# Unifying recommendation and active learning for human-algorithm interactions

**Scott Cheng-Hsin Yang**[1] **(scott.cheng.hsin.yang@gmail.com), Jake Alden Whritner**[1] **(jake.whritner@rutgers.edu), Olfa Nasraoui**[2] **(olfa.nasraoui@gmail.com) & Patrick Shafto**[1] **(patrick.shafto@gmail.com)**

[1] Department of Mathematics & Computer Science, Rutgers University–Newark

[2] Department of Computer Engineering and Computer Science, University of Louisville

## Abstract

The enormous scale of the available information and products on the Internet has necessitated the development of algorithms that intermediate between options and human users. These algorithms do not select information at random, but attempt to provide the user with relevant information. In doing so, the algorithms may incur potential negative consequences related to, for example, "filter bubbles." Building from existing algorithms, we introduce a parametrized model that unifies and interpolates between recommending relevant information and active learning. In a concept learning paradigm, we illustrate the trade-offs of optimizing prediction and recommendation, show that there is a broad parameter region of stable performance that optimizes for both, identify a specific regime that is most robust to human variability, and identify the cause of this optimized performance. We conclude by discussing implications for the cognitive science of concept learning and the practice of machine learning in the real world.

**Keywords:** Recommender systems, active learning, concept learning, filter bubble

Historically, the information each individual had access to was defined by one's local environment: what one could directly observe or purchase, who one had to talk to or do business with, and available texts or catalogs one could access. With the advent of the Internet, information and products became available at a global scale. This vast potential resource creates a problem: how to choose—from billions or trillions of options—which information or products to present to an individual at a given time. Solutions to these problems form the foundation that supports major players in the online business world—from search engines and e-commerce to social network services—such as Google, Amazon, and Facebook. These algorithmic solutions radically affect not only what information and products we are exposed to, but also which information and products we have the *chance* to be exposed to. Thus, these algorithms mediate between us and reality, not by providing a random sample from what is possible, but by carefully selecting a sample which optimizes for the underlying goals and metrics of the company. The consequences of these human-algorithm interactions have been insufficiently explored despite recent interest in cases such as filter bubbles (Pariser, 2011) and algorithmic bias (Baeza-Yates, 2016).

A well-established doctrine in cognitive science asserts that a driving factor of our beliefs is the information we are exposed to. However, the situations investigated in the most typical concept learning experiments (Bruner, Goodnow, & Austin, 1956; Shepard, Hovland, & Jenkins, 1961) differ sharply from the kinds of situations we encounter with recommender systems. In concept learning experiments, examples are typically sampled either exhaustively or randomly, neither of which is feasible in the context of Internet-scale problems. Obviously, enumeration is not feasible. Random sampling is also not feasible because if the quality of the algorithm's suggestions were too poor, human users could simply choose to go elsewhere. This yields a thorny problem: how to select information and products to maximize relevance, while also accurately estimating what users want.

Two classes of algorithms—information filters (Sparck Jones, 1970; Van Rijsbergen, 1979; Salton, Fox, & Wu, 1983) and recommender systems (Goldberg, Nichols, Oki, & Terry, 1992; Maes et al., 1994; Adomavicius & Tuzhilin, 2005)—have been developed to facilitate selection of information for users. Although different in some ways, they share a core assumption that the goal is to deliver humans relevant information or products. Given that these sorts of algorithms have raised concerns about not exposing people to the breadth of potentially relevant information, basic questions one might raise are whether the data they obtain allow them to accurately estimate human users' preferences, and whether there are small adjustments that could be made to optimize for recommendation and learning about the users' preferences.

One approach for obtaining optimally informative data is active learning, well known in both cognitive science (Nelson, 2005) and computer science literatures (MacKay, 1992). Active learning has been proposed both as a model for how humans search for information and as an algorithm for how machines might learn about the world. In the current context, active learning is a method for learning what information is relevant to the human user (Elahi, Ricci, & Rubens, 2016), which—while having advantages for estimating the user's beliefs—is unlikely to produce quality recommendations.

Drawing inspiration from real-world problems of information filtering and recommendation, we seat these problems in a concept learning framework that allows for experimental control of which examples are relevant. This framework allows for an exploration of algorithms that perform better or worse on the joint problems of optimizing recommendations and inferring relevance. We introduce a novel approach to investigating algorithm performance that merges aspects of computational simulation and user testing: people are trained on the true concept, then they interact with the algorithm to test performance. Unlike in computational simulations and user testing, there is both defined ground truth and naturalistic human variability in behavior.

