Effective End-User Interaction with Machine Learning

Saleema Amershi[†], **James Fogarty**[†], **Ashish Kapoor**[‡], **Desney Tan**[‡]

[†]Computer Science & Engineering, DUB Group
University of Washington, Seattle, WA 98195
{samershi, jfogarty} @ cs.washington.edu

[‡]Microsoft Research
One Microsoft Way, Redmond, WA, 98052
{akapoor, desney} @ microsoft.com

Abstract

End-user interactive machine learning is a promising tool for enhancing human productivity and capabilities with large unstructured data sets. Recent work has shown that we can create end-user interactive machine learning systems for specific applications. However, we still lack a generalized understanding of *how to design effective end-user interaction with interactive machine learning systems.* This work presents three explorations in designing for effective end-user interaction with machine learning in CueFlik, a system developed to support Web image search. These explorations demonstrate that interactions designed to balance the needs of end-users and machine learning algorithms can significantly improve the effectiveness of end-user interactive machine learning.

Introduction

End-user interactive machine learning is the process by which people define concepts that can be recognized by an intelligent system. These concepts provide the building blocks needed for configuring complex automated behaviors on large data sets. People define concepts by iteratively providing examples of objects matching a desired concept and inspecting feedback presented by the system to illustrate its current understanding (left in Figure 1). Defining concepts via examples enables end-user personalization of intelligent systems while circumventing the interpretation or manipulation of low-level computational representations.

Recent work has demonstrated several applications of end-user interactive machine learning systems. Fails and Olsen's (2003) Crayons system supports interactive training of pixel classifiers for image segmentation in camera-based applications. Dey *et al*.'s (2004) a CAPpella enables end-user training of a machine learning system for context detection in sensor-equipped environments. Ritter and Basu (2009) demonstrate interactive machine learning in complex file selection tasks. Each of these provides initial evidence of the utility of interactive machine learning, but we still lack a generalized understanding of *how to design effective end-user interaction with interactive machine learning systems*. For instance, which examples should a person provide to efficiently train the system? How should the system illustrate its current understanding? How can a person evaluate the quality of the system's current understanding in order to better guide it towards the desired behavior?

A traditional active learning approach to interaction can meet the needs of the machine learning system by forcing a person to label training examples that provide the greatest information gain. However, treating a person like a passive information oracle can create a frustrating user experience (Baum and Lang 1992). On the other hand, a design that neglects the learning system in favor of end-user flexibility may be equally frustrating if a person cannot effectively train the system. Effective solutions must therefore balance the needs of both the end-user and the machine.

This paper presents three explorations of designing effective end-user interaction with machine learning in CueFlik, a system we developed to support Web image search (Fogarty et al. 2008). Our results show that well designed interactions can significantly impact the effectiveness of the interactive machine learning process. In addition, while our explorations are grounded in CueFlik, we intentionally designed our methods to be independent of CueFlik, image-specific features, and image search. As a result, our findings should generalize to other domains suitable for example-based training.

CueFlik

CueFlik (Figure 1) allows end-users to interactively define visual concepts (e.g., "*product photos*", "*pictures with quiet scenery*", "*pictures with bright psychedelic colors*") for re-ranking web image search results. End-users train CueFlik by providing examples of images with and without the desired characteristics. These examples are used to learn a distance metric as a weighted sum of component distance metrics (including histograms of pixel hue, saturation, luminosity, edges, global shape and texture). Formally, CueFlik minimizes an objective function

