest among human-computer interaction (HCI) researchers due to the unique aspects of establishing effective human interactions under this paradigm.

IML (defined in detail in Section 2) makes machine learning techniques more accessible by careful framing of the training process as an HCI task. The IML process involves the user in the training process by, at the simplest level, using human input in the example selection, creation, and labelling process. A machine learning practitioner may be required to deploy the underlying algorithm for use in the IML system but is not essential to the training process. In an ideal implementation, the machine learning-naïve end user can construct their own learned concepts by creating or collecting training data according to their need. In practice, constructing such a system is a major challenge for both the machine learning practitioner and for the user interface designer. This article focuses primarily on the interface design challenge but does highlight points of intersection with the technical challenges for the machine learning practitioner.

Early efforts by Ware et al. (2001) and Fails and Olsen (2003) demonstrated that typical machine-learning tasks could be framed to take advantage of human input, and over the past decade and a half the IML process has seen increasing attention within the HCI community. The user providing input to the IML system need not possess any deep understanding of the models with which they are interacting. The value of the approach is perhaps most evident when one considers the potential to allow a domain expert to train a model. Researchers have demonstrated IML systems that leverage domain expertise in a range of specific applications including animal behaviour annotation (Kabra et al. 2013), insurance claims auditing (Ghani and Kumar 2011), and conducting systematic reviews of published medical evidence (Wallace et al. 2012). In these applications, the effort required by the user is significantly reduced as the model improves its understanding of the concept being trained. Wallace et al. (2012) achieved a reduction in user workload of approximately 40% by exploiting the predictions of the trained model to mark particular research papers as irrelevant. In this way, IML seeks to apply the complementary strengths of humans and computers.

The IML workflow is inherently co-adaptive in that the user and the target model directly influence each other's behaviour. It is important to recognise that it is in fact the interactions, as an emergent property of the interface, that must actually accommodate the changing behaviour of the user in response to the model. Appropriate design of the interface is *critical* to the success of such systems and this presents a unique user interface design challenge. The approach relies on the transformation of the data inspection and correction task into an intuitive and productive dialog between human and machine. Achieving this is non-trivial.

Designing the interface for an Interactive Machine Learning application exposes four key challenges. First, users can be imprecise and inconsistent. A user may not strictly adhere to a concept during the training process or may introduce their own errors and biases into the learning process. Poor training leads to poor models but unless users perceive their own deficiencies, this failure is attributed to the system.

Second, there is a degree of uncertainty when relating user input and user intent. Users may assign training data based on features that they perceive in an example that the system does not or cannot model. In addition, the fact that a user did not assign an example to a concept does not necessarily imply that it is a counterexample.

Third, interacting with a model is not like interacting with a conventional information structure through a user interface. The machine learning model evolves in response to user input but not necessarily in a way that is perceived as intuitive or predictable by the user.

Fourth, training is open ended. For example, if a user is designing a part in a Computer-Aided Design (CAD) system, then the state of completion is more or less binary: the task is complete when the part is fully specified. By contrast, training a model to be 100% accurate may be both undesirable and impossible. Furthermore, IML has seen extensive use in exploratory and creative applications where the actual model is perhaps of secondary importance to the process itself.

These four key challenges in aggregate make human-centred machine learning, and the IML process specifically, a design challenge that is far more complicated than simply integrating the relevant components.

1.1 Guidance for the Designer

This article consolidates research findings related to interface design for IML with the objective of providing guidance to designers. It seeks to bring a binocular perspective to the challenge of building an effective Interactive Machine Learning system: a unified consideration of the application of intelligent algorithms and the design of the user interface. We present both a structural and behavioural breakdown of an IML system based on our synthesis of the literature. The structural breakdown divides the system into its constituent components (the user, the interface, the data, and the model) with the objective of helping designers think about how they might architect such a system. The IML interface is further decomposed into four distinct interface elements that serve different functions within the broader procedure. Awareness of these distinct interface elements can help designers think about how they present information to users and what interactions they promote.

The behavioural breakdown separates the process of interactively building a model into subtasks. It is important to note, however, that these tasks may overlap and/or be performed iteratively. The structural and behavioural breakdowns are not intended to serve as blueprints but rather to support rich discussion of design considerations with clearly defined terminology. This formalisation of the design space will contribute to a better foundation for exchange of concepts in this emerging field.

A further contribution of this article is the identification of emergent solution principles that can help guide IML interface designers more generally. The six solution principles identified and described in detail in Section 6 are as follows:

(1) Make task goals and constraints explicit
(2) Support user understanding of model uncertainty and confidence
(3) Capture intent rather than input
(4) Provide effective data representations
(5) Exploit interactivity and promote rich interactions
(6) Engage the user

1.2 Outline

The remainder of the article is organised as follows. Section 2 provides a detailed definition of Interactive Machine Learning and seeks to highlight its similarities and distinctions from related techniques. An overview of prominent efforts at making machine learning techniques and advanced analytics more accessible to non-experts is presented in Section 3, categorised by underlying data type. Section 4 presents a structural characterisation of the key interface elements of a generalised IML system. An abstracted description of the typical IML process is presented in

Section 5. Section 6 is an attempt to consolidate recent efforts by identifying a number of common solution principles. We highlight that tackling user interface design challenges and advancing the state of the art requires further research in several key areas. We therefore propose four distinct strands of research in Section 7 for enhancing the performance of Interactive Machine Learning applications from a user interaction perspective. Finally, we summarise the key findings in this article in Section 8.

2 CONTEXT AND DEFINITIONS

This article examines Interactive Machine Learning from a user interface design perspective. Developing an effective IML system is an interdisciplinary effort that draws on both the machine learning and HCI domains. To meet the objectives of this article, we concentrate primarily on IML studies that present user evaluations and offer suggestions for design of the interaction component. Limited attention is given to the underlying technical requirements for the machine learning component of the system beyond their implications for user interaction. In this section, we first seek to establish clear definitions for the terminology used throughout this article.

2.1 Interactive Machine Learning Definition

Interactive Machine Learning is an interaction paradigm in which a user or user group iteratively builds and refines a mathematical model to describe a concept through iterative cycles of input and review. Model refinement is driven by user input that may come in many forms, such as providing indicative samples, describing indicative features, or otherwise selecting high-level model parameters. Interactive Machine Learning is distinct from classical machine learning in that human intelligence is applied through iterative teaching and model refinement in a relatively tight loop of "set-and-check." In other words, the user provides additional information to the system to update the model, and the change in the model is reviewed against the user's design objective. Under this workflow, the change in the model at each iteration may be relatively small. This is in contrast to more traditional machine learning approaches where the workflow requires wholesale pre-selection of training data and significant changes in the model per execution step. Amershi et al. (2014) present a detailed discussion of the distinction between IML and classical machine learning as well as a series of cases studies that highlight the value of the technique. The IML process is characterised by the user being the principle driver of the interaction to deliver desired behaviour in the system. This does not mean that the computer has no influence on the process or does not make independent decisions. Indeed, the application may, for example, intelligently select a subset of data for review. The notion of the user being the principle driver is more reflected in the fact that the IML application seeks to provide the user with control over the high-level behaviour of the system. This may or may not necessarily be exercised in each discrete interaction.

Fails and Olsen (2003) introduced the concept of IML in the interactive image processing tool Crayons, as being chiefly focused on the review and correction of classifier errors. Crayons allows the user to draw directly on images to guide the training of an image classifier. The user reviews the current classifier performance based on its classification of image regions and provides more feedback by drawing on the image if necessary. ReGroup (Amershi et al. 2012) is a tool leveraging the Interactive Machine Learning paradigm to assist users in creating custom contact groups in a social network context. The system trains a model to classify contacts according to their likelihood of membership to the contact group being created. The system takes a selection of a contact for inclusion as a positive sample and the skipping of a contact in an ordered list as an implied negative sample. The user receives feedback in the form of an ordered list of suggested contacts after each selection. Just like Crayons, the user refines the behaviour of the model with incremental improvements derived through iteratively applied user input.

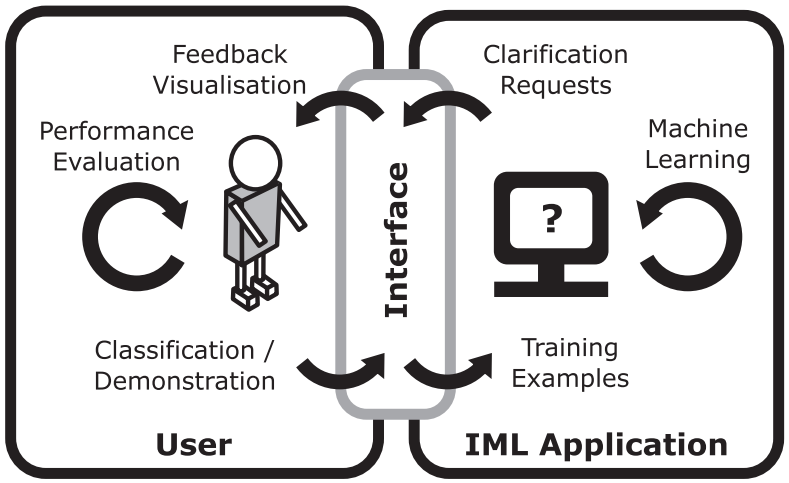

Fig. 1. The user plays a central role in Interactive Machine Learning by actively training the model in an iterative fashion. The user interface is a critical component of this interaction.

The at-completion model behaviour in the Crayons and ReGroup systems described above could of course be replicated using a standard machine learning approach that takes the collection of user inputs as a batch of labelled data. However, in the applications discussed, it is the process of refining the model itself that helps elicit the user input. These two examples help highlight the fact that the benefit of Interactive Machine Learning is observable in situations where the precise design objectives of the user are unclear and/or data labels cannot be obtained *a priori*.

More recent work extends the application of the interactive component beyond just review and correction of classifier errors to other stages of the model construction exercise, i.e., the choice of features, the choice of models, debugging and the evaluation of performance. In the context of this article, we refer to a *comprehensive* IML process as one in which the system delivers all of the necessary functionality to allow a user to configure then train a model on their data and deploy it in the target application (this full generalised workflow is described in detail in Section 5).

During the iterative refinement stage, the user drives the IML process in a tightly coupled arrangement between human and machine. The key elements and flow of information under this methodology are illustrated in Figure 1. The user's interaction may be viewed as a form of pseudo-programming in that the user builds functionality through demonstration and labelling samples rather than by directly writing code (Gillies et al. 2016). As an interaction paradigm, Interactive Machine Learning exists at the overlap between computer-as-tool and computer-as-partner as described by Beaudouin-Lafon (2004). An IML application certainly provides a tool in that it supports the user in building a model fit for their purposes. At the same time, the user and computer engage in an interaction that is a partnership where both sides must respond to input and to some extent, infer desired actions. The metaphor of moving an object with high and/or variable inertia along a desired trajectory provides a very high level interaction model for IML. This interaction model captures the fact that inputs may produce marginal or no response in the object and that a sequence of inputs is required to drive change.

The Interactive Machine Learning approach is likely to see initial adoption at the two extremes of user-data interaction tasks: at the lower complexity end, users selecting examples of things that they do and do not like; and at the higher complexity end, users (and even developers) applying

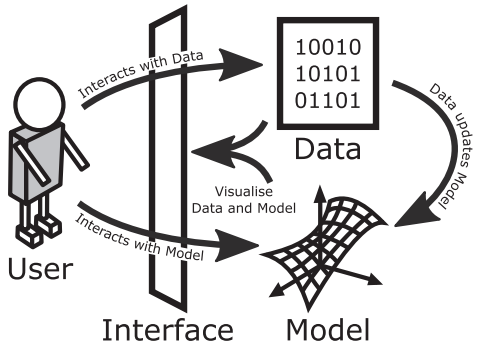

Fig. 2. Structural breakdown of a generic IML system. The *user* steers the *model* towards a desired concept by assigning or creating training *data*. The *interface* may also support direct interaction with the *model* by exposing parameters, editable structures, and choice in alternative techniques. Not independently represented in this figure but encompassed by the interface component, is the conceptual model for the system that frequently provides an abstraction layer between the user and the manipulated objects.

their domain expertise to refine a classifier or build some other intelligent functionality (Groce et al. 2014).

From a structural perspective, we separate an IML system into four key components: (1) the user, (2) the model, (3) the data, and (4) the interface. These components are summarised in Figure 2. The user drives the process by providing feedback and steering model training. The model is the component being trained. The data are a collection of the pre-existing and newly created points on which the model is trained. The interface is the bridge between the user and the model and data and provides the basis for interaction. The interface component is the primary focus of this article and will be dissected further in Section 4. We briefly define the role of each of these components within the IML process below.

2.1.1 User. As has been reiterated several times, the user is the main driver of the IML process. An assumption is made that the user possesses no deep understanding of machine learning techniques. Furthermore, it is unreasonable to expect a typical end-user to appreciate the subtleties of probabilistic inference. The user may possess significant domain expertise relevant to interpretation of the data and evaluation of model outputs. It is also worthwhile highlighting that the "user" may in fact be multiple users interacting with the system serially or in parallel.

The user is a dynamic and potentially unreliable component of the process. A single user's concept may drift over time (Kulesza et al. 2014) and inter-user variability in subjective annotations may be high (Curran et al. 2012). Unaddressed, these deficiencies are likely to have a negative effect on trained model quality. Fortunately, careful design of the interface, both in terms of the information presented and the guidance provided, can help to improve the consistency of users.

2.1.2 Model. The model is the component of the system that takes inputs and determines appropriate outputs based on its current understanding of the concept/process the user is seeking to capture. The high-level goal in machine learning generally is to build a model that accurately describes the relationship between input and output for a given concept based on data indicative of that concept. In machine learning, this model may be implemented using a wide variety of different algorithms and architectures. In Interactive Machine Learning, it is the model that the user is ultimately seeking to refine through their interactions. The user may manipulate the model directly, i.e., by adjusting parameters, or indirectly, i.e., by relabelling or adding new data. The

machine learning community provides an extensive range of modelling techniques, each with their corresponding advantages and disadvantages. It is important to note, too, that certain models, like certain users, may be unreliable in that their behaviour is not necessarily deterministic.

Model/algorithmic agnosticism is a design objective for IML systems but not readily achievable. An IML system that allows the user to switch between different model structures and algorithms while maintaining the same or similar interfaces and interactions provides greater flexibility without a significant re-familiarisation cost. Certain machine learning techniques are more suited to direct interaction and partial inspection than others. The machine learning community is hard at work developing new and more powerful techniques. There is growing interest in probabilistic machine learning methods that better capture and represent uncertainty (Ghahramani 2015). However, techniques such as deep neural networks (which have recently made significant advances) deliver functionality without any clear way to support inspection or interaction.

An IML process may also leverage multiple models, either to deliver ensembled functionality or for comparative evaluation purposes. An ideal IML system architecture would allow for state-of-the-art techniques to be introduced with minimal adjustment.

