An Experimental Investigation of Adaptive Algorithm Understanding

Kazunori Terada (terada@gifu-u.ac.jp)
Department of Information Science, Gifu University
1-1 Yanagido, Gifu, Japan 501-1193

Seiji Yamada (seiji@nii.ac.jp)
National Institute of Informatics / SOKENDAI / Tokyo Institute of Technology
2-1-2 Chiyoda, Tokyo, Japan 101-8430

Akira Ito (ai@gifu-u.ac.jp)
Department of Information Science, Gifu University
1-1 Yanagido, Gifu, Japan 501-1193

Abstract

There have been few studies on a cognitive model for algorithm understanding in a human-computer cooperative situation. In the present study, we conducted an experiment with participants to investigate the cognitive process of higher level abstraction (algorithm understanding) performed in a human-computer collaboration task. The most recently used (MRU) algorithm, known to be one of the simplest adaptive algorithms, and probabilistic MRU algorithm were used to test the human capability to understand an algorithm. The experimental results showed that inductive reasoning in which participants observed the history of computer action, and they updated a statistical model while restricting their focus on a certain history with deteministic bias and Markov bias played key role to correctly understand the MRU algorithm. The results also showed that deductive reasoning was used to understand algorithms when participants rely on prior knowledge, and that there was a case in which the algorithm, even known to be the simplest one, was never understood.

Keywords: algorithm understanding; inductive reasoning; deductive reasoning; adaptive user interface;

Introduction

The number of situations in which humans collaborate with computers has been increasing with the advance of information technology. Although user-adaptive systems that adapt to a user, including adaptive user interfaces, have been a main topic in the human-computer interaction community and artificial intelligence machine learning community (Findlater & McGrenere, 2004; Oviatt, Swindells, & Arthur, 2008; Bigdelou, Schwarz, & Navab, 2012), an adequate design policy for implementing useful user-adaptive systems still remains unclear (Shneiderman & Maes, 1997; Lavie & Meyer, 2010; Gajos, Everitt, Tan, Czerwinski, & Weld, 2008). Furthermore, there have been few studies on a cognitive model for algorithm understanding in the context of human-computer collaboration tasks.

In a human-human collaboration task, mutual intention understanding plays the key role in accomplishing successful work (Byrne & Whiten, 1988; Call & Tomasello, 2008). However, in a collaboration task with a computer, the abstraction level of behavior necessary to understand a collaborator's behavior is lower than that used in a human-human collaboration task (Dennett, 1987). Behavior abstraction in terms of goal (intention) is not required in human-computer collaboration because a goal is given and explicitly shared with both a human and a computer. Instead, algorithm level abstraction is needed. In a human-computer collaboration task, understanding a computer's algorithms in order to accomplish the given goal is quite important because a human relies on the computer's underlying mechanisms in order to predict its behaviors and to adapt to it.

One way to predict the future behavior of a target is to use input-output association acquired on the basis of *sequence learning* (Clegg, DiGirolamo, & Keele, 1998; Sun & Giles, 2001a; Winkler, Denham, & Nelken, 2009). In a typical sequence learning problem (Nissen & Bullemer, 1987), humans learn a recurring loop of action sequences from given examples, and as a result, their reaction time for the given examples decreases. This learning is done both explicitly and implicitly (sensory-motor learning), and currently, implicit sequence learning is actively studied (Sun & Giles, 2001b). The situation in which humans observe only the action sequences given to them is the same in both sequence learning and algorithm understanding. However, the learning target of algorithm understanding is procedures with variables that describe the internal states of computers, and this target is quite different from that of sequence learning (i.e., sequence patterns of values). Obviously, the number of hypotheses in algorithm understanding is far more than that in sequence learning, and this makes algorithm understanding very hard. Hence, algorithm understanding requires quite strong biases to find adequate algorithms. Another difference between understanding cooperative algorithms and sequence learning is the type of interactivity in the tasks. In algorithm understanding in a cooperative situation, a computer's behaviors change depending on the behaviors of humans because it adapts to them. In sequence learning, sequences are given to humans as physical stimuli.

