Detecting Changes in Mental Models during Interaction

Chuyang Wu[1,2,*,†], Shanshan Zhang[1,†] and Jussi P. P. Jokinen[2]

[1]University of Helsinki, Pietari Kalmin katu 5, 00560 Helsinki Finland
[2]University of Jyväskylä, Seminaarinkatu 15, PL 35, 40014 Jyväskylä Finland

Abstract

This paper introduces a novel computational cognitive model that maps latent mental models to observable behaviors, allowing the system to detect changes in users' mental models from their actions. We propose an inference framework to dynamically adjust to the user's evolving understanding and decision-making processes. An empirical experiment demonstrates the framework's ability to accurately detect shifts in users' mental models based on their interactions. The results indicate a consistent improvement in prediction accuracy and a decrease in variance over time, suggesting the model's potential for real-time application in designing adaptive interactive systems.

Keywords

Human-Computer Interaction, User Modeling, Collaborative Human-Computer Systems, Adaptive Systems

1. Introduction

An intelligent interactive system needs to adapt to the behaviors of its users. It should understand their intentions, and anticipate what's coming next. A user's interactive behavior is shaped by their mental model, the user's knowledge and beliefs of the interactive system [1], which is not directly observable. We can parameterize the mental model to build a computational user model [2]. In such a model latent (i.e., unobservable), factors are mapped to observed behavior, allowing us to formalize the mechanism of interactive behavior. We can then build adaptive systems that solve for the mental model from observations, and the interactive system can be designed to adapt accordingly.

However, a problem in inferring mental models is that they are not static during interaction. For example, as users become more experienced, their mental models change [1]. Failures of the interactive system to detect these changes would lead to wrong or obsolete inference of mental models and ineffective adaptation, to the detriment of the user.

Consider a hypothetical scenario involving a multisensor smart scanner that can obtain ultrasound and radio frequency readings of boxes at a warehouse. Suppose that different contents produce different sensor readings. Harry, a novice operator yet to learn to read radio frequencies, relies solely on ultrasound to determine the content. Accordingly, the scanner should provide hints on how to interpret ultrasound readings. If Harry suddenly scans for radio frequency data, it will likely be a mistake, and the scanner should intervene to avert it.

Harry practices reading radio frequency data and associating the readings with the contents. At some point, his mental model – an internal representation of the dynamics and facts of the external task – evolves to have a closer correspondence with reality. If the AI of the scanner does not pick up on this evolution, it will continue to recognize Harry's actions as mistakes and offer ineffective or detrimental hints. Therefore, intelligent interactive systems must accurately infer user's changing mental models to provide useful adaptation.

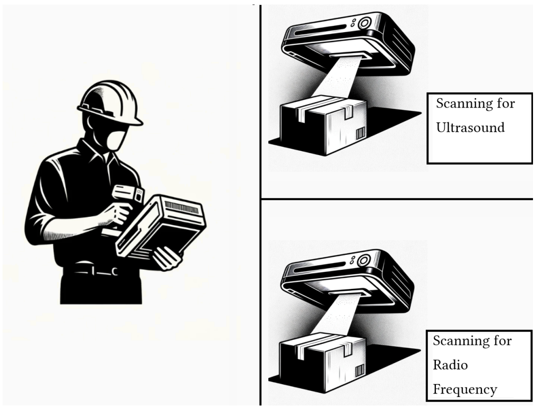

Figure 1: Harry is a novice warehouse operator who previously only understood ultrasound readings. Now he starts to scan for radio frequencies. Is this a mistake, or has he learned how to read radio frequencies?

In this paper, we propose a computational model of interaction that accounts for how changes in the mental model lead to changes in interactive behavior. We then define a framework to infer and quantify the mental model from observed behavior and demonstrate how to detect changes in parameter value from behavioral data with an empirical experiment. In summary, this paper contributes to the computational modeling of interactive behavior by proposing:

- a computational model of how interactive behavior emerges from quantified mental models;
- an inference framework to detect these changes from observed behavior.