2.1.3 Data. The data component in an IML system is analogous to the source code in a conventional program. The user selects, describes or generates data to indicate how the model should behave in response to certain inputs. It can be useful to think of such a labelled data point as a training sample. Unlabelled data may also contribute to the construction of a model by inferring its association with other data points. Data type agnosticism is also a design objective for IML systems but similarly difficult to achieve. Users are likely to value the ability to apply a familiar IML system to whatever data type is dictated by the problem domain. IML systems have been developed targeting a range of data types including images, text, and motion data. Section 3 provides a survey of IML applications across different data types. The features relevant to each data type may vary considerably. Early studies in IML typically avoided user involvement at the feature level but there has been a recent shift towards greater transparency and even user involvement in feature selection. It does indeed seem desirable to apply domain expertise at the feature selection level in certain circumstances provided effective interfaces and interaction methods exist.

2.1.4 Interface. The user *interface* and its interplay with the user is the central focus of this article. A further decomposition of the interface into its potential sub-elements is presented in Section 4. Although this section presents a structural breakdown of an IML system, it is useful to highlight the distinction between the interface component and the actual interactions that it supports. The interactions available to a user are dictated by the interface, but as a designer, it is of course useful to think in terms of interactions when building an interface. Our discussion of the user interface design for IML thus encompasses the design of corresponding interaction techniques, and these two aspects are distinguished where possible.

The interface component is responsible for the bidirectional feedback between the user and the model/data. It must support both the input and output mechanisms necessary to provide this functionality. Constraints imposed on the richness of the labelling or demonstration process represent an efferent filter on the user's intent that may have negative consequences for user satisfaction and model quality. Limitations of the visual representation of sample points for review represent an afferent filter on the user's perception and understanding that may also degrade performance and satisfaction.

A central argument in this article is that the interface design is critical to the success of the IML process. Careful management of the user workflow and interactions can address the four key challenges raised in Section 1: (i) users can be imprecise and inconsistent, (ii) there is typically a degree of uncertainty in the relation between user input and user intent, (iii) interacting with a

model is not like interacting with a conventional information structure through an interface, and (iv) training is open ended.

2.2 Related Domains

The machine learning community has sought to make its techniques more accessible to end users. There are a growing number of packaged toolkits (e.g., WEKA (Hall et al. 2009) and Scikit-learn (Pedregosa et al. 2011)) and automated functionality (e.g., The Automatic Statistician (Lloyd et al. 2014)). Fiebrink's thesis introduces Wekinator (Fiebrink 2011), a stand-alone application for applying supervised machine learning to real-time data streams and particularly suited to music composition and performance.

The concept of "Machine Learning as a Service" (MLaaS) has also recently emerged in response to the demand for advanced analytics and data-driven applications within industry (Ribeiro et al. 2015). Many service-oriented machine learning platforms now exist including those offered by major companies such as Google, Microsoft, Amazon, and IBM. While these services do provide access to machine learning functionality paired with powerful computing back-ends, they still require familiarity with the techniques for their effective use.

Exposure of machine learning techniques to non-experts has occurred only more recently through applications such as email filters and recommender systems. Recommender systems are a form of Interactive Machine Learning in that the model of the user's preferences is iteratively constructed through interpreting their actions such as liking, disliking, purchasing, and reviewing. However, in many of these applications the user may be unaware that they are interacting with an algorithm that is learning in response to their inputs. Some recommender systems do explicitly inform users that they can influence the behaviour of their preference model and/or provide an interface to do so, but even in these cases, the interaction typically focusses more on data capture than on a bipartite process of closed-loop model refinement. In this article, we specifically focus on Interactive Machine Learning from an interface design perspective. While the algorithmic back-ends of recommender systems and their considered approach to implicit and explicit user input is certainly relevant to understanding Interactive Machine Learning more generally, the lack of focus given to the process excludes them from broader discussion in Section 3.

It is useful to explicitly highlight the distinction between Interactive Machine Learning and the related concept of Active Learning. Active Learning is a machine learning technique that focuses on selecting new points for labelling by the user. The key distinction from IML is that point selection is driven by the learner rather than the user. IML may also leverage learner-driven point selection strategies but not to the exclusion of user-driven input. For a survey of Active Learning, see Settles (2010). Reinforcement Learning is a machine learning technique that guides the learner towards a desired behaviour based on simple reward feedback. Research has examined the potential for humans to be positioned inside this loop as a means to steering the Reinforcement Learning algorithm. Knox and Stone (2015) provide a detailed overview of a series of experiments on reinforcement learning based on human derived reward feedback.

Beyond human–computer interactions, IML related concepts have also been explored in human–robot interaction research (Losing et al. 2015; Javdani et al. 2016; Gopinath et al. 2017). Gopinath et al. (2017) are motivated by the need to adjust levels of robotic assistance to accommodate the user's requirements. A motor impairment injury or illness may result in dynamic levels of capability even among a single user through the stages of recovery or degeneration. Supporting user-driven customisation of the level of assistance is a prime application for IML. There is also some overlap between IML and aspects of Knowledge Discovery in Databases (KDD) (see Holzinger (2013)) and Interactive Optimisation (see Meignan et al. (2015)). IML differs from these fields in the emphasis given to the role of the user and the process rather than the end goal.

3 SURVEY OF IML APPLICATIONS

Interactive Machine Learning has been applied over a diverse range of data types for a variety of end use applications. In this section, we present a survey of existing approaches and applications in Interactive Machine Learning. This article does not claim to provide an exhaustive survey of research in Interactive Machine Learning or its applications. The literature discussed herein is that which (i) helps to illustrate the different dimensions of the IML paradigm and/or (ii) gives special emphasis to the interface design considerations of IML systems. A provisional list of literature was obtained by performing a Scopus search on the quote-enclosed phrase *Interactive Machine Learning*. The full set of search results was reviewed and culled based on its contribution to (i) and/or (ii) above. Literature referenced by that in the result set is also included where relevant.

The categorisation employed here is based on the underlying data type explored in the application. It is important to note, however, that many of the interface features and algorithmic methods used are common across data types. An effort is made to highlight the unique requirements of developing an IML system for each data type while extracting the common features that generalise across application areas.

3.1 Text

Machine learning techniques are well suited to application on textual data. Text in specific domains can typically be obtained in high volumes and often already incorporates some degree of manual labelling. Modern spam filters are one example of effective machine learning based on textual data to improve the user experience.

Interactive Machine Learning systems have been developed to target use cases where a more constrained or custom categorisation is desired. Such categorisations and the features that characterise them may be difficult to tease out from domain experts and/or may be difficult to express programmatically. Wallace et al. (2012) present abstrackr, an online tool to support screening for systematic reviews of published medical evidence. The system can iteratively improve its classification of relevant citations as the user works through the provisional list, starting their manual evaluation with publications of highest relevance. In pilot evaluations, the system was able to reduce workload by approximately 40% by successfully reclassifying publications in the provisional list as irrelevant.

The similarly laborious task of statutory analysis is targeted by Šavelka et al. (2015). Šavelka et al. deploy an IML system to facilitate the process of reviewing regulatory frameworks to identify provisions relevant to a given topic (preparedness and response to public health emergencies in the case of the study). The user is presented with a classification of provisions and can modify the label assigned. They may also suggest terms that are important to this classification. At any time, the user may request that the model be re-trained based on their most recent feedback iteration. Interestingly, Šavelka et al. demonstrate that the model trained in one jurisdiction, i.e., Kansas, can be advantageously reused in another jurisdiction, i.e., Alaska, for the same topic analysis. This system is a prime example of the potential for IML to assist the user in completing a highly problem-specific task while also allowing the expended effort to be leveraged on related datasets.

Kim et al. (2015a) demonstrate an interactive interface for building and editing classifications of student code submissions. The approach was successfully applied to grading of coding tasks by allowing teachers to choose class exemplars and thereby generate a classification of different student answer types. Yimam et al. (2015) apply an interactive-learning process to the task of annotating medical abstracts. Huang et al. (2013) show the potential of IML to assist restaurant review authoring and reading. Through an interactive process, the system can learn to highlight and suggest summary sentiment in the three categories of food, service, and price. The study found that providing these suggestions during authoring can motivate users to correct erroneous

predictions. Endert et al. (2012) introduce the concept of semantic interaction in text analytics. They make use of the information embedded in interactions such as word search, highlighting, annotations, and adjustment of document pin locations to iteratively refine a model of document associations. This information is used to update a spatial visualisation of document associations.

Applying an IML process to textual data is challenged by the difficulty of providing succinct representations of documents that facilitate rapid user review. Kim et al. (2015b) compare a variety of machine learning techniques to the task of document feature compression. Their aim is to identify representations that improve the speed of categorisation assessment by users. Several of the discussed studies exploit term highlighting that indicates features relevant to the categorisation result. Giving salience to indicative keywords is hypothesised to speed up the assessment process. In terms of feature labelling, Raghavan et al. (2006) observe that users take up to five times longer to label one document, i.e., assign a document to a topic, than to label one feature, i.e., assign a word as an important term for a topic.

3.2 Images

The categorisation of images is a very popular application area within the machine learning community. The availability of suitable techniques for image based classification has encouraged the examination of the potential pairing of user input through IML. As for the application of IML to textual data, the usefulness of the technique is seen when more subtle concept discrimination capabilities are desired.

At the inter-image level, applications have sought to allow users to train models to understand their own, sometimes obscure, concepts. Fogarty et al. (2008) introduce CueFlik, an interactive application that allows users to create their own concepts to incorporate into image search. The user may define a new search rule by assigning a collection of images to describe a desired concept, e.g., scenic. The user may then apply this rule to their search so that only results consistent with that concept are provided. Fogarty et al. (2008) found that providing a visualisation of both the most-confident positive and most-confident negative samples led to higher quality and faster concept learning than providing the full ranking of images. Amershi et al. (2011a) extend CueFlik by adding features to the interface that better allow users to model changes and visualise the current model state. Amershi et al. also introduced a visualisation of historical model quality (expressed in terms of model reliability and snapshots in time of ranked images) and the ability to revert back to previous model states. Guo et al. (2016) present an IML system for clustering medical images based on the underlying condition and attempt to specifically incorporate expert constraints. Users can drag images apart to indicate that they should be disconnected in the model.

Other applications of IML have sought to train systems to recognise specific objects at the intra-image level. Crayons (Fails and Olsen 2003) is the often cited formative paper in IML that demonstrates an interface whereby users can incrementally train a classifier by directly drawing on an image. This approach has also been leveraged by Tsutsumi and Tateda (2009) to help measure the amount of leaf debris flowing on a river as well as for analysis of satellite, medical, and material science images (Porter et al. 2011, 2013; Harvey and Porter 2016; Kreshuk et al. 2011).

IML is well suited to use with images as there is a rich repository of features that have already been established, and the user can quickly review a large number of images. The ability to provide pixel or region level feedback by directly sketching on images is an interaction technique successfully exploited in several applications.

3.3 Time Series Data: Speech, Audio, Video, and Motion

The transcription of speech into text can be a laborious task when performed manually but there are an increasing number of systems that can autonomously perform this task at very low error

rates. Sanchez-Cortina et al. (2012) pair an automatic speech recogniser system with an IML interface that facilitates error correction. The acceptable word error rate can be adjusted to vary how many uncertain cases are shown to the user.

Kabra et al. (2013) employ an Interactive Machine Learning approach to allow non expert users to build classifiers for recognising distinct animal behaviours. The user reviews video footage of the animal and assigns behaviour labels to segments of the video. When a label is added, the user may initiate re-training of the model, and the classifier will be applied over the duration of the video. Upon encountering a behaviour that has been incorrectly classified, the user may correct the sample to incrementally improve model quality. The classifiers constructed through this process then allow for rapid and accurate annotation of animal behaviour in recorded datasets: a task that is very slow if performed manually and highly error prone if performed autonomously using less advanced methods. Versino and Lombardi (2011) apply an IML process to the filtering of surveillance streams for safeguarding a nuclear processing facility. Users assign positive and negative samples of safeguard-relevant events and the trained model can thus help identify potential hazards.

The challenging task of separating individual sources from a recorded audio track is explored by Bryan et al. (2014). Bryan et al. use a paint on approach (similar to that employed by Fails and Olsen (2003) on images) to train a classifier to recognise the desired source from within an audio file. The user "paints" on regions of a time-frequency visualisation of an audio recording to separate out sources.

BeatBox (Hipke et al. 2014) allows users to train a system to recognise different percussive vocalisations. Having established a custom vocabulary of vocalisations, users can then create a track that recognises and labels the vocalisations employed. The interface separates distinct vocalisation classes into different virtual pads and colour codes recorded vocalisations for a pad according to their recognised class. This colour coding provides a simple to interpret sense of the quality of a particular vocalisation class. A pad that contains recorded samples that are not accurately classified quickly attracts the attention of the user. They can then target a new class with greater discriminative power.

Motivated by relieving the prohibitive challenges people with disabilities face when playing conventional musical instruments, Katan et al. (2015) apply Interactive Machine Learning to the creation of gesture controlled instruments. Katan et al. enabled users to build a gesture vocabulary and customise the relationship between gestures and sounds. Sarasua et al. (2016) show the potential for customised gesture recognition to control music articulations, as a conductor might control an orchestra with a baton. Similar approaches have demonstrated the potential for IML to support unique forms of music composition (Fiebrink et al. 2011; Tsandilas et al. 2009) and interactive experiences (Kleinsmith and Gillies 2013; Brenton et al. 2014; Gillies et al. 2015). These studies highlight that the goal in IML is not necessarily just about making a task more efficient or productive but potentially about extending human creativity.

Others have sought to produce intelligent functionality in response to data streams such as for seizure detection (Chua et al. 2011) and for office task automation (Dey et al. 2004). IML has also been applied in training of brain computer interfaces (Kosmyna et al. 2015).

Gesture Script (Lü et al. 2014) is an interactive tool for creating two-dimensional gesture recognisers. The application allows users to generate feature level descriptions and combine these together to supplement training data. Hartmann et al. (2007) present an interaction authoring tool based on recognising patterns in generic sensor data.

As with textual data, the application of IML to time-series data can also be challenging in terms of providing summative representations of sample points: it can be difficult to succinctly represent the dynamic contents of an image stream or the temporal relationship between time and position that represents a gesture. There is a risk that the requirement for serial review of samples leads to

a highly serialised process that does not scale well. Interfaces designed for this purpose must find ways to allow users to quickly inspect and review samples.

3.4 Assisted Processing of Structured Information

The IML systems discussed in this section exploit structured data to train models that enhance user capabilities. These examples in particular highlight the application of domain or user-specific expertise to the training process. Amershi et al. (2012) present a social networking assistance tool based on the Interactive Machine Learning paradigm. The system builds a probabilistic model based on a Naïve Bayes classifier of contacts likely to be relevant to a particular group and makes suggestions and provides new filters for identifying new members.