Our approach uses the simple concept learning task previously used by Markant and Gureckis (2014). We present two experiments. The first validates the method by demonstrating expected limitations of recommendation and active learning. The second investigates a simple one-parameter generalization—active recommendation—that unifies both approaches. We show that while the extreme cases of pure active learning and pure recommendation yield poor performance, all intermediate values converge to near optimal recommendation and prediction. We also observe that due to human variability, parameter values that are closer to pure recommendation yield the best performance. We conclude by discussing implications for cognitive science in the lab and machine learning in the real world. Overall, the main contributions of our work are: (i) the use of an experiment to gauge how real human users interact with a system that spans graded shades between recommendation and active learning and (ii) how a unified, yet simple and generic model is beneficial in the design and interpreting of real user experiments.

## Unifying recommending and active learning

Given a dataset, $D = \{x_i, y_i\}_i^N$, the goal of a probabilistic classification algorithm is to predict the probability that a new data point $x^*$ belongs to class $y$, $P(y|x^*, D)$. We will be concerned with learning two classes corresponding to irrelevant and relevant information, $y \in \{0, 1\}$. These predictions form the basis of the recommendation and active learning algorithms we will consider. Intuitively, the goal of recommendation is to provide the user with examples that are relevant. This intuition can be formalized directly,

$$x_{rec} = \arg\max_{x^*} P(y = 1|x^*, D). \tag{1}$$

At any point—given previously observed data—this defines which examples are optimal for recommendation: those that maximize the probability of being relevant. Within the closely related problem of probabilistic retrieval (ranking relevant information), this coincides with optimal probabilistic retrieval for the most relevant item (Robertson, 1977).

One intuitive formalization of active learning is to select examples that reduce our uncertainty about which examples are relevant. This can also be formalized directly,

$$x_{act} = \arg\min_{x^*} |0.5 - P(y = 1|x^*, D)|. \tag{2}$$

Given previously observed data, the optimal example to observe is the one about which we have the greatest predictive uncertainty.

It is worth noting that this is not the only formalization of active learning that one may consider. Other well known strategies include optimizing information gain (K-L divergence), diagnosticity, and probability gain (Nelson, 2005). While each differs in formal detail, in many practically relevant situations, their predictions are quite similar. We formalize active learning as the selection of maximally uncertain data to facilitate integration with the recommendation criterion in Eq. 1, as will be seen below.

We propose a unified model of *active recommendation* that exploits the parallel structure in previous models. Our single parameter generalization includes filtering and active learning as extreme cases and thus unifies the two approaches and interpolates between them. Formally,

$$x_\alpha = \arg\min_{x^*} |\alpha - P(y = 1|x^*, D)|, \tag{3}$$

where $\alpha \in [0.5, 1]$. When $\alpha = 0.5$ we recover active learning, as is obvious from inspection of Eq. 2. When $\alpha = 1$, we recover recommendation. In our context, subtracting from 1 and taking the min is equivalent to taking the max in Eq. 1.

Of interest is what happens between the extremes that correspond to recommendation and active learning. Are there parameterizations of active recommendation that optimize accuracy in terms of recommendation and prediction? Are there parameterizations that are more robust to the kinds of variability that are characteristic of human behavior?

## Experiments

In what follows, we empirically investigate these questions using a novel approach. Human subjects were first trained on the underlying conceptual structure that defines which examples are relevant and which are not. The classes of relevant examples are defined by axis-aligned logistic function with data standardization in two dimensions. Next, people are randomly assigned to an algorithm, and are presented with a series of examples, which they label as being in the relevant class or not. The algorithm updates upon receiving each example-label pair, and then selects a new example. This method combines aspects of computational simulation and user testing by providing a ground truth, yet allowing human variability in responses. It thus provides information about when we would expect algorithms to perform well—both absolutely and in the presence of human variability.

The questions of interest are: which algorithms perform well in terms of recommendation and prediction and which ones perform well in terms of robustness to human variability? An algorithm's trial-by-trial recommendation accuracy is the fraction of examples labeled as relevant, at each trial, by a population of participants. Its predictive accuracy for trial $i$ is the fraction of correct predictions—made by the classification algorithm trained with data up to trial $i$—where predictions are tested on a grid of predetermined, held-out test examples. The correctness is judged against the optimal decision boundary that was set in the beginning of the experiment.