The research objective of this study is to build a cognitive model to describe the human capability to understand computer algorithms in the context of a human-computer collaboration task. We introduce one of the simplest human-computer collaboration tasks, in which a computer adapts to humans who are asked to try and understand the computer algorithms. Concretely, we investigated how humans understand the most recently used (MRU) algorithm (Lee et al., 1999; Findlater & McGrenere, 2004; Gajos et al., 2008). The MRU algorithm is well known to be one of the simplest adaptive algorithms in which a computer's current statement simply corresponds to the user's last one. Examples of the implementation of the MRU algorithm are the *most recently used files* (Amer & Oommen, 2006), which lists the user's most recently accessed files in an application, and the most recently used menu (called *adaptive menu* (Arcuri, Coon, Johnson, Manning, & Tilburg, 2000)), which lists the user's most recently used menu.

The MRU algorithm has succeeded in contributing to making useful interactive software that includes adaptive user interfaces (Findlater & McGrenere, 2004). One reason is that it can be easily understood by users. If users can not find any meaning (regularity or rules for computer's behaviors) from a list in which the order of the items is frequently changed, the list causes the user stress. The reason the MRU algorithm is easily understood is that there are explicit descriptions of the algorithm, i.e., there may be a description such as "most recently used file." In this work, we investigate the human ability to understand an algorithm in a situation without such explicit knowledge.

One preferable explanation of algorithm understanding is *induction* because rule finding is considered to be an inductive process (Haverty, Koedinger, Klahr, & Alibali, 2000; Simon & Kotovsky, 1963; Verguts, Maris, & Boeck, 2002; Schmid & Kitzelmann, 2011). In general, induction needs to be done only with a small number of examples. It is hard to induce adequate rules with finite examples that can cover infinite facts because there is a huge number of hypotheses of rules that can be induced from the examples. Thus, we need heuristics (called *inductive biases*) to sufficiently restrict the hypothesis of rules for practical induction. In algorithm understanding, since humans have to induce computer algorithms only with tens of examples, we consider they have a strong bias for algorithm understanding. In this paper, to investigate human algorithm understanding, we hypothesize biases on algorithm understanding and verify them in experiments with participants.

Cognitive Model of Adaptive Algorithm Understanding

Adaptive algorithm understanding is a subclass of algorithm understanding. An adaptation in human-computer interaction refers to a feature of algorithms in which strategies of a computer dynamically change according to user's input in order to pursue given goals. The goals refer not only corporation but also competition (Hampton, Bossaerts, & O'Doherty, 2008). In the present study, we focus on a cooperative situation. We introduce a *cooperative mark-matching game* as a simplified and generalized task of *human-computer adaptation* in which a user adapts to a user-adaptive system.

Cooperative Mark-Matching Game

The cooperative mark-matching game is a repeated game with two players. Each player has the same marks (e.g., ♠, ♢, ♡) and must secretly choose one of the marks. The players then reveal their own choices simultaneously. If the marks match each other, *both players* obtain a certain score, and if not, nobody obtains a score. In our experiments, the two players were a human and a user-adaptive system.

In a situation of the human-computer adaptation, a system predicts the user's next action (e.g., a menu item that will be chosen next by a user in an adaptive menu (Findlater & McGrenere, 2004)) and adapts to him/her by modifying the user interface (e.g., changing the menu item positions (Findlater & McGrenere, 2004)). If the prediction is correct (i.e., the two marks of the human and user-adaptive system matched in the game), the user and system obtain efficiency together. The number of the mark corresponds to the number of menu items in the adaptive menu. The key difference between a cooperative mark-matching game and human-computer adaptation with AUIs is that a user can freely choose his/her next action by him/herself in the game in contrast to the user's action sequence being determined to achieve a task with AUIs.

While the simplest strategy for a cooperative game is for participants in each trial to simply choose the action that in the recent past gave the most rewards (known as reinforcement learning), a more sophisticated strategy is to try to predict the system's next actions by taking into account a statistical model constructed on the basis of the history of prior actions. Studies on game theory (Fudenberg & Levine, 1998)(Berger, 2005) and sequence learning (Sun & Giles, 2001a) with an opponent player (a user-adaptive system) in a game situation suggest that opponent strategy is identified on the basis of a mixed strategy, which is defined as a probability distribution over the alternative actions available to each player.