Ghani and Kumar (2011) apply an IML process to greatly assist the work of auditors processing health insurance claims. In addition to learning to highlight fields of likely interest to support the auditor, the classification method can also help by grouping similar claims in the processing queue to reduce inefficiencies of context switching. This latter strategy is estimated to reduce time spent per claim by 27%.

CueT (Amershi et al. 2011b, 2011c) is an IML system that supports network alarm triage. The system can learn from the actions of operators to suggest alarm groupings. Kose et al. (2015) apply IML to facilitate the process of fraud detection in electronic claims data. The dynamic nature of the underlying data, and the potentially subtle relations that must be identified, motivate the use of a human-in-the-loop approach. AppGrouper (Chang et al. 2016) assists in the process of delivering topically coherent application clusters in an online application store. Users may directly edit clusters and encourage more or fewer clusters to be generated. In evaluation, the approach was shown to improve the quality of clustering results over algorithm-generated clusters. AppGrouper highlights the potential value in pairing human judgement with computational intelligence through a carefully designed interface. Human input is applied only at the stages where it is most useful.

The assisted processing applications discussed here highlight that certain domains will require highly specialised interface design. In many cases, however, the system level architecture may not require modification. Similarly, we hypothesise that the interaction and task level strategies that are effective are likely to remain consistent.

3.5 Raw Numerical Data

In an effort to demonstrate complete data type agnosticism, several studies have applied the IML process to numerical data that is abstracted from any particular application. There are parallels that can be drawn between Interactive Machine Learning and the uptake of spreadsheet applications that placed basic statistical and data representation tools in the hands of non-expert users. Indeed, Sarkar et al. (2015) have even demonstrated the capabilities of BrainCel, which adds machine-learning functionality to a typical spreadsheet type interface. The user can select rows as training data then click a "guess" button to make model predictions based on those rows.

Holzinger et al. (2016) incorporate human feedback with an Ant Colony Optimisation (ACO) framework to solve a traveling salesman problem. Users can select the path joining two cities and adjust the pheromone levels assigned using a slider. Ware et al. (2001) present an interactive tree-based classification application targeted at non-experts. Users create decision branches to build their own classifiers. Brown et al. (2012) present an iterative approach for learning user-preferred distance functions in a multidimensional scaling projection visualisation interface. The user may drag one or many points within a scatterplot visualisation window and then initiate the learning step to identify a new distance function that better reflects the user's desired data arrangement. This process can be performed iteratively, while preserving the user's interaction history, to incrementally find the most suitable distance function.

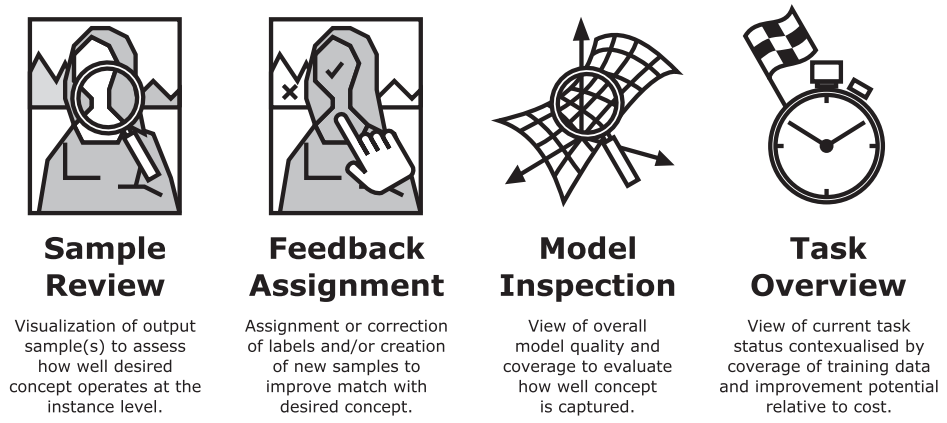

Fig. 3. Distinct interface elements in the IML process.

These studies illustrate that data independent IML systems can be developed and effectively used. This is not to say that abstracting data to a simplified representation is a preferable strategy, and in many cases this approach may be undesirable as it can limit the application of domain expertise. Nevertheless, what these approaches lose in terms of domain relevance they gain in flexibility.

4 COMPOSITION OF AN IML INTERFACE

In this section we describe the structural composition of a generalised Interactive Machine Learning interface. The design of an IML system may, of course, vary considerably depending on the nature of the data, the intended application of the model, and the user experience desired, as well as many other factors. Nevertheless, an understanding of the commonality can help inform the establishment of design principles.

The interface elements presented are based on an identification of the commonality among the broad range of IML systems reviewed. We are also guided by the requirements and desired attributes suggested by others. Amershi et al. (2011a) highlight three important aspects of an IML process that can be thought of as desired attributes of the interface: (1) illustrating the current state of the learned concept, (2) guiding the user to provide input that improves concept quality, and (3) providing revision mechanisms that allow the user to explore the model space. Stumpf et al. (2009) identify three activities relevant to the productive integration of the human user and machine learning techniques: (1) conveying system reasoning to the user, (2) conveying user reasoning to the system, and (3) ensuring both system and user profit from this feedback cycle. We split the third activity identified by Stumpf et al. into inspecting the model and reviewing task progress. The four key interface elements are then as follows: sample review, feedback assignment, model inspection, and task overview. These four interface elements are summarised in Figure 3. We do not suggest that these elements are necessarily visually distinct or indeed necessary in all applications of IML. For example, the sample review and feedback assignment interfaces may often be merged as in Crayons (Fails and Olsen 2003), and ad hoc implementations contrived for a one-time only custom task may have limited or no use for the model inspection or task overview interfaces. Rather, these elements represent distinct functionality that the interface must typically support to deliver a comprehensive IML workflow. Each of these interface elements are discussed in detail below.

4.1 Sample Review

The sample review interface is the channel on which individual or collections of model outputs are presented for review by the user. Sample instances may be requested manually by the user or chosen automatically based on model properties. Automatic selection of samples for review does not negate the principle of the user being in control. Ultimately, it is the user feedback provided on these samples (whether they are chosen manually or suggested) that will influence subsequent changes in the model. The literature highlights a challenge in managing the competing objectives of eliciting the necessary information to enhance the learner while avoiding excessive querying of the user. Users are unwilling to judge large numbers of test cases and this has been the motivation behind efforts that select representative and non-redundant samples for review (Ribeiro et al. 2016; Groce et al. 2014). In addition, the human input required, while of potential value to the model, may not be deemed worthwhile by the user (Amershi et al. 2014). Wong et al. (2011) observe that labelling data is a tedious exercise and that users may need to invest significant effort before a change in the learned model is noticeable.

Presenting samples or posing queries to the user such that they promote understanding rather than distrust is also difficult to achieve. The challenge of appropriately framing input requests is thus also dependent on the level of understanding possessed by the user.

The timing of user input requests is also critical as revealed by Losing et al. (2015) in their application of online user-in-the-loop labelling of obstacles encountered by a robot. The study highlights challenges around capturing training instances when they are relevant to the user versus the impact this may have on violating independent and identically distributed (i.i.d.) assumptions in the model.

The most appropriate means of succinctly displaying output samples for review may vary based both on data and application type. For example, the interface in an inter-image classification system might show the best and worst samples for a given classification. Alternatively, an intra-image classification system might show the regions of the image that belong to a given classification.

A guiding principle for the design of the sample review interface is that representations should, to the extent possible, allow the user to assess the current state of the learned concept. There is a distinction here between assessing the performance as reflected in a specific sample instance versus the generalised model performance as a whole. This latter assessment of the model as a whole is described in Section 4.3. Evaluating concept performance on an individual sample instance may require the provision of additional information on the reasoning behind the state shown. Kulesza et al. (2015) present an intuitive interface for model debugging involving generated explanations for the basis of particular predictions that are human readable with accompanying explanatory plots. For example, the classification of a particular message as being related to the topic "hockey" might be accompanied by the text, "This message has more important words about Hockey than about Baseball" (Kulesza et al. 2015). As Lü et al. (2014) observe, people want to understand why specific instances fail. Understanding why the model fails for a given sample may help the user determine the most appropriate feedback strategy, independent of broader model performance considerations. This suggests avoiding overly abstract or condensed representations of output samples.

4.2 Feedback Assignment

The feedback assignment interface is the channel employed by the user to assign labels, select features, and/or generate new samples. It is this interface that perhaps requires the most careful design in terms of both the interface elements and the interaction techniques provided. User interaction at the point of describing a classification or performing a demonstration can take many

forms. Porter et al. (2013) make a useful distinction between *training vocabulary* and *training dialog*. Porter et al. suggest that the training vocabulary in IML has expanded from just being labels to include constraints and structures. Similarly, the training dialog is no longer exclusively a batch process but can be highly interactive and dynamic. Generally, the preference from the user's point of view is for more complete and more specific feedback assignment methods. There is, however, an inevitable tradeoff as Hartmann et al. (2007) note in terms of balancing the desire for user control over behaviour versus overwhelming users with machine-centric "knobs."

BrainCel (Sarkar et al. 2015) obtains user input through standard spreadsheet paradigms while other studies have captured raw gestures inputs (Katan et al. 2015; Sarasua et al. 2016). Kim et al. (2015a) found that users wish to describe high-level attributes (such as code structure) that are difficult to encode as features. Constraining the labelling or demonstration process may have a filtering effect on the user's intent that may ultimately be detrimental to user satisfaction and model quality. Similarly, subjecting users to poorly implemented or confusing interaction techniques is likely to degrade overall performance.

One difficulty faced in the collection of input from users is the fact that the concept they are trying to train may shift over time (Kulesza et al. 2014). This has motivated efforts to improve the structure and guidance provided around how people are requested for and supply input. Rosenthal and Dey (2010) investigate what information can be provided to improve quality of labels (amounts of context, level of context, uncertainty, prediction, and requests for user feedback). Certain constructions can also make it more engaging for the user and promote higher quality input. For example, Huang et al. (2013) motivate users to fix errors in sentiment analysis of restaurant reviews by providing current predictions of the categories that their reviews cover.

Shilman et al. (2006) introduce five design principles for correction interfaces: (1) minimise decision points by choosing appropriate operators, (2) design seamless transitions between modes, (3) provide reachability of all states, (4) appropriately scope cascading changes, and (5) promote clear user models. Shilman et al. (2006) also summarise possible correction strategies: *n*-best selection, transformation, user-driven hints, and user-driven constraints.

Amershi et al. (2012) note that the ability to model a concept is dependent on the quality of the data. Various studies have sought novel methods for transforming both explicit and implicit user input into data for consumption by the learner. The system described in Amershi et al. (2012) takes advantage of users skipping instances (contacts in a social network) as an indicator for labelling negative samples. Javdani et al. (2016) seek to make use of user resistance to robot actions under shared autonomy as a means for identifying undesirable actions. The potential for feature labelling rather than just instance labelling is also being exploited in IML studies (e.g., Wong et al. (2011)) but does require that the learning algorithm can exploit these identified distinctions among features. In some applications, however, the features themselves may be too abstract for users to interpret or exploit effectively (Gillies et al. 2015).

4.3 Model Inspection

In addition to review of individual sample instances, a comprehensive IML process will typically also support some form of inspection at the model level. This summative view of model quality is referred to here as the *model inspection* interface. The assessment of model quality is not necessarily limited to prediction accuracy and may also incorporate other metrics such as coverage and confidence. Several studies have shown the potential for user interaction with parameters (e.g., Kapoor et al. (2010)) and/or emergent structures (e.g., Chang et al. (2016)) as part of the model inspection interface. GaussBox (Françoise et al. 2016) provides a tangible and interactive tool for inspecting HMMs trained for gesture recognition.

The quality of predictions is ultimately dependent on the quality of the constructed model. Various sources of error may affect model quality. Amershi et al. (2015) identify three main causes of model errors: (1) mislabelled data, (2) feature deficiencies, and (3) insufficient data. Mislabelled data are data the user incorrectly assigned some value. Feature deficiency errors result when the model does not have sufficient samples of a given feature to make a distinction. Insufficient data errors result when there are gaps in the data in areas relevant to model prediction. A comprehensive model inspection interface should thus support the user in identifying such errors to facilitate debugging and improve model quality.

Receiver Operating Characteristic (ROC) and cost curves are commonly used in the machine-learning community to support evaluation of model performance and have been incorporated into IML systems. Alternatively, model performance may be assessed in a more involved manner through targeted testing. Groce et al. (2014) suggest three requirements for test methods: algorithmic agnosticism, sufficiently fast to run in an interactive environment, and able to support effective detection of failures using comparatively small test suites. However, constructing tests to obtain an encompassing metric of model quality is not a simple task, especially for end users. Fiebrink et al. (2011) highlight that evaluating based on training set partitioning may not be meaningful and end users in IML may be particularly ill suited to choosing a test set that represents anticipated future data.

Recent work has given more attention to the quality assessment and debugging task in Interactive Machine Learning. While rapid visual inspection of model performance may be possible in certain applications, such as with image data, doing so with other data formats (e.g., documents) is more complicated. This difficulty grows with increasing dataset size. Amershi et al. (2015) present ModelTracker, a concise visual representation of model performance for use in debugging exercises. The representation allows for the rapid assessment of where labelled samples are correctly or incorrectly predicted and the relative confidence of that prediction.

The model inspection interface provides user visibility of the global task, i.e., typically minimising prediction error. This interface allows users to apply strategies that iteratively work towards desired levels of model quality. We argue later, however, that framing task completion criteria purely in terms of model quality is undesirable. Focusing an end user on model prediction accuracy alone is likely to promote overfitting. For this reason, we make a distinction between the *model inspection* and *task overview* interfaces.

4.4 Task Overview

A self-contained IML system may additionally require information advising the user on task status and termination conditions. In most experimental IML systems, this interface is not included as the interaction is managed through other means such as timed exercises or constrained datasets. In reality, however, there are non-trivial task related decisions that users require guidance on that may be independent of instantaneous model quality.

Yimam et al. (2015) highlight the fact that a text classification system is likely to reach a point of diminishing returns for user annotations. Ghani and Kumar (2011) also argue for a distinction between system performance and classification accuracy. The task overview interface should provide visibility of the global objectives but also contextualise these with other information relevant to the task such as the target application of the model and the availability of training data.

For commercial applications of IML in particular, the goal of the exercise is typically to increase the efficiency of expert users. Investing significant time to improve classification accuracy may not always correlate well with this goal. The task overview interface should help guide the machine-learning-naïve user in making these task-level assessments.