Two experiments follow. Experiment 1 investigates the performance of pure recommendation and active learning, and compares them with random sampling. This experiment allows us to validate that recommendation and active learning fail to predict and recommend well, respectively, and provides a random sampling baseline. Experiment 2 investigates the unified active recommendation model, which interpolates between pure recommendation and pure active learning. This experiment characterizes recommendation and prediction accuracies of the algorithm in the context of concept learning.

### Experiment 1

**Participants.** The experiment was run on Amazon's Mechanical Turk (MTurk) with 30 participants in each of the three conditions: recommend, active learning, and random.

**Stimuli.** Following Markant and Gureckis (2014), the stimuli were circles with a central diameter. The stimuli varied along two dimensions—the size of the circle's radius in pixels and the orientation of the central diameter in degrees (for an example, see Figure 5 B in Markant and Gureckis (2014)). The ranges of the size and orientation were fixed to 110 pixels and 140 degrees, respectively. The minimum radius and minimum orientation for the classes were sampled independently and uniformly from 10 to 30 units and fixed for the whole experiment. This procedure determined a pair of minimum and maximum values $\{min, max\}$ for each dimension.

For each experiment, one of the dimensions (size or orientation) was randomly selected as the separable dimension. Let the $\{min_s, max_s\}$ be the minimum and maximum values of the separable dimension, and $\{min_t, max_t\}$ be the values for the other dimension. Two classes were defined by two two-dimensional normal distributions. Along the separable dimension, the variances of the two classes were both 75, and their means were set at $(min_s + max_s)/2 \pm 30$. Along the other dimension, the variances were both 2250, and the means were both $(min_t + max_t)/2$. Stimuli were sampled from the two-dimensional Gaussian described above. Those that happened to be outside the determined range were resampled.

The experiment consisted of three phases: training, interaction, and testing. In each trial of the training phase, a class was randomly sampled, and a stimulus was sampled according to its class distribution. In the interaction phase, there were several sampling algorithms. Random sampling used the procedure as in the training phase. Recommendation and active-learning sampling followed Eqs. 1 and 2 respectively. The choice was made from a fixed pool of 400 randomly sampled stimuli for each experiment. In the test phase, the stimuli were no longer sampled from the classes but from a test set. The test set consisted of $16 \times 16$ samples that lied on a regular grid covering the area of feature space defined by [10,140] pixels × [10,170] degrees. Five stimuli were randomly selected from each of the four quadrants in that area to form the 20 test stimuli used in the test phase.

**Procedures.** Before the training phase, participants were instructed that throughout the experiment, they would see a series of "loop antennas" that receive signals from music stations called "Beat" and "Sonic" (the two classes of stimuli described above). They were instructed that the station received depends upon the antenna's radius and the orientation of its diameter. The goal of the training phase, as described to the participants, was for them to learn which station was received by a given class of antennas (e.g., Beat antennas have large diameters and Sonic have small). Participants provided input by clicking on one of two buttons (labeled Beat and Sonic respectively). After responding, participants received feedback on whether or not their input was correct. The participants moved on to the interaction phase once they had 19 correct answers in the past 20 trials.

The interaction phase was comprised of two parts. Participants were first instructed to pretend that they preferred either Beat or Sonic. Given this preference, participants were told that they would teach an algorithm to recommend the station that they preferred by indicating–by clicking on a button–that the antenna it chose was one that they "like" or "dislike." Participants were instructed to pay attention to whether the algorithm was improving or not. This part of the interaction phase continued for 20 trials. Next, participants rated the algorithm's improvement—that is, how well the participants thought the algorithm learned to recommend their preferred station—using a slide bar from "very poor" to "excellent."

The final phase of the experiment, the test phase, entailed a classification test to confirm whether participants still remembered the categories correctly. This phase followed the same procedure as the initial training phase, but did not provide participants with feedback (i.e., they were not told whether or not their categorization was correct). Afterwards, participants were asked to provide feedback about the experiment and identify the rule behind the classification they were trained on. Participants who successfully completed all phases of the experiment were compensated via MTurk.

