Accounting for Input Uncertainty in Human-in-the-Loop Systems

Alison Smith
University of Maryland
College Park, MD, USA
amsmit@cs.umd.edu

Varun Kumar
University of Maryland
College Park, MD, USA
varunk@cs.umd.edu

Jordan Boyd-Graber
University of Colorado, Boulder
Boulder, CO, USA
jbg@boydgraber.org

Kevin Seppi
Brigham Young University
Provo, UT, USA
kseppi@byu.edu

Leah Findlater
University of Maryland
College Park, MD, USA
leahkf@umd.edu

Abstract
Human-in-the-loop (HITL) machine learning is necessary to personalize recommendations, maintain high classification accuracy, and imbue systems with domain knowledge. There has been substantial research on how to present system uncertainty to a user (i.e., output), but what about when user *input* includes uncertainty? In this position paper, we discuss a variety of HITL systems and describe how and when user input can be uncertain. We then argue why input uncertainty is a problem and suggest design considerations for HITL systems that deal with uncertain input. These design considerations include ensuring that input mechanisms match user expectation and allow for uncertainty and providing clarifying explanations in the case of user/algorithm mismatch.

Author Keywords
Interactive machine learning; human-in-the-loop machine learning; input uncertainty; interactive topic modeling

ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): User-centered design.

#	Topic Words
1	hold usairways americanair call back phone hours wait change minutes online hour number line system called reservation times answer put
2	flight usairways delayed hrs hours late miss made delay connection flt missed stuck flights home missing ridiculous dca phl day
3	service customer united usairways worst airline experience agents staff flying rude americanair terrible disappointed bad poor great horrible customers worse
4	united bag bags luggage lost baggage check find airport time checked lax claim stop ewr told found denver destination hard
5	united plane gate waiting hour seat sitting crew delay min agent left http mins leave board amp minutes tarmac ord

Table 1: Five of the 10 topics of a model generated from a corpus of tweets with complaints directed to commercial airlines. These topics, displayed as ordered lists of words, exemplify some common complaints, such as long phone hold times, plane delays, and lost bags.

Introduction

Machine learning algorithms and the models that they produce are typically too complex for a non-expert end user to adjust or manipulate; however, there are applications where explicit user input is beneficial or even required. These are known as human-in-the-loop (HITL) systems, and include recommendation systems, which require user preferences, active learning systems where users provide training labels, and interactive machine learning algorithms where users provide domain expertise to update models at run time.

A primary focus in HITL machine learning, and machine learning in general, has been on presenting system uncertainty to a user. When the user must make a decision based on the algorithm's recommendation, for example, presenting uncertainty can foster user trust [1]. User input, however, can also be a source of uncertainty. For example, if a user feels lukewarm about an item but a recommendation system provides only the binary options "like" or "dislike", the explicit preference is a weak approximation of their true feeling. Alternatively, user input for active learning is uncertain if they are not sure about the correct label.

Input uncertainty, if not accounted for, can lead to user frustration, downstream model issues, or decreased user feedback. Just as HITL systems are treated as a collaboration between people and algorithms, the expression of uncertainty should be considered from both sides. Designs should account for uncertainty for both the system and the user. In this position paper, we present cases of three HITL systems where uncertain input may be an issue. We also describe why input uncertainty is worth examining more generally and outline some initial design considerations for input uncertainty in HITL systems that should be explored in future work.

Case 1: Interactive Machine Learning

In interactive machine learning [2], users and algorithms work together with the goal of improving a model. A specific example is Interactive Topic Modeling (ITM) [3], which is an extension of topic modeling [4], a common data-driven technique that discovers abstract topics that occur together in a corpus. A topic model is typically presented as a list of topics, each displayed as a list of its top words. Table 1 shows a sample of a topic model for a corpus of negative tweets directed at commercial airlines[1]. ITM provides a mechanism for users to refine topic models, such as by changing the order of words in a topic, adding a word to a stop words list (meaning that it will be ignored during modeling), and adding or removing topic words.

In ongoing work, we asked 12 participants to refine a topic model built from the airline tweet corpus and to comment about predictability issues during that process, such as refinements not being applied as directed or other unexpected changes occurring. Many issues arose. For example, when users change the order of words within topics it is not guaranteed that the order in the updated model will be exactly as specified. Alternatively, when users remove words from topics, the words could unexpectedly end up in other topics. We found that users had varied responses in these cases of unpredictability; some users were frustrated, saying, for example, "I tried to move this word and it just goes back up", while others trusted the system, saying, "Oh, it must know better than me".

[1] https://www.kaggle.com/crowdflower/twitter-airline-sentiment

While we want to reduce user frustration as it can lead to system disuse, over trust is also a concern, as it could mean that the user will provide less input or not think critically about the results given by the algorithm.

The level of unpredictability varies between different refinements, as ITM treats refinements with varied levels of input uncertainty, without knowing the actual intent of the user. Adding a word to the stop words list (i.e., removing it from the model), for example, is entirely predictable—it is treated as a *command* and implemented with no input uncertainty. Changing the order of words in a topic, however, is treated only as a *suggestion* by the user, and was described by some participants as the least predictable of the six refinements provided in our interface.