2. Background Review

In human-computer interaction, mental models represent how the interaction is internally interpreted and reconstructed by the users [3]. How closely a user's mental model matches the real interactive environment would determine the effectiveness and efficiency of the user's interactive strategy [4]. Particularly, suppose a user fails to understand the designs of an interactive system. In that case, it is more likely that the mental model would be poor, and the user would likely end up missing their goals and have a frustrating experience.

TKTP 2024: Annual Doctoral Symposium of Computer Science, 10.-11.6.2024 Vaasa, Finland
*Corresponding author.
†These authors contributed equally.
chuyang.wu@helsinki.fi (C. Wu); shanshan.zhang@helsinki.fi (S. Zhang); jussi.p.p.jokinen@jyu.fi (J. P. P. Jokinen)

Interactive systems are often designed to adapt to user needs and habits to create an intuitive user experience. The classic approach is to collect behavioral data, such as keystrokes, mouse movements, or system logs, and analyze it for patterns [5]. Interactive systems would update based on similarities between user behaviors and learned patterns. These approaches, however, do not explain the reasons behind the user's actions. When designing such a system, it is therefore desirable for the system to align with the users' mental models [6]. To do so would require a *model of the user's mental model* that accounts for user behavior and decision-making [7], allowing the interactive systems to adapt to the user's goals [8, 9].

Parameterized, computational models of interaction have been proposed to explain the user's decision-making process during an interaction [10, 11]. These models establish a causal link between observed user behavior and latent psychological factors and parameterize the latter to build a computational framework, thus paving a way to infer the values of latent factors from observed behavior [12, 13]. This approach can be extended to study the effect of mental models on user behavior, enabling the design of intelligent interactive systems that adapt to users' mental models.

However, these models have not addressed cases where the latent factors change. A user could gain knowledge and experience during an interaction to become more skillful, which would be reflected in the mental model. Failure to account for such changes would render any interactive system's adaptation ineffective or even detrimental. Consequently, our present work formalizes a computational framework for interaction that detects changes in mental models based on observed user behavior. This would be important for creating intelligent interactive systems and collaborative AI that are truly adaptive to the users.

3. Method

3.1. Interaction as a POMDP

We view the user of an interactive system as an agent trying to solve a Partially Observable Markov Decision Process (POMDP) [14]. POMDP is defined as a tuple $(S, A, T, R, O, \Omega, \gamma)$ where:

- S is a finite set of states of the environment.
- A is a finite set of actions available to the agent.
- $T : S \times A \times S \rightarrow [0, 1]$ is the (probabilistic) transition function, where $T(s, a, s') = P(s'|s, a)$ represents the probability of transitioning to state s' when action a is taken in state s.
- $R : S \times A \times S \rightarrow \mathbb{R}$ is the reward function for each transition from s to s' due to a.
- O is a finite set of possible observations.
- $\Omega : S \times A \times O \rightarrow [0, 1]$ is the (probabilistic) observation function, where $\Omega(s', a, o) = P(o|s', a)$ represents the probability of observation o after action a, in state s'.
- $\gamma \in [0, 1]$ is the discount factor for the present value of future rewards.

The interaction process between an agent and a POMDP environment can now be described in Figure 2. In a POMDP, the agent cannot know the environment state directly. Instead, it observes the state and forms an internal representation of the state as a belief $b \in B$, with B being the set of all

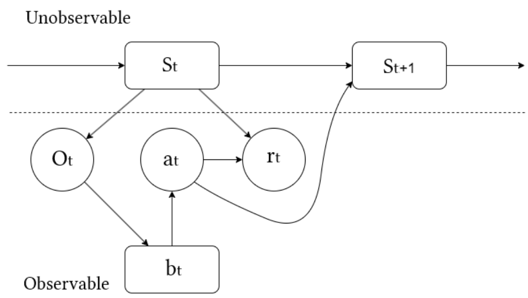

Figure 2: Interaction as a POMDP. The states S are not directly observable. The agent makes an observation O, from which a belief b is formed. Based on the belief, the agent takes an action a which leads to a reward r, as well as a transition to the next state. The reward depends on both the state and the action.

possible beliefs. The agent aims to find an optimal policy $\pi : B \rightarrow A$ to guide its choice of action that maximizes the expected discounted rewards over time. Specifically, the interaction takes place as follows.