**Analysis.** We quantify the behavior of the sampling algorithms by their trial-by-trial recommendation and predictive accuracies, as described previously. The test examples for computing the predictive accuracy consisted of a grid of 10-by-10 examples covering the area spanned by two pairs of $\{min, max\}$ sampled for each experiment (see the Stimuli section under Experiment 1).

We report the first trial index by which an algorithm's recommendation accuracy becomes statistically different from $50\%$ as well as the first trial index at which its recommendation accuracy becomes statistically no different from $95\%$. For these we use the binomial test and claim statistical significance when p-value is less than 0.05. We also report the trial index at which an algorithm's predictive accuracy converges. We formalize this as the first trial at which the predictive accuracy is not statistically different from the prediction accuracy at the last trial, using a one-sample t-test. The accuracy at the final trial is reported as the converged value.

We omit subjects whose test accuracy is below 18 out of 20 (below $90\%$). For the included subjects, we compute a consistency score, which is the fraction of their responses to the recommended examples that matched the expected response. For subjects whose consistency score is below $50\%$, we computed the predictive accuracies after flipping all their responses in the recommendation phase. This allowed us to correct for the responses from subjects who misremembered the preference during the interaction phase. A $0\%$ consistency would flip the classification algorithm's prediction on every test example. We assume that the fraction of properly predicted examples is proportional to consistency. Thus, to maximize the fraction of proper predictions, we flip responses when consistency is $< 50\%$. The number of included subjects are 26/30 (3 flipped) for random, 27/30 (4 flipped) for active

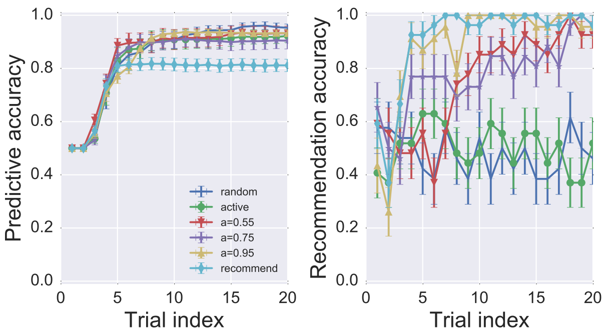

Figure 1: Predictive accuracy and recommendation accuracy for all six conditions over time (trial index indicated on x-axis).

learning, and 27/30 (3 flipped) for recommendation.

**Results.** Figure 1 shows the recommendation and predictive accuracies of the different sampling algorithms. As expected, examples chosen under the recommendation objective result in high recommendation accuracy, but low predictive accuracy. As a function of the number of examples seen (trial index), recommendation accuracy rises above chance level and reaches 95% after 4 examples, [1] while predictive accuracy converges after 5 examples to 81%, which is low compared to the active learning or random algorithms.

Conversely, recommendation accuracy under the active-learning algorithm results in low recommendation accuracy, but high predictive accuracy. As a function of trial index, recommendation accuracy remains at chance level, while predictive accuracy converges after 5 examples to 92%.

For reference, results of random sampling are also presented. These show a pattern similar to that observed for active learning. There is a rapid increase in predictive accuracy, converging after 8 examples to 95%. Recommendation accuracy remains at chance level throughout.

## Experiment 2: Exploring active recommendation

An ideal algorithm would combine both high recommendation and high predictive accuracy. As a function of the number of examples given, one hopes that the recommendation accuracy will approach 1 after a few examples, and the predictive accuracy will steadily increase to 1. Given the sharp dichotomy between the performance on recommendation and active learning, it is not obvious how best to achieve this.

We explore a simple, one parameter generalization of recommendation and active learning that we call, active recommendation. We investigate its trace of accuracy under a range of $\alpha = (0.55, 0.75, 0.95)$. We look at how the predictive and recommendation accuracies interpolate between the active learning and recommendation sampling as a function of $\alpha$. The new sampling algorithm is as described in Eq. 3. The stimuli and procedure are the same as Experiment 1.