5 A GENERALISED WORKFLOW FOR A COMPREHENSIVE IML PROCESS

The previous section presents a structural breakdown of a generalised IML interface. Here we present a discussion of the task related behaviours associated with a generalised Interactive Machine Learning workflow. The workflow proposed by Fails and Olsen (2003) attaches a feedback loop to the training stage of the algorithm. In this workflow, the user interprets the feedback provided and then applies a correction. This workflow model is satisfactory if one constrains the scope of an IML system to just the training stage, but has its limitations if one is interested in developing a comprehensive IML process. Aodha et al. (2014) propose a 15 step "data understanding pipeline" motivated by supporting scientists in capitalising on advances in machine learning. This pipeline begins with "hypothesis formation" and ends with "publicizing of results" and provides a thorough breakdown of the intermediary tasks. The 15 step pipeline offered by Aodha et al. is comprehensive and informative but includes steps that occur both before and after the actual interaction with an IML application. Therefore, it is not particularly well aligned to the key interaction stages of the IML process that have distinct interface requirements. Girardi et al. (2015) presents a process model for domain-expert-in-the-loop knowledge discovery in biomedical research that removes the understanding the domain and understanding the data from the generic knowledge discovery process model. It includes six stages: (1) data modelling, (2) data acquisition, (3) data validation, (4) data preparation, (5) data analysis, and (6) evaluation. This process model highlights the key stages relevant to the knowledge discovery domain but does not adequately capture the interactive nature of IML.

We propose a generalised workflow that extends on the model of Fails and Olsen (2003) but is more concise than Aodha et al.'s "pipeline." We take some inspiration from Chang et al. (2016), who make a distinction between the types of user input captured at the early (initialising the algorithm), mid (steering the algorithm), and late (cluster editing) stages of the process. The six workflow activities we propose are as follows: (1) feature selection, (2) model selection, (3) model steering, (4) quality assessment, (5) termination assessment, and (6) transfer. This workflow is intended to serve as a useful reference for designers as they consider the task transitions and distinct interactions relevant to interface design for IML. It is formulated to be closely aligned with the key interaction stages and to be interpreted in combination with the structural breakdown of the IML interface presented in Section 4. This workflow is illustrated in Figure 4.

The model steering task is the core activity and where the user is likely to spend the majority of their time. The sub-tasks that occur inside this activity are essentially those described in Figure 1. The majority of studies in IML focus on this task alone, eliminating the need to examine the outer activities by appropriately initialising the system and constraining the scope of the experimental task. Indeed, certain application types may render some of these outer activities unnecessary. The workflow activities are not necessarily undertaken in a serial or ordered fashion. For example, model or feature selection may typically occur early in the process though not necessarily. Similarly, quality and termination assessment may typically occur only after several cycles of model steering have been completed. The transfer activity, however, will only occur on completion of all previous activities. We discuss each of the workflow activities in the remainder of this section.

5.1 Feature Selection

Fails and Olsen (2003) suggest that the need for explicit feature selection can be removed thanks to the IML process by incorporating a very large number of features and allowing the classifier to indirectly perform the filtering. While the feature selection activity may indeed be unnecessary in many applications where an appropriate repository of features exist, a more generalised workflow model for IML benefits from its inclusion.

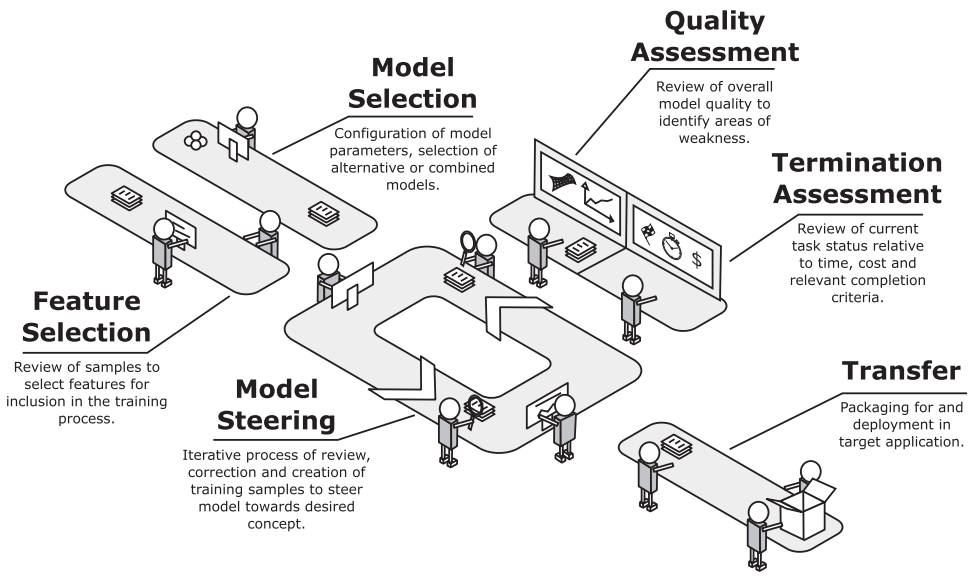

Fig. 4. The IML workflow as a behavioural breakdown into distinct user activities. The *model steering* activity is the iterative feedback-and-review cycle most commonly associated with the pure IML process. The user may initialise or reconfigure the process by engaging in *model selection* and *feature selection*. *Quality assessment* and *termination assessment* are depicted beside the main loop rather than within it. The intent here is to highlight that *model steering* is the primary activity but that this loop may be suspended temporarily, at any time, to undertake quality assessment and termination assessment. On successfully training the model to represent their desired concept, the user engages in *transfer* to deploy the developed model into the target application.

Allowing users to select features has been found to deliver both efficiency (Raghavan et al. 2006) and quality (Brooks et al. 2015) gains. Patel et al. (2008) suggest there is value in integrating data collection and feature creation tools. Kim et al. (2015a) observe a desire among users to introduce higher level features (such as code structure). The fact that human curated features are, by nature, more interpretable (Brooks et al. 2015) may also serve to improve user understanding in other activities in the process.

For other reasons, too, the features themselves may be more relevant to users than the classification they produce (Calderon et al. 2015). For example, rather than identifying how sentiment towards an administration is clustered, it may be more important to explore where and for what reasons this sentiment occurs.

Recent research has given more focus to the potential for human involvement in feature selection and construction. With Flock, Cheng and Bernstein (2015) show the potential for crowdsourcing to facilitate novel feature generation. Aggregating crowd features achieved better performance than just asking the crowd for a prediction or using a standard set of typical machine learning features. The Flock interface prompts users to think of discriminatory features by presenting contrasting samples. FeatureInsight (Brooks et al. 2015) is a tool that enables feature selection in website classification. Users are prompted to consider new dictionary features by being shown misclassified documents.

The studies discussed above suggest that end users are capable of, and indeed enjoy, identifying useful features for incorporation in the training process. Designing the IML system and interface

to accommodate the feature selection activity may thus be desirable, though not always practical. Certain data types and applications are better suited to end-user feature selection than others.

5.2 Model Selection

Studies in IML typically demonstrate the procedure using a machine learning technique specifically chosen by the authors for its applicability to the target data. This approach is entirely reasonable for application areas where the data and feature structures are known *a priori* or can be determined. However, in instances where an IML process is sought but the ideal machine learning technique is undetermined, it may be useful to allow the user to make this selection or perform a comparative evaluation. Furthermore, if one credits the proposition that an ideal IML system be agnostic to the underlying model and algorithm then there is value in considering model selection as an activity the user may perform.

The combination of multiple models through ensemble methods has also proven an effective strategy for the machine learning community. EnsembleMatrix (Talbot et al. 2009) demonstrates an interface that allows users to create combinations of classifiers trained on different features. It is not unreasonable to expect the user to weigh up the potential benefits and disadvantages of different models provided these can be concisely expressed for a given domain.

The selection of the most appropriate model (or their parameters) may also be influenced by the target application. For example, Fiebrink et al. (2011) comment in the context of gesture based music generation that the shape and smoothness of the decision boundaries may be more important to the user than their specific location.

It is perhaps overly optimistic to imagine an IML architecture in which the underlying learning techniques are completely modular and easily changed. Nevertheless, supporting the model selection activity by allowing the model to be configured by the user, even to a limited extent, can help to reduce the burden on machine learning practitioners. If the user can experiment with multiple different machine learning techniques within the same IML application, then the user can develop their own, albeit potentially naïve, sense of the relative advantages and disadvantages without the involvement of a practitioner.

5.3 Model Steering

The model steering activity is the stage in the workflow where most user effort is expended and with which the concept of IML is most traditionally associated. The majority of user focused research in IML examines this particular activity. While engaged in the steering task, the user is seeking to train the model to understand a given concept by iteratively refining the training data. This interaction will typically involve some combination of correcting existing erroneous samples, assigning labels to new samples or generating completely new samples. The emphasis, however, is generally on providing just enough feedback per iteration to push the model in the right direction. Fails and Olsen (2003) hypothesise that "having a very fast training algorithm is more important than strong induction." This is based on the assumption that users will apply an iterative strategy that will eventually deliver the necessary predictive power.

Fogarty et al. (2008) observe that users first focus on getting their classifier working and then seek to improve robustness by introducing more corner cases to the training. This is comparable to the approach followed by teachers in terms of first introducing the basic concept and then later exploring the extensions of this concept. Porter et al. (2013) refers to the exchange between the model and user that occurs during the model steering activity as the training dialog. The richness and flexibility of this dialog is directly influenced by the design of the interface and the interaction methods available. The Gesture Script application (Lü et al. 2014) provides an excellent example of a compelling and engaging dialog that shows how user intelligence can be applied to more than

just assigning or correcting labels. In addition to improving gesture recognition performance by providing new samples, Gesture Script allows users to create rendering scripts that describe the underlying structure of a gesture. For example, the user can specify that an arrow is composed of a line and a tringle-shaped head. This approach supports the synthesis of new gesture samples that help to introduce more variation and thus greater discriminative power. The model steering activity in Gesture Script thus incorporates multiple sub-tasks with which the user can engage and provide input where the greatest benefit is anticipated.

While the model steering activity is normally assumed to be iterative, Kleinsmith and Gillies (2013) found that users will not necessarily follow an iterative strategy. A portion of the participants in Kleinsmith and Gillies' study performed only one or two iterations in training their character behaviour model and preferred to expend their effort in carefully labelling samples. This observation highlights the fact that users may require high-level guidance on the strategy they should apply in the task of model steering (e.g., follow an iterative approach), and indeed, it is unreasonable to expect non-expert users to know what strategy is most appropriate. As Kleinsmith and Gillies suggest, the interface should promote this strategy through the interaction techniques available and the visual feedback presented.

Unfortunately users also suffer from inconsistency and boredom and this can result in changes in their task behaviour over time. Unavoidably, the user's perception of a concept may drift over the duration of a task. For example, what the user might consider to be a "scenic" image or a "local news" article can evolve as they are exposed to more samples. Similarly, the level of expressiveness that a user invests in the first articulation of a gesture may be radically different from their hundredth articulation. Structured Labeling (Kulesza et al. 2014) is an attempt to address concept drift during the training process. Kulesza et al. found that people labelling the same websites in two different sessions separated by 4 weeks were only 81% consistent. Providing more structure to the labelling task produced more consistent labelling. Cakmak and Thomaz (2014) demonstrate that automatic or heuristic-based teaching guidance can also improve the efficiency and quality of user feedback. In Cakmak and Thomaz's study, text-based instructions were presented to help guide the user in selecting positive and negative samples when training their model. In applications where an optimal training strategy can be described, these instructions can be generated algorithmically and in response to the model state. However, even heuristic-based teaching guidance was shown to improve user performance. For example, in the experimental application of teaching a facial expression recogniser, a heuristic for teaching an angry face might be "When you show examples of Angry face vary them as much as possible" (Cakmak and Thomaz 2014). This finding suggests that appropriate framing of the task and even relatively simple guidance in the interface can positively influence user performance and model quality.

Tsandilas et al. (2009) introduce the concept of "semi-structured delayed interpretation" of gestures. This approach encourages gestural annotations without having to specify their meaning. This preserves freedom of expression at gesturing time but still allows semantic meaning to be attached later. This approach may be very useful in situations where users experience concept drift by only requiring formal specification of meaning once the concept has solidified in the user's mind.

Ambiguity in subjective or interpreted concepts can also be problematic. Curran et al. (2012) observe that inter-user variation can be high when annotators are associating subjective attributes with images. Sarkar et al. (2016) achieve more consistent labelling by constructing the feedback task as a setwise comparison.

The model steering activity is the stage in the IML process at which most user effort and time is likely to be spent. Improving the speed and concept growth at each iteration step is thus likely to have the greatest impact on the overall efficiency and usability of the process.

5.4 Quality Assessment

The Interactive Machine Learning process can be alternatively viewed as a user-driven error minimisation loop placed around the model. The user seeks to train the model to some level of acceptable accuracy. The quality assessment task represents the activity of evaluating the trained system performance. Early in the process, the user may not execute this activity based on the assumption that the concept is unlikely to have been sufficiently captured. Indeed, the feedback provided in the sample review interface provides a proxy for system accuracy at this early stage. After sufficient training, the user may periodically check the model quality and compare this against some desired or pre-established threshold.

Amershi et al. (2011a) demonstrate visualisation of the current model accuracy as an interface element present during the standard model steering task. Amershi et al. suggest that this visualisation, coupled with an ability to undo actions, allows users to experiment with different strategies and evaluate their impact on accuracy. The quality assessment task may be interactive such as in the method proposed by Kapoor et al. (2010). ManiMatrix (Kapoor et al. 2010) provides a method for visualising and controlling classification behaviour. The user can directly interact with the confusion matrix and express preferred classification behaviour. Kapoor et al. suggest that this approach is superior to manual tuning of numerical parameters in terms of both speed and quality.

There is, however, a risk that users may focus too much on the quality assessment activity. Fiebrink et al. (2011) observe that users can inadvertently fall into a workflow that focuses on optimising cross-validation accuracy as an end in itself, rather than just using it as feedback to help refine their broader strategy. Supporting users in selecting the appropriate strategy and avoiding overfitting in the model requires careful framing of the quality assessment activity.

5.5 Termination Assessment

Limited attention has been given to the task of determining when to terminate the IML process. While the system may provide advice on termination conditions, it seems most fitting that the user ultimately decides when to end the process. This suggests inclusion of the termination assessment activity within the generic workflow.

There is potentially some overlap between termination and quality assessment activities. Certainly, in some cases the termination criteria may be entirely based on accuracy targets. However, Amershi et al. (2011a) observe that there is typically a degree of model decay between when peak performance is achieved and when users choose to stop. Supporting the identification of this inflection point may deliver improved performance in a comprehensive IML process.

There may, however, be alternative conditions for ending the process, such as further labelling will deliver negligible gain, there is already sufficient confidence in the particular classifications of value, and there is an increasing risk of overfitting. Sanchez-Cortina et al. (2012) describe a unique approach to allowing for dynamic adjustment of the user's workload by changing the acceptable word error rate in speech-to-text transcription. This approach allows the user to find an acceptable balance between supervision effort and recognition error. While allowing the user to configure an acceptable word error rate does not in itself set a termination condition per se, it does offer an interactive tool for exploration of whether further effort will yield worthwhile benefit.

Determining when to stop training under the IML process requires considerable judgment. The user's decision can, however, be informed by improved contextualisation of the various objectives and constraints relevant to the given application.