1. **Initial Belief State:** The interaction starts with the agent having an initial belief state, $b_0(s)$ representing the agent's initial knowledge about the environment, $b_o \in B, s \in S$.
2. **Action Selection:** At each time step t, the agent selects an action $a_t \in A$ based on its current belief state $b_t(s)$ according to a policy π to maximize the expected reward.
3. **Environment Response:** The environment transitions from s_t to s_{t+1} according to $T(s_t, a_t, s_{t+1}) = P(s_{t+1}|s_t, a_t)$. This is not directly observable by the agent.
4. **Observation:** The agent receives an observation $o_{t+1} \in O$, generated according to the observation model Ω: $O(s_{t+1}, a_t, o_{t+1}) = P(o_{t+1}|s_{t+1}, a_t)$.
5. **Belief Update:** The agent performs Bayesian update of its belief to $b_{t+1}(s)$ with observation o_{t+1}, action a_t, and previous belief $b_t(s)$, and revises knowledge about the environment.
6. **Reward:** The agent receives a reward $R(s_t, a_t, s_{t+1})$ based on the state transition.
7. **Repetition:** Steps 2 through 6 are repeated, with the agent continually updating its belief state and selecting actions until a terminal condition is reached.

The agent can use reinforcement learning to find the strategy that maximizes the future-discounted cumulative reward: $V(s) = \max_a \{r(s, a) + \gamma \sum_S T(s'|s, a)V(s')\}$. It has been theorized and shown empirically that as long as the POMDP formalism correctly models the task environment and the relevant parts of human cognition, an optimal policy approximates that of human behavior. This is known as *computational rationality* [15].

3.2. Mental Models and Interactive Behavior

Given that the true state s_i is not directly observable, the agent forms its belief b_i, a probability distribution over all possible states in the environment at i. We propose that the agent performs a Bayesian update to obtain $b_i$using its *mental model*, $\hat{t}$:

$$b_{i+1} \propto \hat{t}(b_i, o_i), \quad (1)$$

In Equation 1, the mental model is a (probabilistic) function that updates the agent's belief given observation and previous belief. Thus the mental model $\hat{t}$ can be viewed as the (imperfect) transition function T of an individual agent. An ideal agent with the perfect knowledge and expertise of the interactive environment would have the *true* mental model identical to T. In reality, even given the same observation, agents with different mental models $\hat{t}$ would have different ways to update their beliefs.

3.3. Inferring Mental Models from Observation

We can use the framework in Sections 3.1 and 3.2 to simulate agents with different mental models and use them to generate simulated behavior. When a human user interacts to generate *real* data, it can then be compared to the simulated data to determine the likely mental model of the human user.

Suppose that the mental model has the probability distribution $P(\hat{t})$. From Sections 3.1 and 3.2 we know how an agent with a mental model $\hat{t}$ would behave. Consequently, we also know the conditional probability distribution of $P(D^o \mid \hat{t})$, given an observed behavior data D^o. Bayes' rule can then be used to invert the conditional probability and find:

$$P(\hat{t} \mid D^o) \propto P(D^o \mid \hat{t}) \cdot P(\hat{t}), \tag{2}$$

Finding the likelihood $P(D^o \mid \hat{t})$ is difficult, both analytically and empirically. Instead, we use a likelihood-free Approximate Bayesian Computation (ABC) [16, 17] to sample possible values of $\hat{t}$, minimize the difference between simulated and observed data, estimated by a Gaussian process regression model [18] and find the posterior distribution.

3.4. Detecting Changes of Mental Models

Equation 2 gives us a probabilistic estimate of mental models, which alone is insufficient in detecting potential changes in mental models. To algorithmically determine whether, given observed data, the mental model has changed significantly, we need to quantify changes in the posterior distribution $P(\hat{t} \mid D^o)$. Depending on the specificities of the interaction, we can choose from various methods, as summarized in Table 1.