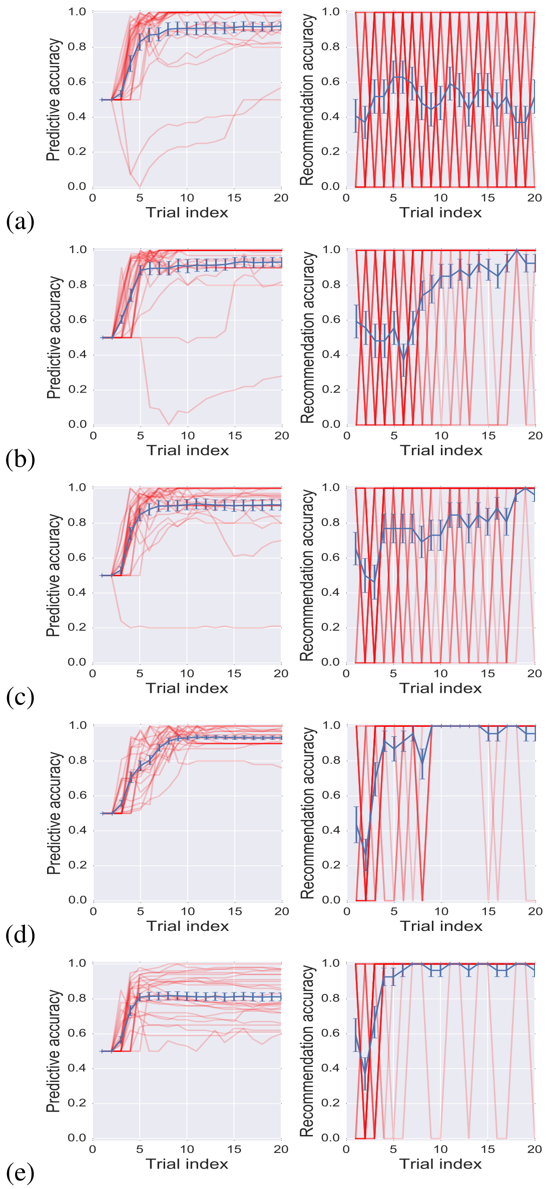

Figure 2: Predictive and recommendation accuracies for all included subjects broken down by condition. Red traces corresponds to individual subjects, and the blue curve is the average. (a) Active training; (b) $\alpha = 0.55$; (c) $\alpha = 0.75$; (d) $\alpha = 0.95$; (e) Recommend. Note that a few mislabeled examples in the early trials can lead to unstable behavior, such as those curves that dip below chance level.

**Participants.** The experiment was run on MTurk with 30 subjects for each of the 3 conditions: $\alpha = (0.55, 0.75, 0.95)$. Following the criteria described above, the number of subjects included in the analysis is 27/30 (4 flipped) for $\alpha = 0.55$, 26/30 (5 flipped) for $\alpha = 0.75$, and 23/30 (3 flipped) for $\alpha = 0.95$.

**Results.** Figure 1 shows the plot of predictive and recommendation accuracies for all conditions. The predictive accuracies in the active-recommendation conditions converge to 93%, 90%, and 93% for $\alpha = (0.55, 0.75, 0.95)$, respectively. These are similar to the 92% in the active condition and bet-

[1] This and subsequent numbers represent statistically significant results, as described above.

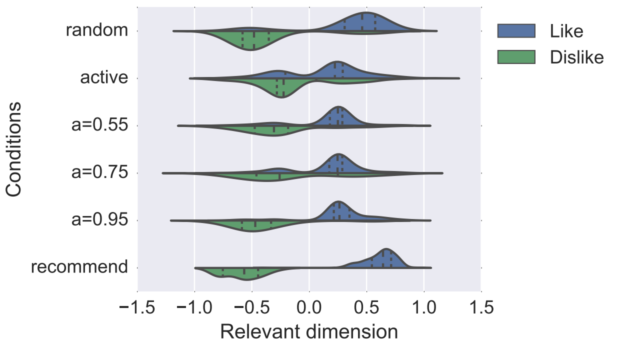

Figure 3: The distributions of like/dislike examples for each condition. The dotted lines in the distributions indicate the distributions' quartiles.

ter than the 81% in the recommend condition. The predictive accuracies of the active-recommendation conditions converged after 5, 5, and 8 examples for $\alpha = (0.55, 0.75, 0.95)$, respectively. The recommendation accuracies in the active-recommendation conditions reached 95% after 12, 16, and 4 examples for $\alpha = (0.55, 0.75, 0.95)$, respectively. These are similar to the recommendation condition in that all reached 95%, whereas recommendation accuracies in the active and random conditions remain at chance level.