Statistical model

We hypothesize that, as mentioned earlier, a higher level abstraction, i.e., *algorithm identification*, for a computer's strategy is carried out on the basis of biases. We set the starting point of our discussion to statistics in which a human updates the conditional probability distribution of the system's next choice over time.

$$p(a^s_t | a^s_{t-1}, \cdots, a^s_j, a^h_{t-1}, \cdots, a^h_k) \quad (1)$$

, where a^h, $a^s \in A$, and A are available choices for both the system and human and $a^s_{t-1}, \cdots, a^s_k$ and $a^h_{t-1}, \cdots, a^h_j$ are the past choices of the system and human, respectively. Indices j and k denote the length of the history, which the human takes into account, and vary depending on focus. However, detecting the computer's algorithm on the basis of only observed behaviors is an ill-posed inverse problem because humans do not know how to restrict their focus to a certain history, and in addition, different strategies sometimes produce the same

Table 1: Conditional probability distributions correspond to most recently used and probabilistic most recently used algorithm

		a^h_{t-1}		
		♡	♠	♢
a^s_t	♡	1	0	0
	♠	0	1	0
	♢	0	0	1

(a) MRU

		a^h_{t-1}		
		♡	♠	♢
a^s_t	♡	.9	.05	.05
	♠	.05	.9	.05
	♢	.05	.05	.9

(b) Probabilistic MRU

history. Thus, we consider that a human does sufficiently restricted exploration with inductive biases.

The MRU algorithm is formalized as the following distribution.

$$p(a^s_t|a^h_{t-1}) \tag{2}$$

The actual distribution produced by the MRU algorithm in the cooperative mark-matching game is shown in Table 1(a). The system's choice (a^s_t) depends only on the human's most recent choice (a^h_{t-1}) and is independent from any other history of choices. If the human's most recent choice is heart, for example, the system's next choice will be heart, represented as $p(a^s_t = \heartsuit | a^h_{t-1} = \heartsuit) = 1$. Infinite numbers of trials are, theoretically, required to convince a human that the probability is 1. Hence, one reasonable strategy for this problem is to use inductive biases to adequately control the inference process. As such inductive biases, we consider *deterministic bias* and *Markov bias*. If a human has a deterministic bias that assumes computer's behaviors are deterministic, not probabilistic, only one piece of evidence is necessary to estimate the probability distribution. Markov bias, in which the conditional probability distribution of the next choice depends only upon the present choice, not on the sequence of events, is also necessary to ignore any unnecessary history of choice.

Experiments

We conducted an experiment with participants to investigate the cognitive process of higher level abstraction (algorithm identification) performed in the context of a human-computer cooperation task. The MRU algorithm and probabilistic MRU algorithm was used to test the human capability of algorithm understanding. Participants were asked to play a cooperative game with a computer, and after that they were asked to answer the computer's algorithm.

A 50-round repeated cooperative mark-matching game with different statistical profiles of the MRU algorithm was used. We used the following two conditions.

Deterministic (D) condition Computer's choice is completely the same as the human's most recent choice (deterministic MRU algorithm, see Table 1(a)).

Probabilistic (P) condition Although 90% of the computer's choices are the same as the human's most recent choices, 10% differs (probabilistic MRU algorithm). The actual distribution produced by the probabilistic MRU algorithm is shown in Table 1(b).

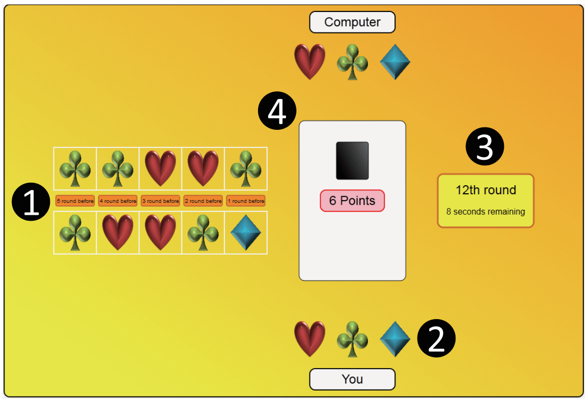


Figure 1: Interface of on-line experimental system: 1) history of both players' choices, 2) choice marks (marks are clickable), 3) round number and remaining time, 4) place for unveiling players' choice and scores for both players