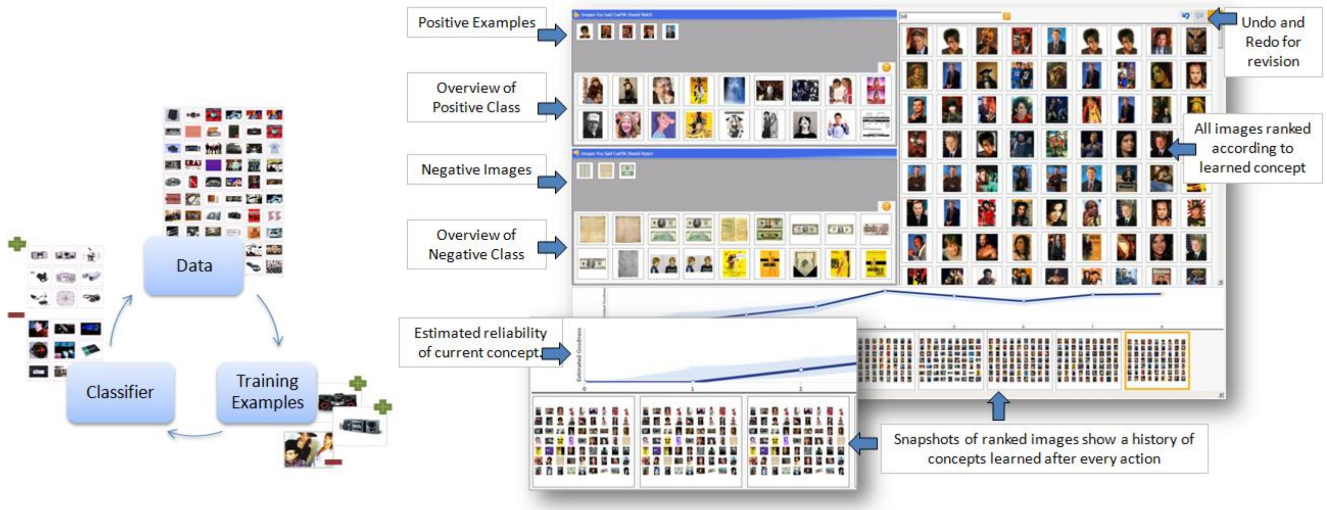
Data
Training Examples
Classifier
Positive Examples
Overview of Positive Class
Negative Images
Overview of Negative Class
Estimated reliability of current concept
Undo and Redo for revision
All images ranked according to learned concept
Snapshots of ranked images show a history of concepts learned after every action

Figure 1. In end-user interactive machine learning (left), a person iteratively provides a system with training examples for a desired concept. Those examples train a classifier that is applied to the remaining data. A person then inspects presented data and decides how to best proceed with refining the classifier. CueFlik (right) supports end-user interactive machine learning in Web image search.

separating positive examples from negative examples while keeping examples in the same class close together:

$$f(\mathit{weights}) = \sum_{i,j\,\in\,Pos} D(i,j) + \sum_{i,j\,\in\,Neg} D(i,j) + \sum_{i\,\in\,All} \ln \sum_{j\,\in\,All} e^{-D(i,j)}$$

where *D(i, j)* is the distance metric computed as a weighted sum of CueFlik's component metrics. The first two terms correspond to within-class distances. Minimizing the function therefore favors weights that collapse the positive and negative classes. The third term considers all examples, thus favoring maximum separation of classes.

CueFlik uses provided examples to update its distance metric. It then uses a nearest-neighbor classifier to re-rank images according to their likelihood of membership in the positive class. Throughout the iterative training process, CueFlik presents examples illustrating its current understanding of the desired concept and end-users decide how to proceed with improving system understanding.

Designing Effective Interactions with CueFlik

There are many possible ways to design the various interactions during the end-user interactive machine learning process. In our explorations with CueFlik, we attempt to move beyond previous naïve or ad-hoc approaches by designing general techniques that balance the needs of both end-users and machine learning algorithms. The techniques we present here target three important aspects of the end-user interactive machine learning process: (1) effectively illustrating the current version of a learned concept (Fogarty et al. 2008), (2) guiding end-users to select training examples that result in higher quality concepts (Amershi et al. 2009), and (3) enabling effective and lightweight end-user exploration of multiple potential models (Amershi et al. 2010).

We present the results of our interaction evaluations in terms of two key measures: quality of end-user-trained concepts and their efficiency in training. For study details, please refer to our original publications.

Illustrating the Current Learned Concept

A fundamental issue in end-user interactive machine learning is illustrating the system's current understanding of a learned concept. An effective illustration can help people asses the quality of the current concept and in turn inform whether and how they proceed in training.

We examined two methods for illustrating CueFlik's current version of a learned concept: *single* versus *split* presentation (Fogarty et al. 2008). The *single* method provides access to the entire set of images, ranked by their likelihood of membership in the positive class (right in CueFlik interface, Figure 1). The *split* method instead shows only the best and worst matching images in the set (left in CueFlick interface, Figure 1). The best matches show a small number of high-certainty positive images (extremely close to positive training examples). The worst matches show a small number of high-certainty negative images (extremely close to negative training examples).

In addition, we experimented with integrating active learning examples into both the *single* and *split* presentations interfaces. These examples were chosen using standard active learning heuristics for selecting examples that provide the system with the most information gain (e.g., examples the system is currently most uncertain about, such as examples near the boundary of the positive and negative classes).