3.4.1. Example: Mental Models with Categorical Values

Which quantification method to use depends on the characteristics of mental models. Suppose we have a categorical mental model, which is the case we could use *maximum a posteriori estimate* (MAP) to determine the values of $\hat{t}$, and detect any changes.

1. **Calculate Posterior Distribution**
   For each category c in the mental model categories C (i.e. Equation 2):

   $$\text{Posterior}[c] = \text{Likelihood}[c] \times \text{Prior}[c]$$

   Normalize the Posterior for each category c by dividing by the sum of all Posterior values:

   $$\text{Posterior}[c] \leftarrow \frac{\text{Posterior}[c]}{\sum_{c' \in C} \text{Posterior}[c']}$$

Category	**Description**
Measurement	Utilize statistical distance measures (e.g., KL divergence, Total Variation distance, Wasserstein distance) to quantify the difference between successive posterior distributions of the mental model ($P(\hat{t} \mid D^o)$) to assess how one distribution diverges from another.
Threshold	Define a threshold for a significant change, based on domain knowledge, statistical criteria, or adaptive methods. Validate this threshold through simulations or historical data to ensure it effectively differentiates between routine updates and significant model changes.
Monitoring	Continuously or periodically calculating the distance measure between the current and previous posterior distributions, storing past distributions for comparison. If the distance exceeds the threshold, infer a significant change in the mental model has occurred.

Table 1
Different Algorithmic Approaches to Infer Changes in an Agent's Mental Model

2. **Identify MAP Estimate**
   Determine the category c_{MAP} with the highest Posterior probability:

   $$c_{\text{MAP}} = \arg\max_{c \in C} \text{Posterior}[c]$$

3. **Decide the Value of the Mental Model**
   Update the value of the mental model with the MAP estimate:

   $$\hat{t} = c_{\text{MAP}}$$

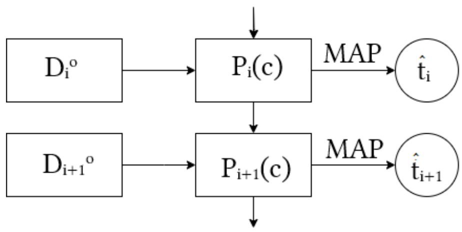

Figure 3: Quantifying the value of mental model with MAP. The estimated value is calculated from the posterior probability distribution and updated every term.

The pipeline for inferring and quantifying the value of mental models is shown in Figure 3. Here, the current distribution $P_i(c)$ of mental models is used as priors and observation to produce the next period's distribution. From each distribution, the mental model's value is determined using MAP.

4. Evaluation

We use an experiment to demonstrate how the framework outlined in Section 3 quantifies and detects changes in the latent mental models of human participants interacting with an interactive system. We change the instructions given to the participants during the experiment to mimic changes in mental models and showcase how the model prediction successfully reflects these changes.

4.1. Participants

We recruited 10 participants online[1], of which 8 identified as females, and 2 as males, coming from 5 different nations. They are between the ages of 20 and 48, averaging at 29. The participants were paid compensation for taking part in the experiment.

4.2. Materials

We conducted our experiment remotely using a webpage designed to simulate a hypothetical scenario where participants interact with the simulation environment and make decisions based on feedback and prior instructions. Participants interact by clicking buttons which are logged as experiment data.

Scenario Picture a warehouse of unmarked boxes containing electric and electronic waste, including *used batteries*, *LED lights*, and *household appliances*. To identify what each box contains, there's an advanced scanner equipped with ultrasound, X-ray tomography, magnetic resonance imaging (MRI), and radio frequency sensors. The warehouse manager can select a sensor to scan a box and get specific results. Each type of waste generates unique readings on the sensors. By scanning a box, the manager aims to determine its specific contents. Specifically, each waste has four features: ultrasound, x-ray, MRI, and radio frequency. Each feature value can be either *high* or *low*.

The scenario is represented on a webpage, and the participants play the role of warehouse manager. In each task, the participant is presented with a box of unknown contents, and given a goal of finding particular contents. The participant must scan the box for the four features and decide whether to open the box or abandon it, given their mental model of what contents produce what sorts of scanner readings, and what their goal is.