The P condition was prepared to contrast the effect of noise on the inductive reasoning performed to understand the MRU algorithm. In particular, we expected that the deterministic bias was strongly affected by the noise and performance deteriorated in the P condition. It was also expected that the score of those who participated in the P condition was at most 10% worse than that of the D condition if the participants merely estimated the probability distribution and did not use any biases to identify an algorithm.

Experimental setup and measurement

The game was implemented with JavaScript and HTML and played in a Web browser (Firefox). Figure 1 shows the game interface. The computer's choices were automatically controlled by a JavaScript program. Participants were instructed to click the mark corresponding to his/her choice within 10 seconds for every round. Scores for both players were shown in the interface. The choices of the past five rounds for both players remained displayed so that the participant was able to recognize the computer's strategy.

A single-factor two-level between-subject experimental design was used. Fifty people (9 female) aged 19 to 47 (mean = 28) recruited via direct e-mail participated in the experiment. All participants had moderate to high experience using computers. Participants were randomly assigned to either a deterministic or probabilistic condition. Participants were informed of an ostensible goal of the experiment - that the point of the experiment was to assess the usability of an on-line game system. They were also informed that "the computer was cooperative." Participants were told that they would win a PC gadget as a prize according to the score (under 20 points: around \$5, 21 to 44 points: around \$15, 45 to 50: around \$30).

In the P condition, a 50-round sequence with 10% random noise, which corresponds to 5 rounds in which MRU rules

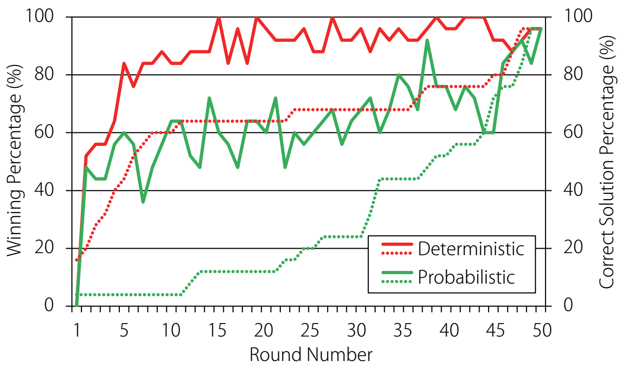


Figure 2: Percentage of participants who won each round (solid line) and percentage of participants who started to take a "fixed choice strategy" (correct solution to the game) throughout the remaining rounds (dotted line)

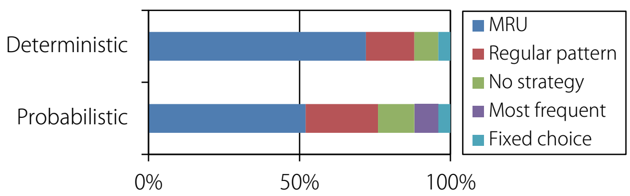


Figure 3: Computer's algorithm identified by participants

are violated, is generated, and sequences that do not fit the following criteria are omitted: 1) errors do not appear in the first and last 5 rounds and 2) five errors appear within the remaining 40 rounds. The computer's choice for the first round was selected not to match the participant's choice in both conditions.

The outcomes of all 50 rounds were recorded. The round in which participants became aware of the correct solution to the game was identified by detecting the round in which participants started to continue to select the same mark throughout the remaining rounds. After the game, participants were asked to answer 7-point Likert scale questions, such as *Q. Did the computer make its choices strategically?*, and one open-ended question if participants gave a rating of 5 to 7 (positive) to this question - *Describe the computer's strategy*.