From our evaluation, we found that participants using the *split* presentation created CueFlik concepts of significantly higher quality, using significantly fewer training examples, in significantly less time than participants using the *single* method. One explanation of this is that the *split* presentation encouraged participants to focus on whether the system's understanding was mostly correct (i.e., whether the best and worst matches corresponded to their desired concept). In contrast, presenting the entire set of images (*single*) exposes participants to images for which the system is more uncertain (e.g., images in the middle of the ordered set). These images may have led participants to find relatively minor inconsistencies, prompting them to continue adding examples and take more time. Furthermore, as people label more of these uncertain images, CueFlik may begin to learn irrelevant aspects of those examples. Interestingly, neither the presence nor absence of active learning examples (i.e., examples that are theoretically intended to provide the machine with the most information about the model being trained) had a significant effect on participant ability to train models. These findings suggest further exploration of how to best guide people to select effective training examples during interactive machine learning.

Soliciting Effective Training Examples

Our initial exploration showed that the *split* method of presenting examples led participants to train better concepts. This result mixes two possible explanations for the improvement: (1) the use of a split presentation with a small number of examples illustrating the positive and negative regions during interactive refinement of a learned concept, and (2) that those examples were selected as representative of the positive and negative regions because of their *high-certainty*. We hypothesized that the first of these explanations is indeed important. However, because *high-certainty* examples are extremely similar to already labeled examples, they provide little additional information to the machine learning algorithm during training.

We examined two strategies for selecting small sets of examples of high-value to the machine learning algorithm that also provide the end-user with an intuitive overview of the positive and negative regions of a space defined by a learned concept (Amershi et al. 2009). Our first strategy presents a *global* overview, selecting examples to provide good coverage of the positive and negative regions (left in CueFlik interface, Figure 1). We use a sensor-placement strategy (Krause et al. 2008) to select examples, *i*, that maximize the mutual information gain between currently selected, *S*, and unselected, *U*, examples (and are therefore of high quality from the learner's perspective):

$$MI(U-i;S\cup i)-MI(U-i;S)$$

To achieve this, we take a Gaussian Process perspective and select examples that maximize:

$$f(i)=\frac{1-K_{i,S}K_{S,S}^{-1}K_{S,i}}{1-K_{i,U-i}K_{U-i,U-i}^{-1}K_{U-i,i}}$$

where $K_{S,S}$ is the similarity matrix among S, $K_{U\text{-}i,U\text{-}i}$ is the similarity matrix among U excluding i, and $K_{i,S}$, $K_{S,i}$, $K_{i,U\text{-}i}$, and $K_{U\text{-}i,i}$ are each similarity vectors between i and the respective sets (Amershi et al. 2009). Intuitively, examples maximizing this ratio are most dissimilar to selected examples and most representative of those unselected.

Our second strategy emphasizes *projected* overviews, selecting instances that illustrate variation along major dimensions of the positive and negative regions. We first obtain a set of principle dimensions in each region and then select examples along each. We use a non-linear projection technique similar to Principal Component Analysis to compute principle dimensions, as this best respects the structure of the underlying data (Amershi et al. 2009). To select instances that best illustrate the intended variation (i.e., provide coverage of a single principal dimension but also vary as little as possible in all other dimensions), we modify our sensor placement strategy to maximize:

$$f(i)=\left(\frac{1-K_{i,S}K_{S,S}^{-1}K_{S,i}}{1-K_{i,U-i}K_{U-i,U-i}^{-1}K_{U-i,i}}\right)\left(\frac{1}{1-\overline{K}_{i,S}\overline{K}_{SS}^{-1}\overline{K}_{S,i}}\right)$$

where K is the similarity matrix for the principal dimension for which we are currently selecting a set of representative examples and $\overline{K}$ is the similarity matrix for all of the other principal dimensions (Amershi et al. 2009).

We compared our new overview-based strategies to the best performing strategy from our initial work (i.e., the *high-certainty* strategy presenting the best and worst matches). We found that our overview-based strategies of presenting high-value examples guided participants to select better training examples and train significantly higher quality concepts than the *high-certainty* strategy. However, we also found that participants spent more time training when using the overview-based strategies.

During our evaluation, we observed that participants often continued providing additional training examples even when they did not seem to be further improving a concept. This obviously increases the training time and we believed it could also negatively impact final concept quality. We therefore further analyzed the point where participants obtained their *best* learned concept. This showed that our overviews led participants to train better *best* concepts in the same amount of time and with fewer examples than the *high-certainty* strategy. This analysis also showed that all of our interfaces suffered from some model decay (from best to final concepts). Participant feedback indicated they were often unable to revert back to previous model states during training when quality started to decay (e.g., "*it was weird, sometimes it would start out doing really well, but as I kept going it did worse*"). Our overviews, however, helped to reduce the magnitude of this decay compared to the *high-certainty* condition.