Input uncertainty also arises when user assumptions are inconsistent with either the algorithm or the data—a situation that users may suspect or of which they may be completely unaware. For example, if users specify refinements that do not align with the backing data, such as adding words to topics that do not exist in the full vocabulary, the system will be even more unlikely to incorporate the refinements as expected.

Without knowing the user's input uncertainty, it is impossible for the algorithm to appropriately match it. For cases where the system does "know better", such as when the user specifies a word to add that is not in the vocabulary, the system should communicate that confidence back to the user as an explanation. A complementary approach is to design and evaluate mechanisms for the user to specify with what level of certainty their input should be applied—from a deterministic *command* to a weak *suggestion*. These system-led and user-led mechanisms can also work together; for example, an explanation for the system's confidence in the user's input can in turn cause the user to adjust their own specified uncertainty.

Case 2: Recommendation Systems

Recommendation systems [5] require input in the form of preferences to build user profiles and make recommendations. The collection mechanism can create uncertainty when preferences are explicitly collected from the user. The personalized radio service Pandora, for example, allows a simple "thumbs up" or "thumbs down" option for whether a song is liked, introducing uncertainty if the user's preference does not fit a binary scale. Complicating the issue of input uncertainty, there is wide variability in how preferences may be explicitly collected, from open-ended questions to ratings on Likert scales to hybrid techniques [6]. There is variance even in rating scale—compared to Pandora, for example, Netflix supports movie ratings on a five-point scale, which is more complex but also more precise.

Future work should explore what level of uncertainty, if any, to attach to user input. For example, should the "thumbs down" option in Pandora be taken as a strong *suggestion* rather than direct *command*? User studies are needed to measure the tradeoff between the benefit of direct control of input uncertainty and the negative effects of added interaction complexity.

Case 3: Active Learning

Any supervised machine learning algorithm is only as good as its human-labeled training data. A requisite amount of training data can be costly if not impossible to obtain for a given application. An HITL approach to this problem is called active learning, where the

algorithm queries the user for the desired outputs of new data points instead of requiring true labels for all instances [7]. Typically, the instances shown to the user are those for which the learner is most uncertain or are representative instances for which a label will provide the most benefit to the algorithm [8].

Although prior work has explored whether the learner should display its uncertainty about an instance to a user [9], the label provided by the user may have a degree of uncertainty as well. User uncertainty could arise, for example, if the user does not have global knowledge of the corpus or if the example is particularly hard to label – an issue that is not unlikely as the algorithm selects instances for the user to label for which it is especially uncertain. Although the accuracy of these systems is typically robust to a few incorrect training examples, an open research question is to analyze whether allowing a user to specify input uncertainty would lead to increased classification accuracy. Additionally, this may minimize the frustration that results from a user being uncertain about a label but unable to specify that to the system.

Conclusion

In this paper, we discussed three cases of HITL systems where uncertain input may be an issue. Here we outline two primary design considerations for accounting for input uncertainty in HTIL systems that should be explored in future work. First, input mechanisms should allow for uncertainty. How user input is handled – from direct *commands* to weak *suggestion* – currently varies across HITL systems. How to best allow a user to specify uncertainty of their input, whether a topic refinement, training label or a song preference, and how to best account for this uncertainty in the algorithms are open research questions. Second, explanations should be provided in the case of user and algorithm mismatch. Explanations of cases where there is discrepancy between user input and the algorithm help the user to provide more certain input going forward and determine whether to adjust how the input is applied in the algorithm.

References

[1] A. Bussone, S. Stumpf, and D. O'Sullivan, "The role of explanations on trust and reliance in clinical decision support systems," in *Proceedings - 2015 IEEE Int. Conf. on Healthcare Informatics, ICHI 2015*, 2015, pp. 160–169.

[2] J. A. Fails and D. R. Olsen, "Interactive machine learning," *Proc. 8th Int. Conf. Intell. User interfaces IUI 03*, pp. 39–45, 2003.

[3] Y. Hu, J. Boyd-Graber, B. Satinoff, and A. Smith, "Interactive Topic Modeling," *Mach. Learn.*, vol. 95, no. 3, pp. 423–469, Jun. 2014.

[4] D. Blei, L. Carin, and D. Dunson, "Probabilistic topic models," *IEEE Signal Process. Mag.*, vol. 27, no. 6, pp. 55–65, 2010.

[5] M. J. Pazzani and D. Billsus, "Content-Based Recommendation Systems," *Adapt. Web*, vol. 4321, pp. 325–341, 2007.

[6] K. Swearingen and R. Sinha, "Interaction design for recommender systems," *Des. Interact. Syst.*, pp. 1–10, 2002.

[7] B. Settles, "Active Learning Literature Survey," *Mach. Learn.*, vol. 15, no. 2, pp. 201–221, 2010.

[8] Y. Fu, X. Zhu, and B. Li, "A survey on instance selection for active learning," *Knowl. Inf. Syst.*, vol. 35, no. 2, pp. 249–283, 2013.

[9] S. L. Rosenthal and A. K. Dey, "Towards maximizing the accuracy of human-labeled sensor data," *Int. Conf. Intell. User Interfaces, Proc. IUI*, pp. 259–268, 2010.