5.6 Transfer

The activity of deploying the trained model in the target domain is described here as *transfer*. In certain applications, this task may be as trivial as pointing to a test dataset. In other applications,

however, the transfer of model functionality into practical use may be considerably more complicated. While the IML workflow should endeavour to achieve good generalisability from the outset, this activity recognises the fact that tools are often applied in ways that cannot be predicted at design time. Packaging the model for end use and ensuring that it generalises is important to actually delivering an application that is useful.

Importantly, some applications such as abstrackr (Wallace et al. 2012) and the assisted claims processing system of Ghani and Kumar (2011) make no real distinction between training and deployment. The model learns and becomes more helpful as its predictive power increases. Nevertheless, it is useful to consider how one might subsequently transfer such developed models to the next context. This is indeed what Šavelka et al. (2015) explored in their reuse of their statute classifier trained in one jurisdiction in another jurisdiction. Talbot et al. (2009) observed that a model trained by a participant that achieved the highest accuracy on the training set did not generalise well to the test set due to overfitting. Arguably the transfer activity should be supported by tools that aid in identifying and correcting such situations.

The IML system demonstrated by Ghani and Kumar (2011) highlights the fact that deployment considerations can have significant effects on the efficiency of the process. For example, in the claims processing system, auditors were presented the next case based on similarity rather than confidence. This decision delivered a performance improvement by reducing the time lost to context switching.

The transfer activity has received limited attention in the literature on IML but should be recognised as a necessary stage in a comprehensive IML process. At the same time, it is important to note that not all applications require a transfer activity. Several of the applications reviewed in Section 3 do not involve an end-point or target dataset but rather focus on the process itself. Nevertheless, it is useful to consider the end use of trained models where appropriate and the implications this may have on the design of the interface.

6 EMERGENT SOLUTION PRINCIPLES

While the value of placing machine learning functionality in the hands of non-experts is gaining recognition, and emerging applications are enhancing user capability, there has been limited attention given to the generalised user interface design principles for such systems. This section seeks to consolidate observations from user studies in IML and synthesise these with relevant recognised HCI theory to establish some basic solution principles for interface design in IML. The solution principles are intended to serve as guidance to interface designers tasked with constructing effective interfaces for IML systems.

The specific interpretation and relative importance of the solution principles are likely to vary depending on the level of involvement expected of the user and on the complexity of the required functionality. Compare, for example, a user seeking to instruct a content suggestion service to obtain better recommendations, versus a user seeking to train a robot to perform some function in response to a given input. This across-application variability frustrates efforts to obtain concise and generalisable solution principles. Nevertheless, it is perhaps useful to view interactions of this type as being a kind of pseudo-programming task. This level of abstraction provides a useful frame for identifying the commonality.

We propose six key solution principles for the design of the IML interface:

(1) Make task goals and constraints explicit
(2) Support user understanding of model uncertainty and confidence
(3) Capture intent rather than input
(4) Provide effective data representations

(5) Exploit interactivity and promote rich interactions
(6) Engage the user

These principles cover multiple aspects across the different activities performed in the IML workflow. They also have specific implications for the interface components described in Section 4. Each is discussed in detail in the remainder of this section.

6.1 Make Task Goals and Constraints Explicit

Establishing clear task goals is an important part of Interactive Machine Learning for non-experts given that the process is largely user driven. Similarly important is an understanding of what is not possible, i.e., the constraints in the process. The iterative model refinement paradigm implies that users work in pursuit of some desired goal state. For users to construct effective strategies in pursuit of this goal state, it is necessary that they understand their task and its constraints.

This solution principle is largely a response to two of the key challenges for IML introduced in Section 1: users can be imprecise and inconsistent, and training is open ended. In terms of the first challenge, constructing explicit task goals and constraints can aid in focusing user efforts and improving consistency. Many recommender systems exploit user input for dual purposes, i.e., a like provides a signal of interest to a social group but may also inform predicted recommendations. While dual-purpose input with unclear objectives may be suitable for recommender system, we hypothesise that being explicit when requesting user input, whether singular, or multi-purpose, is preferable. It stands to reason that when asking a user to serve as a component in a closed feedback loop, understanding the information presented and actions required is paramount. However, too much information can also be confusing and detrimental to performance.

If user effort and attention is considered as a resource with limited supply, then it is sensible to ensure that all expended effort is in pursuit of the goal state. However, this goal state must be clearly defined. For example, Amershi et al. (2011a) observed that when a model reliability metric plot was introduced to the CueFlik interface to provide feedback on quality, users began to focus on maximising this quantity as their primary objective. Careful framing of the task goals and awareness of the interface features that may promote or hinder this understanding is required. The literature has explored and discussed the role that model and prediction explanations may have on the understanding and mental models of users (Kulesza et al. 2013; Sarkar 2015), but there has been less attention given to the effect that prior framing of the task may have on users. It is not unreasonable to hypothesise that supporting user perception of the global task and constraints may improve their ability to select model refinement actions although this should be investigated.

The second challenge revisited above (i.e., the fact that training is open ended) can also be mitigated in part through improved user understanding of goals and constraints. Interaction with machine learning models differs from more conventional computer interactions in that the user may have limited direct control over the behaviour exhibited. A key limitation of machine learning is that false positives/negatives produced are typically difficult to eradicate. For example, a user may identify a suggested movie as being completely outside their category of interest despite it possessing many attributes consistent with other favoured content. Amershi et al. (2012) note that "this problem occurs in all machine learning systems when the hypothesis language is insufficiently expressive to model the true concept." In other cases, numerous user actions may be required before the learning model reflects a desired change (Wong et al. 2011). An application that does not respond in a timely manner to user input is in violation of the principles of direct manipulation and is a potential cause of significant frustration. The degree to which a user understands their limitations in addressing such model failings may have a significant impact on their satisfaction with the interface and the refinement actions with which they proceed.

Without user understanding of the limitations of machine learning techniques, the process also exposes the potential risk of overfitting. Overfitting is the construction of a model that, although it matches to the data available, does not truly represent the phenomenon. Avoiding overfitting while supporting continued model refinement poses a unique challenge for supporting non-expert interaction. Expert machine learning practitioners are aware of this risk but naïve users may be prone to over-specification. The interactive tree-based classifier demonstrated in Ware et al. (2001) framed the task goal in a very clear and intuitive form but lead to some users expending significant effort to maximise classification accuracy with little real gain.

The user may in many cases not require an understanding of the global objective they are working towards provided they understand their role in the immediate interaction task. It is unclear what impact understanding the global task may have on user performance; however, we would argue it is essential to effective performance in the termination assessment and transfer activities.

6.2 Support User Understanding of Model Uncertainty and Confidence

Uncertainty is an inevitable feature of data-driven models in most real-world applications. Correspondingly, the concepts of uncertainty and confidence cannot be removed from the IML process. This solution principle seeks to address the fact that interacting with a model is not like interacting with a conventional information structure (one of the key challenges highlighted in Section 1). Both the task of steering the model as well as the ultimate application of the model to a target dataset involve uncertainty. It is important that users are aware of this uncertainty for two reasons: (1) to manage their expectations during interaction and (2) to manage their expectations of final system performance.

The concept of a probabilistic model and its limitations can be difficult to convey to non-experts and so many applications of IML are likely to rely on simplified explanations. Users without experience in machine learning are unlikely to comprehend the implications of working with a probabilistic model. User studies have found that even a single outlier in a classifier can result in significant confusion for users (Kim et al. 2015a). Users will calibrate their trust in the model both through individual predictions as well as the performance of the model as a whole (Ribeiro et al. 2016). Furthermore, IML is a co-adaptive process in that both the user and model will respond to the behaviour of the other (Gillies et al. 2016). Establishing the right level of understanding among users and framing the task appropriately is critical and non-trivial.

Non-experts unfamiliar with the internal behaviours of a computer program will construct their own mental model to aid their formulation of interaction strategies. This model will be derived in part from their past experience and knowledge. While the mental model constructed does not have to be accurate, a poor model may have a highly detrimental effect on user performance and thus, their perception of the effectiveness of the program (Norman 2014). It is perhaps useful to make the distinction between functional models that allow one to use a system versus structural models that allow one to comprehend how and why it works. Gillies et al. (2015) argue that users should be aided in their construction of conceptual models to enhance their debugging capabilities. As Fogarty et al. (2008) observe, evolution of the predictive model can result in seemingly unpredictable behaviour from the user's perspective. Kulesza et al. (2013) investigate the impact that different explanations have on the fidelity of the mental models constructed by end-users. The results indicate that more detailed explanations about intelligent agents are useful if added understanding can be leveraged by the user to improve outcomes. Sarkar (2015) proposes to exploit metamodels for confidence (is an output correct?), command (is the understanding complete?), and complexity (how simple was it to arrive at the output?) to augment machine learning models. Such metamodels would capture the information that is more intuitive and relevant for communication to end-users to support their understanding.

A number of strategies have been explored as a means to simplify the interpretation of model behaviour. ManiMatrix (Kapoor et al. 2010) allows users to interact directly with the classifier's confusion matrix and thereby steer classification behaviour. Ribeiro et al. (2016) present explanations that are locally faithful representations of considerably more complex models. This approach supports interpretation while hiding the potentially confusing complexity underneath. Vidulin et al. (2014), referencing constructivist learning theory, propose constraining the construction of decision trees to only represent relationships that are credible to the user. The use of exemplars to support understanding of classes appears to be a promising solution that resonates with users (Kim et al. 2015a). As a summative view of model quality, presenting best and worst matching samples has been shown to support more efficient model evaluation than ranking of the *n* best (Fogarty et al. 2008). ReGroup, the social network group creation tool introduced in Amershi et al. (2012), presents filters that were generated based on features in the model. Participants noted that these filters provided insight on the patterns that were being exploited by the model and thus served the dual purpose of explaining the model as well as their intended function as an interaction element.

Uncertainty can be difficult to represent succinctly in a user interface. Sarkar et al. (2015) demonstrated the potential for colouration to represent confidence within their BrainCel application; however, representing confidence through colouration in a speech recognition application (Vertanen and Kristensson 2008) did not yield an improvement in user performance. Within the field of information visualisation, the representation of uncertainty is a key area of investigation. In general, the objective of uncertainty visualisation is to provide representations that aid data analysis and decisions making (Pang et al. 1997). It can be useful to distinguish between different forms of uncertainty. Pang et al. (1997) describe three types of uncertainty: statistical (distribution of the data), error (delta compared to datum), and range (interval of possible values).

Within the machine learning community there is also keen interest in representing model quality in ways that support human understanding. The technique known as t-Distributed Stochastic Neighbour Embedding (t-SNE) enables the visual representation of clustering models (Van der Maaten and Hinton 2008). Such representations are easily queried and support non-expert reasoning on the level of confidence in the underlying model. Micallef et al. (2012) present an investigation of explanatory methods for supporting Bayesian reasoning. The observations of this study reveal the difficulty of the design problem in that text *without* numbers paired with visual aids yielded higher performance than text *with* numbers and visuals.

The literature suggests that there is likely to be a close relationship between user tolerance of error and the level of clarity in system uncertainty (Sacha et al. 2016). The degree of error a user will tolerate in an application is task specific (e.g., compare an error encountered while withdrawing money from an ATM versus an erroneous turn instruction given by a navigation system (Fogg and Tseng 1999)). If the user understands that they are in part responsible for an erroneous output, then they may be more forgiving in their perception of the system. Users will calibrate their trust of a system based on an understanding of the system properties. Muir and Moray (1996) argue that behaviours must be observable for trust to grow.

Making informed predictions represents a key challenge for Interactive Machine Learning. This is in contrast to more traditional programming tasks where systems are predictable and repeatable by nature. The uncertainty in the model is of fundamental interest to expert practitioners of machine learning. However, effectively interpreting model accuracy can be very difficult and a learned skill in itself.

6.3 Capture Intent Rather Than Input

Careful design of the interface may help to extract user intent from potentially noisy input actions. This solution principle is a response to the challenge raised in Section 1 that there is uncertainty

in the relationship between user intent and user input. Reducing this uncertainty is obviously an objective in HCI more generally, but the specifics of IML mean that confusion about user input is particularly detrimental to the process of training a model.

Fogarty et al. (2008) note that it is impossible for an algorithm to distinguish between what the user deems highly relevant samples versus what are perhaps only minor inconsistencies. CueFlik (Fogarty et al. 2008) helped mitigate this issue by displaying only the best and worst matches so that less certain training samples were typically out of view. The user is therefore guided towards focusing on samples that are definitely "good" or definitely "bad."

The study performed by Amershi et al. (2012) provides a number of potentially useful insights related to interactivity in terms of explicit and implicit user input. The Interactive Machine Learning application developed for building custom social networks exploits both explicit and implicit user action. For example, a user skipping past contacts is used as an indicator that these contacts should be labelled as negative samples. Ritter and Basu (2009) applies similar assumptions to extract implied user intent from cursor pass over behaviour in a file selection IML application. Self et al. (2016) present an informative study of how careful interaction design can assist in making better inferences about user intent from user actions. For example, Self et al. demonstrate how adding a circular selection tool when dragging a data point primes users to think about and select other points with which that point should be clustered.

As Porter et al. (2013) observes, "users are quite good at finding creative ways to use a small set of (inadequate) tools to reach their objectives, and so computers must learn to exploit a more unstructured dialog." In other words, in circumstances where the user cannot explicitly express their intent, they may still be able to achieve their goal by exploiting a sequence of actions. An example of this is Gesture Script (Lü et al. 2014), which allows users to synthesise gesture samples by drawing sub-segments and then programmatically describing how they combine together to form a complete gesture. To accommodate the "creative" ways the user may come to steer the process, the interface designer should thus consider avoiding overly constrained functionality or workflows.

6.4 Provide Efficient Data Representations

An effective IML interface will enhance user perception or at least make best use of human perception capabilities. The sub-task of reviewing sample outputs within the model steering activity in the IML process is largely a comprehension task. Improving the speed and quality of this operation has high impact given the iterative nature of the model steering activity. The adaptive interaction framework (Payne and Howes 2013) further suggests that the strategy employed by a user will be dictated by their experience, their task level goals, and their ability to process information relevant to the task. From a user interaction perspective then, this third factor suggests a potential lever in terms of amplifying the cognitive ability of the user that might be activated to improve performance and hence model quality.

Extracting information from small or simple datasets is typically achievable with standard analytic techniques. Machine learning comes into its own when the complexity and size of the target data would frustrate more established approaches. Consequently, end-user interaction with machine learning must also provide ways for viewing and interacting with voluminous, multi-dimensional and multi-modal data. Some interfaces support rapid understanding of model outputs such as the visualisation of image regions associated with different classification predictions (Ribeiro et al. 2016). Text classification applications may highlight words or n-grams to help the user perceive what features are being exploited by the model (Wallace et al. 2012). The general design objective would appear to be to maximise user perception of the features

relevant to understanding the function of the model and any of its deficiencies so that appropriate refinement actions can then be selected.