Results

The average scores were 43.7 (SD = 7.0) in the D condition and 31.4 (SD = 7.5) in the P condition. ANOVA revealed that there was statistically significant difference $(F(1,48) = 33.99, p < 0.01)$ between the two conditions. The difference of the average scores between the two conditions was 12.3. A difference of more than 5 (10%) indicates that participants used deterministic bias to accomplish the game. This gap is explained by the difference in the increasing rate of the winning percentage. While the winning percentage of the D condition rapidly reached a high value (e.g., 80% at the sixth round), that of the P condition slowly increased (e.g., 80% at the 35th round). The slower increase of the winning percentage in the P condition indicates that the 10% noise in the MRU algorithm caused the computer's algorithm to be difficult to identify and made the participants require longer rounds to identify it.

Figure 2 shows the percentages of the participants who won the round plotted against the round numbers (solid line). The dotted line in Figure 2 represents the percentage of participants who took a "fixed choice strategy," indicating the percentage of participants who became aware of the correct solution to the game. Note that the correct solution is found not only by identifying the MRU algorithm, but also by merely choosing the same mark without thought.

Figure 3 illustrates the computer's algorithm identified by participants. While 72% of participants in the D condition correctly identified the MRU algorithm after 50 rounds, only 52% in the P condition succeeded in identifying it. However, a chi-square test revealed that there was no statistically significant difference in the distribution of the identified algorithm between the two conditions ($\chi^2(4) = 3.41$, $p = 0.49$).

Discussions

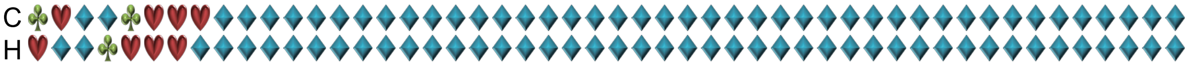
In the present study, we investigated the human capability to understand the MRU algorithm. In particular, we expected that inductive biases such as deterministic and Markov bias are used to understand the algorithm. In the succeeding subsections, we will discuss whether these biases were applied to accomplish the game.

Inductive algorithm understanding

The red dotted line in Figure 2 reveals that 60% of participants (15 participants) in the D condition found the correct solution to the game. The result of the questionnaire revealed that while 13 of the 15 participants inferred the computer's algorithm as the MRU, one inferred no strategy, and one inferred a fixed choice. A typical behavioral pattern for these kinds of participants is shown in Figure 4(a). They observed the history of the choices and might have inferred the MRU algorithm on the basis of the obtained statistical model. However, while detecting a statistical model of the computer's strategy essentially requires an infinite number of trials, they rapidly identified certain algorithms. One explanation for this rapid identification is the *deterministic bias* and *Markov bias*. If the algorithm was assumed to be deterministic, the participants did not need to take into account the six cases filled out as zero in Table 1(a) and required at least three trials to determine the computer's strategy. Without Markov bias, participants could not focus only on the one round past choice and required longer rounds.

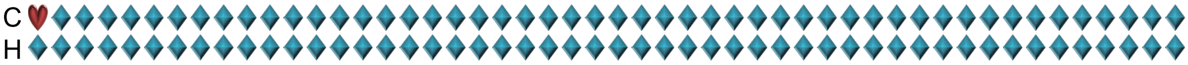
The deterministic bias also accounts for the worse performance of those who participated in the P condition. If the participants merely estimated the probability distribution, as expected, an optimal strategy against a mixed strategy would have been taken, and performance would have been at most 10% worse than in the D condition.

The lowest score for all 50 participants was 19, which was higher than the theoretically calculated score (16.67) when



(a) Understanding algorithm on the basis of inductive reasoning (correct identification). After eight trials of active learning phase, the participant realized the algorithm was the MRU one.

(b) Understanding algorithm on the basis of inductive reasoning (wrong identification). The detected algorithm was "the computer increased the number of times by repeating the same choice."



(c) Understanding algorithm on the basis of deductive reasoning. The participants used a heuristic from the beginning: "Adaptive system ⇒ MRU algorithm."

(d) The participant did not detect any algorithm.