Importantly, if we move slightly away from the active condition (sampling from slightly farther away from the boundary than active; i.e., from $\alpha = 0.5$ to 0.55), we can achieve much higher recommendation accuracy (it rises above chance after 8 examples and reaches 95% after 12 examples vs. at chance level throughout), while also achieving similar predictive accuracy. Similarly, if we move slightly away from the recommendation condition (from $\alpha = 1$ to 0.95), we can maintain the recommendation accuracy while improving the predictive accuracy. Thus, these intermediate conditions (in terms of $\alpha$) appear to allow the algorithms to uncover more of the space that is relevant.

Figure 2 employs the same measures as Figure 1, but displays each of the six conditions separately, with individual participant performance (red lines) and the results averaged over all participants (blue lines). Figures 2a-2e allow for a closer look at individual variability during the experiment, and in particular highlight the difference in recommendation accuracy from the active learning and $\alpha = 0.55$ conditions. If we compare Figures 2a and 2b, we can see that variation in recommendation accuracy across individuals persists for all trials in the active condition, while reducing greatly after 8 to 12 trials in the $\alpha = 0.55$ condition. Comparing Figure 2d and 2e, we can see the predictive accuracy across individuals varies much less in the $\alpha = 0.95$ condition than in the recommendation condition, resulting in the better average predictive accuracy for $\alpha = 0.95$.

The cause of the improved performance of intermediate $\alpha$ values can be traced back to the examples they select. The distributions of "likes" and "dislikes" are plotted in Figure 3 alongside random sampling, active learning, and recommendation. At the top, random sampling replicates the true distribution (up to some small number of inconsistent responses). Active learning selects examples that are evenly distributed across likes and dislikes but shifted toward the boundary between the two categories. At the bottom, recommendation selects examples that are skewed away from the boundary and the balance of examples is strongly tilted toward likes, consistent with the goal of recommending relevant examples. Of particular interest are the three alpha conditions. There are minor differences focused on the distribution of disliked items. What is most notable are the similarities among them and the active learning distribution for likes. Unlike the recommend condition, all three intermediate conditions disproportionately select "liked" examples that are close to the boundary. They all also select relatively few "disliked" examples. Cross-referencing against Figure 2, these disliked items happen only in the early trials. To summarize, the advantage of the active recommendation approach is a bias to select uncertain items *within* the relevant category. This allows them to achieve both high recommendation and predictive accuracy.

Interestingly, if we include only the fully consistent subjects, the $\alpha$ values dictate a strict ordering in both the predictive and recommendation accuracy. Increasing $\alpha$ from 0.5 to 1, one sees a monotonic decrease in the converged predictive accuracy and a monotonic increase in the rate at which recommendation accuracy reaches 1. The stochasticity in the subjects' responses can break the ordering in two ways. First, algorithms that provide examples closer to the boundary will receive more noisily labeled examples. Second, randomness in responses slows down the convergence of the classification algorithm. These effects cause the converged prediction accuracies, in small $\alpha$ conditions, to be lower than what they could be with less variable responses.

## Discussion

Information filters and recommender systems mediate between humans and the vast information and product stores on the Internet. Naturally, these algorithms aim to provide relevant information, but this goal may also lead to negative consequences by overly restricting experience. Embedding recommendation into a concept learning framework, we investigate the conditions under which we may observe high recommendation *and* predictive accuracy, in the presence of naturalistic human variability. We introduced a unified model of recommendation and active learning which we call active recommendation. In well-controlled experiments, we show that—across a wide range of parameterizations—active recommendation converges toward optimal predictive and recommendation accuracy. We also observe that parameterizations closer to pure recommendation yield better performance in terms of faster convergence and greater robustness to human variability. We trace the success of active recommendation to the fact that all parameterizations automatically combine rapid convergence toward selecting only relevant items and actively exploring informative examples from *within* that set. Parameterizations close to pure recommendation perform best because they minimize exploration of regions of the space where human actions are most variable—near the boundary and in the non-focal category.

Our approach is unusual in that the goal is to use humans to investigate the behavior of algorithms. This makes sense because the algorithms are meant for recommending options to humans. In contexts where recommendation is typically applied, however, there is no known ground truth, which makes assessing the performance of algorithms difficult. One could assume a ground truth and perform computational simulations, but these assume that your simulation is robust to human-like variability, which is rarely known or checked. In our experiments, humans were taught very simple concepts that governed relevance. They then labeled data for the algorithm, which captures the kinds of uncertainty associated with cognition—stochasticity across time, in response to recent input, and features of the concepts. The results bear the fruits of the approach. If one considers only the people who labeled correctly in their interactions, active recommendation performs comparably well across a wide range of parameterizations. However, human variability is concentrated at the boundary and toward the non-focal concept, which gives parameterizations closer to pure recommendation a distinct advantage in recommendation and predictive accuracy.