Interfaces designed to support interaction with complex data will typically aim to present a simplification of the data (e.g., clustered, hierarchically organised, subset, etc.) and/or amplify the user's cognitive ability (e.g., highlight relevant regions, support interaction, provide useful tools, etc.). Visualising multidimensional data in a sensible manner can be extremely difficult and the preferred representations may be highly data specific. This complicates efforts to generalise design principles for representation techniques in end-user applications. Parallel coordinates is one technique for transforming high-dimensional data into a representation with reduced dimensionality. However, collapsing dimensions typically degrades the visual information available and prevents more subtle patterns from being perceived. The presentation of multidimensional data may also introduce new obstacles to user cognition such as disorientation and occlusion. Researchers in information visualisation seek to enhance user cognition by making patterns, trends, relationships, outliers, and other correlations more immediately observable (Card et al. 1999). Interfaces for Interactive Machine Learning must also exploit these techniques. Stolper et al. (2014) offers four design guidelines for dynamic visualisations in progressive analysis: (1) avoid distractions due to excessive view changes, (2) highlight where new results have appeared, (3) provide a refresh capability to add new results, and (4) allow users to designate regions of interest and parts of the problem space to ignore.

6.5 Exploit Interactivity and Promote Rich Interactions

The user's interactions during the IML process are what drive model development. The model development process can thus be made more efficient by enabling users to fully express their intent and apply their insight. This solution principle suggests that the IML interface should be maximally interactive and should exploit rich forms of interaction where possible, for example, direct sketching on images, constructing explanations, and editing of samples.

6.5.1 Interaction for Understanding. Supporting user interaction with machine learning algorithms and complex datasets introduces new dimensions to the typical user interface design task. The interaction with a model rather than a distinct object makes it difficult to apply many of the user interface design principles that are known to be effective. The principles of direct manipulation proposed by Shneiderman (1982) are well recognised among interface designers. However, the nature of the task of building and interacting with a machine learning model may prohibit application of such heuristics, especially for non-expert users. In response, it is necessary to foster the construction of appropriate mental models among users that help frame this non-traditional interaction experience. Shneiderman (1996) presents a type by task taxonomy (TTT) that describes seven basic tasks performed by users on seven data types. The framework provides guidance on what features either help or hinder in performing different tasks on different forms of data. As an example, the application of filters to remove uninteresting data from view must be rapidly reflected in the display. However, as studies in IML have shown, it may take many user actions before an effect in the model is observable (Wong et al. 2011). Horvitz (1999) sought to bridge the principles of direct manipulation and enhanced computer autonomy through what he termed mixed-initiative interfaces. Horvitz identified 12 critical factors in ensuring effective collaboration with intelligent services. Importantly, Horvitz cautions against patching poor interface design by overusing machine intelligence and highlights the requirement to consider the design of both components in combination.

Interactivity may also be important at the point of understanding the sample data or inspection of whole datasets involved in the IML process. It is widely observed that interaction supports

understanding (Card et al. 1999; Chi and Riedl 1998) and so interactive data exploration tools are likely to be of value. This is particularly true for multidimensional or very large datasets that are difficult to visualise in a single view. Carpendale (2008) provides a summary of different tasks relevant to information visualisation and makes a distinction between low-level, detail-oriented tasks such as associating, ranking, clustering versus high-level, cognitive tasks such as identifying trends, causal relationships, and understanding uncertainty. Various data characteristics may frustrate simplistic approaches to visualisation, for instance, data properties such as non-linearity, holoarchy, and internal causality.

Inevitably there will be a coupling between the data type and the most suitable representation and interaction methods. A unique aspect of the model training task is the potential overlap in the sample review and feedback assignment interfaces. Creating a meaningful and productive interface that serves these dual functions requires careful design.

6.5.2 Make the Most of the User. The ability to reverse actions appears to be a highly valued interaction feature in IML. This observation is consistent with the "user control and freedom" usability heuristic proposed by Nielsen (1995). In constructing and refining a model, users require the ability to retrace steps in the event that recent actions have resulted in an undesired outcome (Talbot et al. 2009; Kapoor et al. 2010; Amershi et al. 2010). Related to the ability to "undo" is the provision of a visualisation of the history of model quality. Amershi et al. (2010) demonstrated that visualisation of model improvement or degradation is useful in guiding the non-expert user's efforts to refine the model.

Another interesting aspect of end-user interaction with machine learning is perception of relevance. Users want to feel like they are contributing value and that their inputs are utilised. A number of IML interfaces have attempted to address this issue by allowing users to label features rather than instances, such as allowing users to select words considered representative of a document class rather than just assigning documents to categories. This approach better exploits human insight and appears to promote user engagement. Sun and DeJong (2005) present an augmentation of a Support Vector Machine (SVM) that can incorporate domain knowledge to improve SVM learning. The study does not apply domain knowledge directly extracted from domain experts but does provide a suitable framework. The difficulty lies in elucidating that domain knowledge and transforming it into a format that is useful to the learner.

Users do not enjoy being repeatedly asked "yes" or "no" questions and so frustration and interruptability needs to be carefully managed. Cakmak et al. (2010) highlight that an ignorant learner asking many questions can be perceived as annoying. To reduce user annoyance, there are a number of potential strategies for minimising the frequency of interaction requests or minimising the impact of individual interruptions. For example, it may be possible to reduce the frequency of interaction requests by ensuring only questions of high relevance are posed to the user.

A novel approach suggested by Wallace et al. (2012), in seeking to exploit the knowledge and experience of domain experts, is to allocate classification labelling to different users depending on the confidence of the learner. The probabilistic measures generated for a particular instance may be useful in deciding whether to present that instance for review by a highly experienced user versus a less experienced user.

The importance of interactivity in IML is further emphasised by the fact that many applications in IML may only be feasibly evaluated by the end-users themselves (Groce et al. 2014). A user training a learner for a specific task must be provided with the tools to understand how well it performs. No external methods or testing can be applied to evaluate their specific use case on their specific dataset.

The execution speed of the learning algorithm may also have implications for interactivity. Fails and Olsen (2003) hypothesise that fast algorithms are more important than inductive power. Versino and Lombardi (2011) express this attribute in another way as the need to ensure computation time at each iteration stays within human acceptable levels. These observations are consistent with the direct manipulation concept of "rapid incremental reversible operations whose impact on the object of interest is immediately visible" (Shneiderman 1982). However, certain machine-learning techniques applied on large volumes of data may unavoidably have very large execution times. Careful design of the interface can help to mitigate the imposition on interactivity under these circumstances.

6.6 Engage the User

An effective system will engage the user in the task being performed so that they are motivated to achieve the desired outcomes of the IML application. This engagement, however, should not come at the expense of excessive mental load. Early et al. (2016) suggest that presenting partial predictions can keep users engaged by encouraging them to actively improve prediction quality. Such feedback, even though incomplete, can be helpful in reminding the user that they are a critical component in the process and that their activities are having an effect.

Porter et al. (2013) suggest a distinction between users of IML operating in domains where they must be enticed for feedback versus applications such as those in science, engineering, health and defence in which users will be inherently motivated to train a high quality model. In the latter case, users may seek to employ highly sophisticated strategies whereas in the former case, the objective may be to minimise imposition on the user. Although dependent on the application, users typically do not enjoy performing trivial labelling tasks or responding to repeated yes/no type questions (Cakmak et al. 2010) and do seek to provide insight where it exists (Kim et al. 2015a). Nevertheless, users exhibit bounded rationality and will satisfice based on perceived gains versus effort (Pirolli and Card 1995). Promoting engagement may increase the time users are willing to invest in the refinement process with corresponding benefits in trained model quality.

Reducing the effort to both interpret outputs (see Section 6.4) and express feedback (see Section 6.5) may enhance user engagement. Summative views such as those typical in image focussed IML systems are likely far more engaging than the comparable view in a textual data IML system. Similarly the level of interactivity supported in the task may also serve to promote user engagement. Interfaces such as those introduced in Katan et al. (2015) and Sarasua et al. (2016) allow users to generate and demonstrate samples through physical activity.

Providing intuitive representations of progress in training a classifier, such as ModelTracker (Amershi et al. 2015), are another potential way to allow users to engage in the task. Huang et al. (2013) found that users who could visualise the current predictions related to their restaurant reviews were motivated to fix any issues.

7 OPEN RESEARCH PROBLEMS

There are emerging areas of investigation in Interactive Machine Learning that promise many new and powerful applications. Many other valuable avenues of research also require attention from a user interaction perspective. The following research strands are proposed based on the gaps identified in the literature and also gaps identified through the consolidation and synthesis process applied in preparing this article.

7.1 Priming and Guiding the User

The role of the user's mental model in the IML process has received some recent attention (Kulesza et al. 2013; Sarkar et al. 2015) as has the potential for improved task guidance during

training (Cakmak and Thomaz 2014; Kulesza et al. 2014). These studies show that relatively simple strategies can be applied to improve user consistency and understanding. In addition to extending this work on guiding task level strategies at execution, further research should examine the initial instructions and framing given to users. An improved understanding of appropriate user priming is perhaps more likely to be generalisable across IML applications than execution related guidance. This in turn may help highlight more concrete solution principles related to the formation of user mental models in IML.

As more comprehensive IML processes covering the full range of activities described in Section 5 emerge, there is value in investigating the user guidance required to inform decisions on switching between the global task activities. It is necessary that users can build effective strategies that in some way relate the model steering activity to the broader task goals and related activities such as model configuration and deployment.

7.2 Capturing, Representing, and Reasoning with Uncertainty

As highlighted earlier, a key aspect of prediction using a model is uncertainty. The concept of data and prediction uncertainty is not always well understood by end users who traditionally associate computers with precision and repeatability. Furthermore, uncertainty can be difficult to succinctly represent visually. Micallef et al. (2012) show that non-experts struggle in probabilistic reasoning tasks, even when concise visual and textual guidance is provided. While the most appropriate approach may differ depending on the data and application, Bayesian machine learning methods provide a distinct advantage in their ability to incorporate the notion of uncertainty (Ghahramani 2015).

There is value in studies investigating the application of these approaches coupled with intuitive uncertainty visualisation techniques. Horvitz's early work on mixed-initiative interfaces applied probabilistic models of user intention to support selection of different interface modes. Similar methods may be appropriate for capturing uncertainty in user input. The specific advantage of a Bayesian framework is to embed the estimation of uncertainty from the outset as well as throughout the model components. Allowing users to view model uncertainty and its evolution may support improved understanding and performance but care must be taken to ensure such feedback provides genuine insight without adverse secondary effects, such as becoming a distraction or attracting attention away from other critical aspects of the interface.

7.3 Enhancing Perception, Interaction, and Engagement via Immersive Environments

Holzinger and Jurisica (2014) argue for tools that are interactive and representations that support high dimensions and multiple modalities. The recent advancement in augmented and virtual reality head mounted displays presents new opportunities for users to interact with and explore complex multidimensional datasets. It is reasonable to propose that more natural interaction may thus support improved understanding and reduced perceived cost of model refinement.

The placement of a user into an immersive environment can also facilitate more natural mechanisms of discovery and learning. Gillies et al. (2015) show the potential of full bodied interaction as part of an IML process. The placement of users within an immersive environment is likely to have particular relevance to IML processes applied in gesture and motion recognition applications. Immersive three-dimensional interfaces also offer additional data representation opportunities over conventional two-dimensional displays. A further ancillary benefit is that an enjoyable immersive experience may also improve user engagement. However, as the mass of research effort in the field of information visualisation shows, the construction of intuitive and informative multi-dimensional data representations is non-trivial.

7.4 Evaluating Interfaces and User Performance in IML

The task of evaluating user interfaces for Interactive Machine Learning requires careful consideration given the various characteristics that distinguish them from more traditional user interfaces. In the case of a user training a model to capture their own personal concept, Groce et al. (2014) highlight that potentially only that particular individual can properly evaluate the quality of their model.

Additionally, interfaces designed for cognitively intense tasks can be difficult to evaluate due to the longitudinal nature of analysis and learning. A key challenge for evaluating information visualisations designed to provide enhanced cognition is the fact that insight is "temporally elusive" in that a stimulus may trigger a response long after exposure (Carpendale 2008).

Gillies et al. (2016) point to different traditions for evaluation of developed methods between the HCI and machine learning communities (i.e., user studies versus benchmark testing). Regrettably, unlike in traditional machine learning research where alternate techniques may be repeatably evaluated on a single dataset, a human user, once exposed to a problem as part of an IML evaluation, cannot unlearn their experience and be retested. The HCI community may do well to establish representative datasets, and perhaps even representative structured tasks, relevant to the specific challenges introduced by Interactive Machine Learning. This may foster better engagement from the broader machine learning community while at the same time support the identification of generalisable requirements and enable comparative evaluation of interfaces.

8 CONCLUSIONS

Machine learning techniques are slowly creeping into the lives of non-expert users. Enabling users to efficiently interact with such algorithms is likely to be a key design challenge in the coming decade. Interactive Machine Learning promises new opportunities for assisting users in data intensive processing tasks, enhancing outcomes in analytic tasks, and supporting the development of comparatively complex functionality without explicit programming. IML is a co-adaptive process, driven by the user, but inherently dynamic in nature as the model and user evolve together during training. The user may adjust their strategy in response to the observed behaviour in the model, and the model correspondingly changes but in ways that are not entirely predictable.

There are aspects of the task of training a model that make the interaction requirements distinct from more conventional human–computer interfaces. It is important that the application of new and exciting machine intelligence is accompanied by careful design of the user interfaces for such applications. Research is beginning to focus more on the interface features that make such applications effective and enjoyable to use. Studies have shown that even non-expert users want to understand more about the models they are interacting with and how they can refine their accuracy.

This article has presented a survey of prominent efforts to apply the IML process in a range of applications. The survey illustrates that the IML approach has utility on a range of underlying data types. Furthermore, there is a degree of commonality across implementations. A synthesis of the literature forms the basis for the structural and behavioural IML models presented. The interface elements identified in Section 4 provide a generalised perspective on the interface design problem. The workflow described in Section 5 covers the activities involved in the IML process from feature and model selection up to transfer of the learned model into the target application. A consolidation of user interface related findings from the literature is presented as a set of emergent solution principles for IML. It is hoped that this set of solution principles will provide guidance to the interface designer in terms of both framing the IML process and developing productive and enjoyable functionality. The preparation of this article also led to the identification of several open

problems for the field of IML. We propose four strands of investigation for advancing the state of the art.

As users are increasingly attempting to gain insights from ever larger volumes of data and more and more user interfaces become driven by machine learning algorithms, Interactive Machine Learning is likely to become a more central theme in user interface design. We believe that there is value in establishing a generic model and provisional set of solution principles relevant to the Interactive Machine Learning process. This article is an attempt to provide that foundation and it is hoped that further efforts will refine and expand on these concepts.