Figure 4: Examples of typical behavioral pattern in the D condition. C: computer, H: human.

participants did not take any strategy, i.e., a random strategy. This implies that almost all of the participants arbitrarily attributed some kind of strategy to the computer's choice. In fact, the rules of the game allowed the participants to attribute strategies other than the MRU algorithm, such as "the computer simply selected the same mark" (fixed choice strategy) and "the computer changed its choice alternatively" or "the computer increased the number of times by repeating the same choice such as ♢♠♠♡♡♡" (increasing number strategy), see Figure 4(b)). Three participants in the D condition answered that the computer's algorithm was "increasing number strategy." Interestingly, they did not aware that the timing to change the mark was determined by themselves. They completely unaware of the rule in which the computer changed its output according to their input.

Deductive algorithm understanding

The results also indicated that some participants understood the algorithm on the basis of deductive reasoning. Sixteen percent of participants (four participants) in the D condition and four percent (one participant) in the P condition fixed their choice in the first round and never changed during the game (see Figure 4(c)). Surprisingly, all of them described their identified computer algorithm as the MRU. The prior knowledge given to the participants in the instruction phase lead them to deduct the following logic:

$$Adaptive\ system \Rightarrow MRU\ algorithm \tag{3}$$

In the instruction phase, participants were explicitly informed that the goal of the task was to get as much points as possible in cooperation with the partner computer. This *top down adaptive bias* might have enabled them to identify the algorithm immediately without exploring the computer's strategies. They might have logically inferred that the cooperative system acted adaptively to humans and that the most efficient algorithm for human-computer cooperation was the MRU algorithm. In fact, their MRU algorithm hypothesis was confirmed by the computer's succeeding choice. The *confirmation bias* (Klayman & Ha, 1987) was used to convince them that the computer used the MRU algorithm. They marked the highest score 49 (all participants were sure to lose the first round because of the game setting). There was no incentive to explore another strategy and gather evidence to test another hypothesis unless their hypothesis was violated because their goal was to get as many points as possible and not to detect the algorithm exactly. Indeed, while three participants in the P condition started to fix their choice in the first round, two of the three changed their choice after the noise pattern in the computer's choice appeared, indicating that their confirmation bias was destroyed by the noise (*falsification*).

Algorithm detection fail

Surprisingly, even though the MRU has been supposed to be one of the most predictable adaptation algorithms, the result showed that two participants in the D condition and three in the P condition failed to identify any strategy in the 50 rounds (see Figure 4(d)). The visual cue shown in the history area in the game ' s interface might have been a strong cue indicating that the computer's choice was the same as participant's choice one round before. However, they could not detect the algorithm. Further investigation will be required to account for this failure.

Summary

To the best of our knowledge, this is the first study to investigate the human capability to understand adaptive algorithm in a human-computer collaboration task. In the theoretical model of a human cognitive process for algorithm understanding, a user identifies a computer's algorithm by estimating the conditional probability distribution associated with a particular strategy and restricting his/her focus on certain history data by using inductive biases. The most recently used (MRU) algorithm, known to be one of the simplest adaptive algorithms, was used to test the human capability to understand an algorithm. The probabilistic MRU algorithm was also used to contrast the effect of noise on the inductive reasoning performed to understand the MRU algorithm. The experimental results indicated that most participants correctly identified the MRU algorithm and used deteministic bias and Markov bias in their inductive reasoning for algorithm identification. The results also indicated that some participants understood the algorithm on the basis of deductive reasoning. Surprisingly, few participants failed to identify any algorithm within 50 rounds.

The present findings implies that designed behavior of computers is not necessarily understood correctly, suggesting that both an understandable algorithm and transparency of the internal state of a computer might be important for designing effective adaptive systems.

Acknowledgments

This work was supported by MEXT KAKENHI Grant Number 24118703.

References

Amer, A., & Oommen, B. J. (2006). Lists on lists: A framework for self-organizing lists in environments with locality of reference. In *Proceedings of the 5th International Workshop on Experimental Algorithms* (pp. 109–120).

Arcuri, M., Coon, T., Johnson, J., Manning, A., & Tilburg, M. van. (2000). *Adaptive menus.* (US Patent: 6,121,968).

Berger, U. (2005). Fictitious play in 2×N games. *Journal of Economic Theory*, *120*, 139–154.

Bigdelou, A., Schwarz, L., & Navab, N. (2012). An adaptive solution for intra-operative gesture-based human-machine interaction. In *Proceedings of the 17th International Conference on Intelligent User Interfaces* (pp. 75–84).