4.3. Experiment Procedure

The experiment is carried out as follows:

- Each participant performs 12 rounds of tasks.
- During each round, the webpage refreshes and randomly generates a box as described above.
- During each round, each participant is randomly assigned a type of waste to look for.
- The participant scans the box, and decides whether to accept or reject it.

Each participant is rewarded points for accepting the box containing the assigned waste or rejecting the box not containing it. If a participant wrongly accepts or rejects a box, a penalty is applied. Scanning a feature will also cost points. Therefore participants are instructed to act economically to make the right decision with minimal costs.

The 10 participants are divided into 2 groups of 5. In round 1, we give each group a table containing the probability of finding each waste given a set of features.

- Group 1: the probabilities of finding each waste given all features except *radio frequency*;
- Group 2: the probabilities of finding each waste given all features except *MRI*.

After round 5, all participants are given a new table containing the probabilities of finding each waste given all features, with no features withheld. These tables represent the participants' mental models ($\hat{t}$ in our computational model). The mental models of the initial 5 rounds belong to those participants not having learned to associate certain features with the underlying probabilities. We assign $\hat{t}_1$ to the initial mental model of Group 1, and $\hat{t}_2$ to that of Group 2. The new mental model assigned after round 5 is $\hat{t}_0$.

ult.	x-ray	MRI	radio	batt.	lights	app
high	high	high	-	0.7	0.3	0.6

Table 2
A snippet of the table shown to Group 1

A snippet of the table given to Group 1 is shown in Table 2. Using this knowledge, if a participant obtains the corresponding readings, they would know that the likelihood of finding a battery is 0.65. Taking into consideration the action costs, they can calculate the expected reward and decide whether they would accept the box.

The switch at round 5 is designed to model users acquiring a new mental model during an interaction after gaining knowledge and expertise about the environment and correctly associating all features with the probabilities.

Summary Statistics The experiment data gathered are the sequence of actions performed by each participant, recorded as a list of button IDs. To eliminate unnecessary randomness, we transform the data using *summary statistics*: we ignore any repetitions in the action and its order. As a result, we are only concerned with whether each sensor has been used, and whether the participant decides to accept or reject the box.

Inference 10 participants each performed 12 tasks to generate 12 results of button clicks. In total 60 sequences are collected and transformed by summary statistics into sets of boolean variables. Each result records the status of the 6 buttons, with 1 corresponding to the button being clicked, and 0 otherwise. For example, if a participant chooses to scan the X-ray and MRI, and rejects the box, the resultant data would be: $[1, 2, 3]$, and transformed into $[0, 1, 1, 0, 1, 0]$.

As described in Section 3.4, the mental model c can be quantified as a categorical variable. We divide the unit interval into thirds so that each third corresponds to one of three mental models $\hat{t}_0$, $\hat{t}_1$ and $\hat{t}_2$. We create *simulated agents* with the three mental models to produce *simulated data*. For each $\hat{t}$, we use Proximal Policy Optimization with the default parameters [19] to train the simulated agents.

Using the mechanism in Section 3, our model samples possible values of c and compares the simulated results with participant data to produce a probabilistic distribution of c values. We use MAP estimates to determine their values, as outlined in Section 3.4. For each round of tasks each participant performs, we sample the corresponding simulated result 200 times.

4.4. Experiment Result

We can calculate the accuracy of our inference: the percentage of the 200 inferred c that matches the correct mental

[1] www.prolific.co

model $\hat{t}_i, i = 0, 1, 2$. Averaged over all participants, we thus obtained 12 average prediction accuracies throughout the iteration. The result is presented in Figure 4.

We plot the results in Figure 4. We observe the model's average prediction accuracy for each participant's mental model across the 12 rounds. The red, vertical dotted line marks the switching of $\hat{t}$ as participants receive the new table after round 5.

Furthermore, we also calculate the standard deviation of the inferred values of mental models for each round, and average over all participants. The result is shown in the Figure 5. The switching of $\hat{t}$ is also marked by a red, vertical dotted line.