Examining Multiple Potential Models

Participants in our second exploration were unable to revert back to previous models when they observed that CueFlik was not behaving in the desired manner. We hypothesized that this was partly due to an implicit assumption in prior research about how people should interact with machine learning. Machine learning systems learn by generalizing from examples of object classes. Prior research has thus focused interaction on prompting a person to answer "*what class is this object?*" (e.g., Tong and Chang 2001). Such an approach permits simulated experiments with fully-labeled datasets. However, treating a person simply as an oracle neglects human ability to revise and experiment. We therefore propose that a person instead consider "*how will different labels for these objects impact the system in relation to my goals?*"

Our third exploration examines the impact of end-user comparison of multiple potential models during the interactive machine learning process (Amershi et al. 2010). Comparison of multiple alternatives is a proven technique in human-computer interaction but has not been explored in the context of people interacting with machine learning. We examine this with a history visualization showing recently explored models and support for revision (see CueFlik interface in Figure 1). The history contains a plot of each model's estimated reliability, updated after every end-user interaction (e.g., labeling examples). Model reliability is measured using leave-one-out-cross-validation on the current set of training examples. The history also shows snapshots of each model's top ranked images for visual comparison. Revision can be achieved by removing examples directly, via undo/redo, and by clicking directly within the history to revert back to previous models.

Our evaluation showed that the history visualization led participants to spend more time and perform more actions to train concepts without improving overall model quality. Although the plot used an accepted metric to estimate model reliability (leave-one-out-cross-validation accuracy), end-users seemed to use it less like an tool for helping them interpret model quality and more like a quantity to maximize (e.g., "*I wanted the graph to go up instead of concentrating on [the results]*"). This emphasizes the need to consider a person's understanding of the limitations (and benefits) of accepted machine learning techniques when designing interactive machine learning systems.

Our evaluation also found that participants readily adopted revision mechanisms, making use of them in 68% of their tasks when it was available. Revision also led participants to achieve better quality final models in the same amount of time than when revision was not available. Furthermore, examining and revising actions is consistent with how people expect to interact with applications. One participant commented that without revision "*it felt a little like typing on a keyboard without a backspace key*".

While revision led our participants to create better quality final models, we still observed some decay in all conditions. This problem of helping people determine appropriate stopping points is related to the machine learning problem of identifying overfitting. Therefore, a perspective that considers both the human and the machine introduces new opportunities for solving these and other open problems in interactive machine learning.

Conclusion

In this work we explore *how to design effective end-user interaction with interactive machine learning systems*. While important problems remain, our explorations with CueFlik demonstrate that careful designs considering the needs of both end-users and machine learning algorithms can significantly impact the effectiveness of end-user interaction. Moreover, many of our techniques are not specific to image search or features of images. Techniques like overview-based example selection or revision of previous models can therefore potentially impact a wide variety of machine learning based applications.

References

Amershi, S., Fogarty, J., Kapoor, A. and Tan, D. 2010. Examining Multiple Potential Models in End-User Interactive Concept Learning. In *Proceedings of CHI 2010*, 1357-1360. New York, NY: ACM Press.

Amershi, S., Fogarty, J., Kapoor, A. and Tan, D. 2009. Overview-Based Examples Selection in Mixed-Initiative Interactive Concept Learning. In *Proceedings of UIST 2009*, 247-256. New York, NY: ACM Press.

Baum, E.B. and Lang, K. 1992. Query Learning can work Poorly when a Human Oracle is Used. In *Proceedings of IJCNN 1992*, 609-614. Washington, DC: IEEE Computer Society.

Dey, A.K., Hamid, R., Beckmann, C., Li, I. and Hsu, D. 2004. a CAPpella: Programming by Demonstrations of Context-Aware Applications. In *Proceedings of CHI 2004*, 33-40. New York, NY: ACM Press.

Fails, J.A., Olsen Jr., D.R. 2003. Interactive Machine Learning. In *Proceedings of IUI 2003*, 39-45. New York, NY: ACM Press.

Fogarty, J., Tan. D., Kapoor, A. and Winder, S. 2008. CueFlik: Interactive Concept Learning in Image Search. In *Proceedings of CHI 2008*, 29-38. New York, NY: ACM Press.

Krause, A., Singh, A. and Guestrin, C. 2008. Near-optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *Journal of Machine Learning Research* 9 : 235-284.

Ritter, A. and Basu, S. 2009. Learning to Generalize for Complex Selection Tasks. In *Proceedings of IUI 2009*, 167-176. New York, NY: ACM Press.

Tong, S. and Chang, E. 2001. Support Vector Machine Active Learning for Image Retrieval. In *Proceedings of Multimedia 2001*, 107-118. New York, NY: ACM Press.