Byrne, R. W., & Whiten, A. (1988). *Machiavellian intelligence: Social expertise and the evolution of intellect in monkeys, apes, and humans* (R. W. Byrne & A. Whiten, Eds.). Oxford Science Publications.

Call, J., & Tomasello, M. (2008, May). Does the chimpanzee have a theory of mind? 30 years later. *Trends in Cognitive Sciences*, *12*(5), 187–192.

Clegg, B. A., DiGirolamo, G. J., & Keele, S. W. (1998). Sequence learning. *Trends in Cognitive Sciences*, *2*(8), 275–281.

Dennett, D. C. (1987). *The intentional stance*. Cambridge, Mass, Bradford Books/MIT Press.

Findlater, L., & McGrenere, J. (2004). A comparison of static, adaptive, and adaptable menus. In *Proceedings of the 11th international conference on intelligent user interfaces* (pp. 89–96).

Fudenberg, D., & Levine, D. K. (1998). *The theory of learning in games*. MIT Press, Cambridge, MA.

Gajos, K. Z., Everitt, K., Tan, D. S., Czerwinski, M., & Weld, D. S. (2008). Predictability and accuracy in adaptive user interfaces. In *Proceeding of the 26th annual sigchi conference on human factors in computing systems* (pp. 1271–1274).

Hampton, A. N., Bossaerts, P., & O'Doherty, J. P. (2008). Neural correlates of mentalizing-related computations during strategic interactions in humans. *Proceedings of the National Academy of Sciences*, *105*(18), 6741–6746. (fictitious play, Reinforcement Learning)

Haverty, L. A., Koedinger, K. R., Klahr, D., & Alibali, M. W. (2000). Solving inductive reasoning problems in mathematics: Not-so-trivial pursuit. *Cognitive Science*, *24*(2), 249–298.

Klayman, J., & Ha, Y. won. (1987). Confirmation, disconfirmation, and information in hypothesis testing. *Psychological Review*, *94*(2), 211–228.

Lavie, T., & Meyer, J. (2010). Benefits and costs of adaptive user interfaces. *International Journal of Human-Computer Studies*, *68*, 508–524.

Lee, D., Choi, J., Kim, J.-H., Noh, S. H., Min, S. L., Cho, Y., et al. (1999). On the existence of a spectrum of policies that subsumes the least recently used (lru) and least frequently used (lfu) policies. In *Proceedings of the 1999 acm sigmetrics international conference on measurement and modeling of computer systems* (pp. 134–143). New York, NY, USA: ACM.

Nissen, M., & Bullemer, P. (1987). Attentional requirements of learning: Evidence from performance measures. *Cognitive psychology*, *19*(1), 1–32.

Oviatt, S., Swindells, C., & Arthur, A. (2008). Implicit user-adaptive system engagement in speech and pen interfaces. In *Proceedings of the 26th annual sigchi conference on human factors in computing systems* (pp. 969–978).

Schmid, U., & Kitzelmann, E. (2011). Inductive rule learning on the knowledge level. *Cognitive Systems Research*, *12*(3–4), 237–248.

Shneiderman, B., & Maes, P. (1997). Direct manipulation vs. interface agents. *Interactions*, *4*(6), 42–61.

Simon, H. A., & Kotovsky, K. (1963). Human acquisition of concepts for sequential patterns. *Psychological Review*, *70*(6), 534–546.

Sun, R., & Giles, C. (Eds.). (2001b). *Sequence learning: Paradigms, algorithms, and applications* (Vol. 1828). Springer.

Sun, R., & Giles, C. L. (2001a). Sequence learning: From recognition and prediction to sequential decision making. *IEEE Intelligent Systems*, *16*(4), 67–70.

Verguts, T., Maris, E., & Boeck, P. D. (2002). A dynamic model for rule induction tasks. *Journal of Mathematical Psychology*, *46*(4), 455–485.

Winkler, I., Denham, S. L., & Nelken, I. (2009). Modeling the auditory scene: predictive regularity representations and perceptual objects. *Trends in Cognitive Sciences*, *13*(12), 532–540.