Our proposed unified model of active recommendation takes pure recommendation and active learning as a starting point. However, across a wide range of parameterizations, the unified model exhibits behavior that is qualitatively different from either. That is, it achieves good performance on both goals of recommendation and active learning simultaneously. It is useful to consider this behavior in contrast with more explicit alternative approaches, namely, managing the exploitation-exploration trade-off in reinforcement learning. Formalizing and training a policy about when to apply recommendation (exploitation) or active learning (exploration) would certainly be more involved than the simple model we presented; it would also arguably miss the point. The active recommendation approach, in denying the existence of the dichotomy, allows simultaneous optimization of recommendation and prediction.

Active recommendation can be recast as a social active learning model where an agent asks questions to learn from another agent who may not answer because of disinterest, ignorance, or some other factors. In these social scenarios, good questions should depend on the answerer's preference, knowledge state, etc.. In the cognitive development literature, empirical studies have shown that children select questions based on the answerer's expertise (Kushnir, Vredenburgh, & Schneider, 2013). This exemplifies an interesting connection between our model and human social learning.

Although active recommendation has demonstrated excellent performance, the problems considered here are vastly simpler than those more typical of real-world recommendation or information filtering. In light of this, one may reasonably ask whether the results are likely to generalize to more complex, high-dimensional problems. Of course, active learning becomes decreasingly tractable as the space grows. This is why active recommendation may be expected to perform well. Instead of exploring the space of possibilities, active recommendation focuses on exploring the space of relevant possibilities. An important direction for future work is to formalize and test this question.

Often experimental control and real-world relevance are seen in competition. However, there are ways in which they can and should be complementary. Real-world applications of machine learning are especially amenable to this due to their algorithmic nature. In addition to the user studies that are typical of the applied computer science, we propose that more controlled experimental and modeling approaches in cognitive science can shed light on the core strengths and limitations of these algorithms.

## Acknowledgments

This research was supported in part by NSF grant NSF-1549981 to P.S. and O.N. through IIS and SBE.

## References

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734–749.

Baeza-Yates, R. (2016). Data and algorithmic bias in the web. In *Proceedings of the 8th ACM Conference on Web Science.*

Bruner, J. S., Goodnow, J., & Austin, G. (1956). *A study of thinking*. New Brunswick, New Jersey: Transaction Publishers.

Elahi, M., Ricci, F., & Rubens, N. (2016). A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, *20*, 29–50.

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992, December). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, *35*(12), 61–70.

Kushnir, T., Vredenburgh, C., & Schneider, L. A. (2013). who can help me fix this toy? the distinction between causal knowledge and word knowledge guides preschoolers' selective requests for information. *Developmental psychology*, *49*(3), 446.

MacKay, D. J. (1992). Information-based objective functions for active data selection. *Neural computation*, *4*(4), 590–604.

Maes, P., et al. (1994). Agents that reduce work and information overload. *Communications of the ACM*, *37*(7), 30–40.

Markant, D. B., & Gureckis, T. M. (2014). Is it better to select or to receive? learning via active and passive hypothesis testing. *Journal of Experimental Psychology: General*, *143*(1), 94.

Nelson, J. D. (2005). Finding useful questions: on bayesian diagnosticity, probability, impact, and information gain. *Psychological Review*, *112*(4), 979.

Pariser, E. (2011). The filter bubble: What the internet is hiding from you. *Penguin Press*.

Robertson, S. (1977). The probability ranking principle in ir. *Journal of Documentation*, *33*(4), 294-304.

Salton, G., Fox, E. A., & Wu, H. (1983). Extended boolean information retrieval. *Communications of the ACM*, *26*(11), 1022–1036.

Shepard, R. N., Hovland, C. I., & Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological Monographs: General and Applied*, *75*(13), 1.

Sparck Jones, K. (1970). Some thoughts on classification for retrieval. *Journal of Documentation*, *26*(2), 89–101.

Van Rijsbergen, C. (1979). *Information retrieval.* London, UK: Butterworths.