REFERENCES

Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Mag.* 35, 4 (2014), 105–120. DOI: http://dx.doi.org/10.1609/aimag.v35i4.2513

Saleema Amershi, Max Chickering, Steven M. Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. 2015. ModelTracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI'15)*. ACM, New York, NY, 337–346. DOI: http://dx.doi.org/10.1145/2702123.2702509

Saleema Amershi, James Fogarty, Ashish Kapoor, and Desney Tan. 2010. Examining multiple potential models in end-user interactive concept learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'10)*. ACM, New York, NY, 1357–1360. DOI: http://dx.doi.org/10.1145/1753326.1753531

Saleema Amershi, James Fogarty, Ashish Kapoor, and Desney Tan. 2011a. Effective end-user interaction with machine learning. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*. AAAI Press, 1529–1532.

Saleema Amershi, James Fogarty, and Daniel Weld. 2012. ReGroup: Interactive machine learning for on-demand group creation in social networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'12)*. ACM, New York, NY, 21–30. DOI: http://dx.doi.org/10.1145/2207676.2207680

Saleema Amershi, Bongshin Lee, Ashish Kapoor, Ratul Mahajan, and Blaine Christian. 2011c. CueT: Human-guided fast and accurate network alarm triage. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'11)*. ACM, New York, NY, 157–166. DOI: http://dx.doi.org/10.1145/1978942.1978966

Saleema Amershi, Bongshin Lee, Ashish Kapoor, Ratul Mahajan, and Blaine Christian. 2011b. Human-guided machine learning for fast and accurate network alarm triage. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*. AAAI Press, 2564–2569. DOI: http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-427

Oisin Mac Aodha, Vassilios Stathopoulos, Gabriel J. Brostow, Michael Terry, Mark Girolami, and Kate E. Jones. 2014. Putting the scientist in the loop – Accelerating scientific progress with interactive machine learning. In *Proceedings of the 2014 22nd International Conference on Pattern Recognition*. 9–17. DOI: http://dx.doi.org/10.1109/ICPR.2014.12

Michel Beaudouin-Lafon. 2004. Designing interaction, not interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI'04)*. ACM, New York, NY, 15–22. DOI: http://dx.doi.org/10.1145/989863.989865

Harry Brenton, Andrea Kleinsmith, and Marco Gillies. 2014. Embodied design of dance visualisations. In *Proceedings of the 2014 International Workshop on Movement and Computing (MOCO'14)*. ACM, New York, NY, 124–129. DOI: http://dx.doi.org/10.1145/2617995.2618017

Michael Brooks, Saleema Amershi, Bongshin Lee, Steven M. Drucker, Ashish Kapoor, and Patrice Simard. 2015. FeatureInsight: Visual support for error-driven feature ideation in text classification. In *Proceedings of the 2015 IEEE Conference on Visual Analytics Science and Technology (VAST'15)*. 105–112. DOI: http://dx.doi.org/10.1109/VAST.2015.7347637

E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. 2012. Dis-function: Learning distance functions interactively. In *Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology (VAST'12)*. 83–92. DOI: http://dx.doi.org/10.1109/VAST.2012.6400486

Nicholas J. Bryan, Gautham J. Mysore, and Ge Wang. 2014. ISSE: An interactive source separation editor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'14)*. ACM, New York, NY, 257–266. DOI: http://dx.doi.org/10.1145/2556288.2557253

Maya Cakmak, Crystal Chao, and Andrea L. Thomaz. 2010. Designing interactions for robot active learners. *IEEE Trans. Auton. Mental Dev.* 2, 2 (Jun. 2010), 108–118. DOI: http://dx.doi.org/10.1109/TAMD.2010.2051030

Maya Cakmak and Andrea L. Thomaz. 2014. Eliciting good teaching from humans for machine learners. *Artif. Intell.* 217 (Dec. 2014), 198–215. DOI: http://dx.doi.org/10.1016/j.artint.2014.08.005

Nadya A. Calderon, Brian Fisher, Jeff Hemsley, Billy Ceskavich, Greg Jansen, Richard Marciano, and Victoria L. Lemieux. 2015. Mixed-initiative social media analytics at the World Bank: Observations of citizen sentiment in Twitter data to explore "trust" of political actors and state institutions and its relationship to social protest. In *Proceedings of the 2015 IEEE International Conference on Big Data (Big Data'15)*. 1678–1687. DOI: http://dx.doi.org/10.1109/BigData.2015.7363939

Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. 1999. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, San Francisco, CA.

Sheelagh Carpendale. 2008. Evaluating information visualizations. In *Information Visualization: Human-Centered Issues and Perspectives*, Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, and Chris North (Eds.). Springer, Berlin, 19–45. DOI: http://dx.doi.org/10.1007/978-3-540-70956-5_2

Shuo Chang, Peng Dai, Lichan Hong, Cheng Sheng, Tianjiao Zhang, and Ed H. Chi. 2016. AppGrouper: Knowledge-based interactive clustering tool for app search results. In *Proceedings of the 21st International Conference on Intelligent User Interfaces (IUI'16)*. ACM, New York, NY, 348–358. DOI: http://dx.doi.org/10.1145/2856767.2856783

Justin Cheng and Michael S. Bernstein. 2015. Flock: Hybrid crowd-machine learning classifiers. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW'15)*. ACM, New York, NY, 600–611. DOI: http://dx.doi.org/10.1145/2675133.2675214

E. H. Chi and John Riedl. 1998. An operator interaction framework for visualization systems. In *Proceedings of the 1998 IEEE Symposium on Information Visualization (INFOVIS'98)*. IEEE Computer Society, Washington, DC, 63–70. http://dl.acm.org/citation.cfm?id=647341.721078

Eric C.-P. Chua, Kunjan Patel, Mary Fitzsimons, and Chris J. Bleakley. 2011. Improved patient specific seizure detection during pre-surgical evaluation. *Clin. Neurophysiol.* 122, 4 (2011), 672–679. DOI: http://dx.doi.org/10.1016/j.clinph.2010.10.002

William Curran, Travis Moore, Todd Kulesza, Weng-Keen Wong, Sinisa Todorovic, Simone Stumpf, Rachel White, and Margaret Burnett. 2012. Towards recognizing "cool": Can end users help computer vision recognize subjective attributes of objects in images? In *Proceedings of the 17th International Conference on Intelligent User Interfaces (IUI'12)*. ACM, New York, NY, 285–288. DOI: http://dx.doi.org/10.1145/2166966.2167019

Anind K. Dey, Raffay Hamid, Chris Beckmann, Ian Li, and Daniel Hsu. 2004. A CAPpella: Programming by demonstration of context-aware applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'04)*. ACM, New York, NY, 33–40. DOI: http://dx.doi.org/10.1145/985692.985697

Kirstin Early, Stephen E. Fienberg, and Jennifer Mankoff. 2016. Test time feature ordering with FOCUS: Interactive predictions with minimal user burden. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'16)*. ACM, New York, NY, 992–1003. DOI: http://dx.doi.org/10.1145/2971648.2971748

Alex Endert, Patrick Fiaux, and Chris North. 2012. Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'12)*. ACM, New York, NY, 473–482. DOI: http://dx.doi.org/10.1145/2207676.2207741

Jerry Alan Fails and Dan R. Olsen, Jr. 2003. Interactive machine learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces (IUI'03)*. ACM, New York, NY, 39–45. DOI: http://dx.doi.org/10.1145/604045.604056

Rebecca Fiebrink, Perry R. Cook, and Dan Trueman. 2011. Human model evaluation in interactive supervised learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'11)*. ACM, New York, NY, 147–156. DOI: http://dx.doi.org/10.1145/1978942.1978965

Rebecca Anne Fiebrink. 2011. *Real-time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance*. Ph.D. thesis. Princeton University, Princeton, NJ.

James Fogarty, Desney Tan, Ashish Kapoor, and Simon Winder. 2008. CueFlik: Interactive concept learning in image search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'08)*. ACM, New York, NY, 29–38. DOI: http://dx.doi.org/10.1145/1357054.1357061

B. J. Fogg and Hsiang Tseng. 1999. The elements of computer credibility. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'99)*. ACM, New York, NY, 80–87. DOI: http://dx.doi.org/10.1145/302979.303001

Jules Françoise, Frédéric Bevilacqua, and Thecla Schiphorst. 2016. GaussBox: Prototyping movement interaction with interactive visualizations of machine learning. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA'16)*. ACM, New York, NY, 3667–3670. DOI: http://dx.doi.org/10.1145/2851581.2890257

Zoubin Ghahramani. 2015. Probabilistic machine learning and artificial intelligence. *Nature* 521 (2015), 452–459. DOI: http://dx.doi.org/10.1038/nature14541

Rayid Ghani and Mohit Kumar. 2011. Interactive learning for efficiently detecting errors in insurance claims. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*. ACM, New York, NY, 325–333. DOI: http://dx.doi.org/10.1145/2020408.2020463

Marco Gillies, Harry Brenton, and Andrea Kleinsmith. 2015. Embodied design of full bodied interaction with virtual humans. In *Proceedings of the 2nd International Workshop on Movement and Computing (MOCO'15)*. ACM, New York, NY, 1–8. DOI: http://dx.doi.org/10.1145/2790994.2790996

Marco Gillies, Rebecca Fiebrink, Atau Tanaka, Jérémie Garcia, Frédéric Bevilacqua, Alexis Heloir, Fabrizio Nunnari, Wendy Mackay, Saleema Amershi, Bongshin Lee, Nicolas d'Alessandro, Joëlle Tilmanne, Todd Kulesza, and Baptiste Caramiaux. 2016. Human-centred machine learning. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA'16)*. ACM, New York, NY, 3558–3565. DOI: http://dx.doi.org/10.1145/2851581.2856492

Marco Gillies, Andrea Kleinsmith, and Harry Brenton. 2015. Applying the CASSM framework to improving end user debugging of interactive machine learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI'15)*. ACM, New York, NY, 181–185. DOI: http://dx.doi.org/10.1145/2678025.2701373

Dominic Girardi, Josef Kueng, and Andreas Holzinger. 2015. A domain-expert centered process model for knowledge discovery in medical research: Putting the expert-in-the-loop. In *Proceedings of the International Conference on Brain Informatics and Health*. Springer, 389–398. DOI: http://dx.doi.org/10.1007/978-3-319-23344-4_38

D. Gopinath, S. Jain, and B. D. Argall. 2017. Human-in-the-loop optimization of shared autonomy in assistive robotics. *IEEE Robot. Automat. Lett.* 2, 1 (Jan 2017), 247–254. https://doi.org/10.1109/LRA.2016.2593928

A. Groce, T. Kulesza, C. Zhang, S. Shamasunder, M. Burnett, W. K. Wong, S. Stumpf, S. Das, A. Shinsel, F. Bice, and K. McIntosh. 2014. You are the only possible oracle: Effective test selection for end users of interactive machine learning systems. *IEEE Trans. Softw. Eng.* 40, 3 (Mar. 2014), 307–323. DOI: http://dx.doi.org/10.1109/TSE.2013.59

Xuan Guo, Qi Yu, Rui Li, Cecilia Ovesdotter Alm, Cara Calvelli, Pengcheng Shi, and Anne Haake. 2016. An expert-in-the-loop paradigm for learning medical image grouping. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 477–488. DOI: http://dx.doi.org/10.1007/978-3-319-31753-3_38

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *ACM SIGKDD Explor. Newslett.* 11, 1 (Nov. 2009), 10–18. DOI: http://dx.doi.org/10.1145/1656274.1656278

Björn Hartmann, Leith Abdulla, Manas Mittal, and Scott R. Klemmer. 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 (CHI'07)*. ACM, New York, NY, 145–154. DOI: http://dx.doi.org/10.1145/1240624.1240646

Neal Harvey and Reid Porter. 2016. User-driven sampling strategies in image exploitation. *Inf. Vis.* 15, 1 (2016), 64–74. DOI: http://dx.doi.org/10.1177/1473871614557659

Kyle Hipke, Michael Toomim, Rebecca Fiebrink, and James Fogarty. 2014. BeatBox: End-user interactive definition and training of recognizers for percussive vocalizations. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI'14)*. ACM, New York, NY, 121–124. DOI: http://dx.doi.org/10.1145/2598153.2598189

Andreas Holzinger. 2013. Human-computer interaction and knowledge discovery (HCI-KDD): What is the benefit of bringing those two fields to work together? In *Availability, Reliability, and Security in Information Systems and HCI*, Alfredo Cuzzocrea, Christian Kittl, Dimitris E. Simos, Edgar Weippl, and Lida Xu (Eds.). Springer, Berlin, 319–328. DOI: http://dx.doi.org/10.1007/978-3-642-40511-2_22

Andreas Holzinger and Igor Jurisica. 2014. The future is in integrative, interactive machine learning solutions. In *Knowledge Discovery and Data Mining in Biomedical Informatics*, Andreas Holzinger and Igor Jurisica (Eds.). Springer, Berlin, Heidelberg, Chapter 1, 1–18. DOI: http://dx.doi.org/10.1007/978-3-662-43968-5_1

Andreas Holzinger, Markus Plass, Katharina Holzinger, Gloria Cerasela Crişan, Camelia-M. Pintea, and Vasile Palade. 2016. Towards interactive machine learning (iML): Applying ant colony algorithms to solve the traveling salesman problem with the human-in-the-loop approach. In *Availability, Reliability, and Security in Information Systems*, Francesco Buccafurri, Andreas Holzinger, Peter Kieseberg, A. Min Tjoa, and Edgar Weippl (Eds.). Springer International Publishing, Cham, 81–95. DOI: http://dx.doi.org/10.1007/978-3-319-45507-5_6

Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'99)*. ACM, New York, NY, 159–166. DOI: http://dx.doi.org/10.1145/302979.303030

Shih-Wen Huang, Pei-Fen Tu, Wai-Tat Fu, and Mohammad Amanzadeh. 2013. Leveraging the crowd to improve feature-sentiment analysis of user reviews. In *Proceedings of the 18th International Conference on Intelligent User Interfaces (IUI'13)*. ACM, New York, NY, 3–14. DOI: http://dx.doi.org/10.1145/2449396.2449400

Shervin Javdani, James Andrew Bagnell, and Siddhartha S. Srinivasa. 2016. Minimizing user cost for shared autonomy. In *Proceedings of the 11th ACM/IEEE International Conference on Human Robot Interaction (HRI'16)*. IEEE Press, Piscataway, NJ, 621–622. http://dl.acm.org/citation.cfm?id=2906831.2907011

Mayank Kabra, Alice A. Robie, Marta Rivera-Alba, Steven Branson, and Kristin Branson. 2013. JAABA: Interactive machine learning for automatic annotation of animal behavior. *Nat. Methods* 10, 1 (2013), 64–67. DOI: http://dx.doi.org/doi:10.1038/nmeth.2281

Ashish Kapoor, Bongshin Lee, Desney Tan, and Eric Horvitz. 2010. Interactive optimization for steering machine classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'10)*. ACM, New York, NY, 1343–1352. DOI: http://dx.doi.org/10.1145/1753326.1753529

Simon Katan, Mick Grierson, and Rebecca Fiebrink. 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 (CHI'15)*. ACM, New York, NY, 251–254. DOI: http://dx.doi.org/10.1145/2702123.2702474

Been Kim, Elena Glassman, Brittney Johnson, and Julie Shah. 2015a. *iBCM: Interactive Bayesian Case Model Empowering Humans via Intuitive Interaction*. Technical Report. MIT Computer Science and Artificial Intelligence Laboratory.