4.5. Discussion

We can discover several trends in the results as shown in Figures 4 and 5. The accuracy of model prediction of mental model $\hat{t}$ increases per round (Figure 4). This is due to the Bayesian update of the model incorporating the results from previous rounds into the following rounds as prior information. Consequently, the inference improves in accuracy as confoundments are gradually resolved. This is also shown in the decrease of standard deviations in Figure 5. In earlier rounds, there is relatively little information and more confoundments, leading to greater uncertainty in inference results. As evidence accumulates and confoundments are resolved, uncertainty also decreases.

Importantly, both figures show a drastic change between rounds 5 and 6, when the mental models $\hat{t}$ are switched. The accuracy goes down and the standard deviation slightly increases. This means that at round 6, the priors from previous rounds still have a strong influence on the inference results, and the model clings to the prediction that the data were produced by agents with the old mental model (either $\hat{t}_1$ or $\hat{t}_2$). However, as can be seen in Figures 4 and 5, evidence accumulates due to our model's Bayesian setup, suggesting that a new mental model was likely behind the observed data. Towards the later rounds, accuracy has recovered and the model now firmly predicts the new mental model $\hat{t}_0$. Similar trends can also be observed in average standard deviations, as the value goes up slightly after round 5 before continuing to descend.

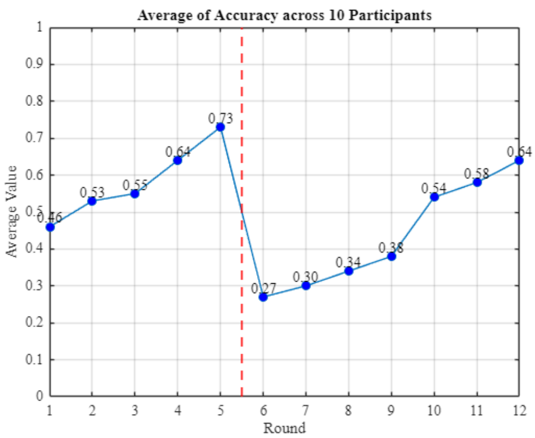

Figure 4: Average accuracy of model prediction of all 10 participants across 12 rounds. The red dotted line indicates the switching of instructions.

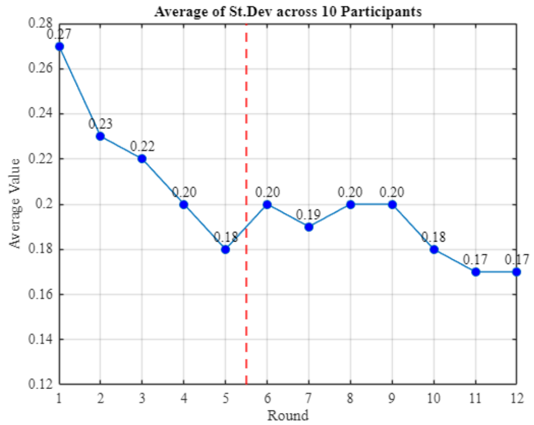

Figure 5: Standard deviation of the mental models across 12 rounds. The red dotted line indicates the switching of instructions.

5. Future Research

In this paper, we present a formal, computational model to infer a user's mental model during interaction. It can detect changes in the mental model and dynamically updates the inference once sufficient evidence is accumulated. The experiment demonstrates a consistent trend of improving accuracy and decreasing variance in the model predictions. The model can be a starting point for building an intelligent interactive system that truly understands its users.

Currently, the model needs to run ABC and sample at each round of inference, as outlined in section 3. This makes the model too slow to be implemented in real applications. Consequently, a key improvement would be to make the model more lightweight and efficient so that inferences and adaptations can be implemented in real-time. One idea worth exploring is amortizing the inference by pre-training the model using simulation [20].

The entire inference framework must also be tested with real HCI tasks, such as menu search and typing. To do so we need to define both the computational model of interaction and the mental model. This would also allow us to compare our proposed approach to existing methods and conduct statistical analysis with more participants. To do so would likely require insights from psychology, behavioral science, etc., and is beyond the scope of this work.

Acknowledgments

This research has been supported by the Academy of Finland (grant 330347).

References

[1] S. J. Payne, Mental models in human-computer interaction, The Human-Computer Interaction Handbook (2007) 89–102.

[2] N. Banovic, T. Buzali, F. Chevalier, J. Mankoff, A. K. Dey, Modeling and understanding human routine behavior, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, 2016, pp. 248–260.

[3] H. Rutjes, M. Willemsen, W. IJsselsteijn, Considerations on explainable ai and users' mental models, in:

CHI 2019 Workshop: Where is the Human? Bridging the Gap Between AI and HCI, Association for Computing Machinery, Inc, 2019.
[4] M.-A. Storey, F. D. Fracchia, H. A. Müller, Cognitive design elements to support the construction of a mental model during software exploration, Journal of Systems and Software 44 (1999) 171–185.
[5] S. Dumais, R. Jeffries, D. M. Russell, D. Tang, J. Teevan, Understanding user behavior through log data and analysis, Ways of Knowing in HCI (2014) 349–372.
[6] P. Langley, User modeling in adaptive interface, in: UM99 User Modeling: Proceedings of the Seventh International Conference, Springer, 1999, pp. 357–370.
[7] P. Legrenzi, V. Girotto, Mental models in reasoning and decision-making processes, in: Mental models in cognitive science, Psychology Press, 2013, pp. 95–118.
[8] A. Howes, J. P. Jokinen, A. Oulasvirta, Towards machines that understand people, AI Magazine 44 (2023) 312–327.
[9] Z. Duric, W. D. Gray, R. Heishman, F. Li, A. Rosenfeld, M. J. Schoelles, C. Schunn, H. Wechsler, Integrating perceptual and cognitive modeling for adaptive and intelligent human-computer interaction, Proceedings of the IEEE 90 (2002) 1272–1289.
[10] G. Bailly, A. Oulasvirta, D. P. Brumby, A. Howes, Model of visual search and selection time in linear menus, in: Proceedings of the sigchi conference on human factors in computing systems, 2014, pp. 3865–3874.
[11] X. Chen, G. Bailly, D. P. Brumby, A. Oulasvirta, A. Howes, The emergence of interactive behavior: A model of rational menu search, in: Proceedings of the 33rd annual ACM conference on human factors in computing systems, 2015, pp. 4217–4226.
[12] S. Sarcar, J. P. Jokinen, A. Oulasvirta, Z. Wang, C. Silpasuwanchai, X. Ren, Ability-based optimization of touchscreen interactions, IEEE Pervasive Computing 17 (2018) 15–26.
[13] A. Kangasrääsiö, J. P. Jokinen, A. Oulasvirta, A. Howes, S. Kaski, Parameter inference for computational cognitive models with approximate bayesian computation, Cognitive science 43 (2019) e12738.
[14] M. T. Spaan, Partially observable markov decision processes, in: Reinforcement learning: State-of-the-art, Springer, 2012, pp. 387–414.
[15] A. Oulasvirta, J. P. Jokinen, A. Howes, Computational rationality as a theory of interaction, in: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, 2022, pp. 1–14.
[16] M. A. Beaumont, W. Zhang, D. J. Balding, Approximate bayesian computation in population genetics, Genetics 162 (2002) 2025–2035.
[17] K. Csilléry, M. G. Blum, O. E. Gaggiotti, O. François, Approximate bayesian computation (abc) in practice, Trends in ecology & evolution 25 (2010) 410–418.
[18] J. Lintusaari, H. Vuollekoski, A. Kangasrääsiö, K. Skytén, M. Järvenpää, P. Marttinen, M. U. Gutmann, A. Vehtari, J. Corander, S. Kaski, Elfi: Engine for likelihood-free inference, Journal of Machine Learning Research 19 (2018) 1–7.
[19] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 (2017).
[20] H.-S. Moon, A. Oulasvirta, B. Lee, Amortized inference with user simulations, in: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, 2023, pp. 1–20.