Been Kim, Kayur Patel, Afshin Rostamizadeh, and Julie Shah. 2015b. Scalable and interpretable data representation for high-dimensional, complex data. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*. AAAI Press, 1763–1769. http://dl.acm.org/citation.cfm?id=2886521.2886565.

Andrea Kleinsmith and Marco Gillies. 2013. Customizing by doing for responsive video game characters. *Int. J. Hum.-Comput. Stud.* 71, 7–8 (2013), 775–784. DOI: http://dx.doi.org/10.1016/j.ijhcs.2013.03.005

W. Bradley Knox and Peter Stone. 2015. Framing reinforcement learning from human reward: Reward positivity, temporal discounting, episodicity, and performance. *Artif. Intell.* 225 (2015), 24–50. DOI: http://dx.doi.org/10.1016/j.artint.2015.03.009

Ilker Kose, Mehmet Gokturk, and Kemal Kilic. 2015. An interactive machine-learning-based electronic fraud and abuse detection system in healthcare insurance. *Appl. Soft Comput.* 36 (Nov. 2015), 283–299. DOI: http://dx.doi.org/10.1016/j.asoc.2015.07.018

Nataliya Kosmyna, Franck Tarpin-Bernard, and Bertrand Rivet. 2015. Adding human learning in brain–computer interfaces (BCIs): Towards a practical control modality. *ACM Trans. Comput.-Hum. Interact.* 22, 3, Article 12 (May 2015), 37 pages. DOI: http://dx.doi.org/10.1145/2723162

Anna Kreshuk, Christoph N. Straehle, Christoph Sommer, Ullrich Koethe, Marco Cantoni, Graham Knott, and Fred A. Hamprecht. 2011. Automated detection and segmentation of synaptic contacts in nearly isotropic serial electron microscopy images. *PLoS ONE* 6, 10 (10 2011), 1–8. DOI: http://dx.doi.org/10.1371/journal.pone.0024899

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 SIGCHI Conference on Human Factors in Computing Systems (CHI'14)*. ACM, New York, NY, 3075–3084. DOI: http://dx.doi.org/10.1145/2556288.2557238

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)*. ACM, New York, NY, 126–137. DOI: http://dx.doi.org/10.1145/2678025.2701399

Todd Kulesza, Simone Stumpf, Margaret Burnett, Sherry Yang, Irwin Kwan, and Weng-Keen Wong. 2013. Too much, too little, or just right? Ways explanations impact end users' mental models. In *Proceedings of the 2013 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'13)*. 3–10. DOI: http://dx.doi.org/10.1109/VLHCC.2013.6645235

James Robert Lloyd, David Duvenaud, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. 2014. Automatic construction and natural-language description of nonparametric regression models. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14)*. AAAI Press, 1242–1250.

V. Losing, B. Hammer, and H. Wersing. 2015. Interactive online learning for obstacle classification on a mobile robot. In *Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN'15)*. 1–8. DOI: http://dx.doi.org/10.1109/IJCNN.2015.7280610

Hao Lü, James A. Fogarty, and Yang Li. 2014. Gesture script: Recognizing gestures and their structure using rendering scripts and interactively trained parts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'14)*. ACM, New York, NY, 1685–1694. DOI: http://dx.doi.org/10.1145/2556288.2557263

David Meignan, Sigrid Knust, Jean-Marc Frayret, Gilles Pesant, and Nicolas Gaud. 2015. A review and taxonomy of interactive optimization methods in operations research. *ACM Trans. Interact. Intell. Syst.* 5, 3, Article 17 (Sept. 2015), 43 pages. DOI: http://dx.doi.org/10.1145/2808234

Luana Micallef, Pierre Dragicevic, and Jean-Daniel Fekete. 2012. Assessing the effect of visualizations on Bayesian reasoning through crowdsourcing. *IEEE Trans. Vis. Comput. Graph.* 18, 12 (2012), 2536–2545. DOI: http://dx.doi.org/10.1109/TVCG.2012.199

Bonnie M. Muir and Neville Moray. 1996. Trust in automation. Part II. experimental studies of trust and human intervention in a process control simulation. *Ergonomics* 39, 3 (1996), 429–460. DOI: http://dx.doi.org/10.1080/00140139608964474

Jakob Nielsen. 1995. Heuristic evaluation. In *Usability Inspection Methods*, J. Nielsen and R. L. Mack (Eds.). Wiley & Sons, Chapter 2, 25–62.

Donald A. Norman. 2014. Some observations on mental models. In *Mental Models*, Dedre Gentner and Albert L. Stevens (Eds.). Psychology Press, New York, NY, Chapter 1, 7–14.

Alex T. Pang, Craig M. Wittenbrink, and Suresh K. Lodha. 1997. Approaches to uncertainty visualization. *Vis. Comput.* 13, 8 (1997), 370–390.

Kayur Patel, James Fogarty, James A. Landay, and Beverly Harrison. 2008. Investigating statistical machine learning as a tool for software development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'08)*. ACM, New York, NY, 667–676. DOI: http://dx.doi.org/10.1145/1357054.1357160

Stephen J. Payne and Andrew Howes. 2013. Adaptive interaction: A utility maximization approach to understanding human interaction with technology. *Synth. Lect. Hum.-Centered Inf.* 6, 1 (2013), 1–111. DOI: http://dx.doi.org/10.2200/S00479ED1V01Y201302HCI016

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu

Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* 12 (Nov. 2011), 2825–2830.

Peter Pirolli and Stuart Card. 1995. Information foraging in information access environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'95)*. ACM Press/Addison-Wesley, New York, NY, 51–58. DOI: http://dx.doi.org/10.1145/223904.223911

Reid Porter, Christy Ruggiero, Don Hush, Neal Harvey, Patrick Kelly, Wayne Scoggins, and Lav Tandon. 2011. Interactive image quantification tools in nuclear material forensics. *Proc. Soc. Photo-Opt. Instrum. Eng.* 7877 (2011), 787708-1–787708-9. DOI: http://dx.doi.org/10.1117/12.877319

Reid Porter, James Theiler, and Don Hush. 2013. Interactive machine learning in data exploitation. *Comput. Sci. Eng.* 15, 5 (Sept 2013), 12–20. DOI: http://dx.doi.org/10.1109/MCSE.2013.74

Hema Raghavan, Omid Madani, and Rosie Jones. 2006. Active learning with feedback on features and instances. *J. Mach. Learn. Res.* 7 (2006), 1655–1686.

M. Ribeiro, K. Grolinger, and M. A. M. Capretz. 2015. MLaaS: Machine learning as a service. In *Proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA'15)*. 896–902. DOI: http://dx.doi.org/10.1109/ICMLA.2015.152

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*. ACM, New York, NY, 1135–1144. DOI: http://dx.doi.org/10.1145/2939672.2939778

Alan Ritter and Sumit Basu. 2009. Learning to generalize for complex selection tasks. In *Proceedings of the 14th International Conference on Intelligent User Interfaces (IUI'09)*. ACM, New York, NY, 167–176. DOI: http://dx.doi.org/10.1145/1502650.1502676

Stephanie L. Rosenthal and Anind K. Dey. 2010. Towards maximizing the accuracy of human-labeled sensor data. In *Proceedings of the 15th International Conference on Intelligent User Interfaces (IUI'10)*. ACM, New York, NY, 259–268. DOI: http://dx.doi.org/10.1145/1719970.1720006

Dominik Sacha, Hansi Senaratne, Bum Chul Kwon, Geoffrey Ellis, and Daniel A. Keim. 2016. The role of uncertainty, awareness, and trust in visual analytics. *IEEE Trans. Vis. Comput. Graph.* 22, 1 (2016), 240–249. DOI: http://dx.doi.org/10.1109/TVCG.2015.2467591

Isaias Sanchez-Cortina, Nicolás Serrano, Alberto Sanchis, and Alfons Juan. 2012. A prototype for interactive speech transcription balancing error and supervision effort. In *Proceedings of the 17th International Conference on Intelligent User Interfaces (IUI'12)*. ACM, New York, NY, 325–326. DOI: http://dx.doi.org/10.1145/2166966.2167035

Alvaro Sarasua, Baptiste Caramiaux, and Atau Tanaka. 2016. Machine learning of personal gesture variation in music conducting. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI'16)*. ACM, New York, NY, 3428–3432. DOI: http://dx.doi.org/10.1145/2858036.2858328

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'15)*. 23–36.

Advait Sarkar, Mateja Jamnik, Alan F. Blackwell, and Martin Spott. 2015. Interactive visual machine learning in spreadsheets. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'15)*. 159–163. DOI: http://dx.doi.org/10.1109/VLHCC.2015.7357211

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 Lindley. 2016. Setwise comparison: Consistent, scalable, continuum labels for computer vision. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI'16)*. ACM, New York, NY, 261–271. DOI: http://dx.doi.org/10.1145/2858036.2858199

Jaromır Šavelka, Gaurav Trivedi, and Kevin D. Ashley. 2015. Applying an interactive machine learning approach to statutory analysis. In *Proceedings of the 28th Annual Conference on Legal Knowledge and Information Systems (JURIX'15)*. IOS Press, 101–110. DOI: http://dx.doi.org/10.3233/978-1-61499-609-5-101

Jessica Zeitz Self, Radha Krishnan Vinayagam, J. T. Fry, and Chris North. 2016. Bridging the gap between user intention and model parameters for human-in-the-loop data analytics. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics (HILDA'16)*. ACM, New York, NY, Article 3, 6 pages. DOI: http://dx.doi.org/10.1145/2939502.2939505

Burr Settles. 2010. *Active Learning Literature Survey*. Technical Report 1648. University of Wisconsin-Madison.

Michael Shilman, Desney S. Tan, and Patrice Simard. 2006. CueTIP: A mixed-initiative interface for correcting handwriting errors. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST'06)*. ACM, New York, NY, 323–332. DOI: http://dx.doi.org/10.1145/1166253.1166304

Ben Shneiderman. 1982. The future of interactive systems and the emergence of direct manipulation. *Behav. Inf. Technol.* 1, 3 (1982), 237–256. DOI: http://dx.doi.org/10.1080/01449298208914450

Ben Shneiderman. 1996. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*. IEEE, 336–343. DOI: http://dx.doi.org/10.1109/VL.1996.545307

Charles D. Stolper, Adam Perer, and David Gotz. 2014. Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE Trans. Vis. Comput. Graph.* 20, 12 (Dec. 2014), 1653–1662. DOI: http://dx.doi.org/10.1109/TVCG.2014.2346574

Simone Stumpf, Vidya Rajaram, Lida Li, Weng-Keen Wong, Margaret Burnett, Thomas Dietterich, Erin Sullivan, and Jonathan Herlocker. 2009. Interacting meaningfully with machine learning systems: Three experiments. *Int. J. Hum.-Comput. Stud.* 67, 8 (2009), 639–662. DOI: http://dx.doi.org/10.1016/j.ijhcs.2009.03.004

Qiang Sun and Gerald DeJong. 2005. Explanation-augmented SVM: An approach to incorporating domain knowledge into SVM learning. In *Proceedings of the 22nd International Conference on Machine Learning (ICML'05)*. ACM, New York, NY, 864–871. DOI: http://dx.doi.org/10.1145/1102351.1102460

Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S. Tan. 2009. EnsembleMatrix: Interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'09)*. ACM, New York, NY, 1283–1292. DOI: http://dx.doi.org/10.1145/1518701.1518895

Theophanis Tsandilas, Catherine Letondal, and Wendy E. Mackay. 2009. Musink: Composing music through augmented drawing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'09)*. ACM, New York, NY, 819–828. DOI: http://dx.doi.org/10.1145/1518701.1518827

Fujio Tsutsumi and Yutaka Tateda. 2009. A method to recognize and count leaves on the surface of a river using user's knowledge about color of leaves. In *New Frontiers in Applied Data Mining*, Sanjay Chawla, Takashi Washio, Shin-ichi Minato, Shusaku Tsumoto, Takashi Onoda, Seiji Yamada, and Akihiro Inokuchi (Eds.). Springer, Berlin, 203–212. DOI: http://dx.doi.org/10.1007/978-3-642-00399-8_18

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9 (2008), 2579–2605.

Cristina Versino and Paolo Lombardi. 2011. *Filtering Surveillance Image Streams by Interactive Machine Learning*. Springer, Berlin, 289–325. DOI: http://dx.doi.org/10.1007/978-3-642-19551-8_10

Keith Vertanen and Per Ola Kristensson. 2008. On the benefits of confidence visualization in speech recognition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'08)*. ACM, New York, NY, 1497–1500. DOI: http://dx.doi.org/10.1145/1357054.1357288

Vedrana Vidulin, Marko Bohanec, and Matjaž Gams. 2014. Combining human analysis and machine data mining to obtain credible data relations. *Inf. Sci.* 288 (2014), 254–278. DOI: http://dx.doi.org/10.1016/j.ins.2014.08.014

Byron C. Wallace, Kevin Small, Carla E. Brodley, Joseph Lau, and Thomas A. Trikalinos. 2012. Deploying an interactive machine learning system in an evidence-based practice center: Abstrackr. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium (IHI'12)*. ACM, New York, NY, 819–824. DOI: http://dx.doi.org/10.1145/2110363.2110464

Malcolm Ware, Eibe Frank, Geoffrey Holmes, Mark Hall, and Ian H. Witten. 2001. Interactive machine learning: Letting users build classifiers. *Int. J. Hum.-Comput. Stud.* 55, 3 (2001), 281–292. DOI: http://dx.doi.org/10.1006/ijhc.2001.0499

Weng-Keen Wong, Ian Oberst, Shubhomoy Das, Travis Moore, Simone Stumpf, Kevin McIntosh, and Margaret Burnett. 2011. End-user feature labeling: A locally-weighted regression approach. In *Proceedings of the 16th International Conference on Intelligent User Interfaces (IUI'11)*. ACM, New York, NY, 115–124. DOI: http://dx.doi.org/10.1145/1943403.1943423

Seid Muhie Yimam, Chris Biemann, Ljiljana Majnaric, Šefket Šabanović, and Andreas Holzinger. 2015. Interactive and iterative annotation for biomedical entity recognition. In *Brain Informatics and Health*, Yike Guo, Karl Friston, Faisal Aldo, Sean Hill, and Hanchuan Peng (Eds.). Springer International Publishing, Cham, 347–357. DOI: http://dx.doi.org/10.1007/978-3-319-23344-4_34

Received December 2016; revised December 2017; accepted January 2018