frontiers in Computational Neuroscience

ORIGINAL RESEARCH
published: 08 September 2017
doi: 10.3389/fncom.2017.00080

Modeling Search Behaviors during the Acquisition of Expertise in a Sequential Decision-Making Task

Cristóbal Moënne-Loccoz[1], Rodrigo C. Vergara[2], Vladimir López[3,4], Domingo Mery[1] and Diego Cosmelli[3,4]**

[1] Department of Computer Science, School of Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile, [2] Facultad de Medicina, Biomedical Neuroscience Institute, Universidad de Chile, Santiago, Chile, [3] Center for Interdisciplinary Neuroscience, Pontificia Universidad Católica de Chile, Santiago, Chile, [4] School of Psychology, Pontificia Universidad Católica de Chile, Santiago, Chile

Our daily interaction with the world is plagued of situations in which we develop expertise through self-motivated repetition of the same task. In many of these interactions, and especially when dealing with computer and machine interfaces, we must deal with sequences of decisions and actions. For instance, when drawing cash from an ATM machine, choices are presented in a step-by-step fashion and a specific sequence of choices must be performed in order to produce the expected outcome. But, as we become experts in the use of such interfaces, is it possible to identify specific search and learning strategies? And if so, can we use this information to predict future actions? In addition to better understanding the cognitive processes underlying sequential decision making, this could allow building adaptive interfaces that can facilitate interaction at different moments of the learning curve. Here we tackle the question of modeling sequential decision-making behavior in a simple human-computer interface that instantiates a 4-level binary decision tree (BDT) task. We record behavioral data from voluntary participants while they attempt to solve the task. Using a Hidden Markov Model-based approach that capitalizes on the hierarchical structure of behavior, we then model their performance during the interaction. Our results show that partitioning the problem space into a small set of hierarchically related stereotyped strategies can potentially capture a host of individual decision making policies. This allows us to follow how participants learn and develop expertise in the use of the interface. Moreover, using a Mixture of Experts based on these stereotyped strategies, the model is able to predict the behavior of participants that master the task.

Keywords: sequential decision-making, Hidden Markov Models, expertise acquisition, behavioral modeling, search strategies

Edited by:
Guenther Palm,
University of Ulm, Germany

Reviewed by:
Subramanian Ramamoorthy,
University of Edinburgh,
United Kingdom
Pablo Varona,
Universidad Autonoma de Madrid,
Spain

****Correspondence:***
Cristóbal Moënne-Loccoz
cmmoenne@uc.cl
Diego Cosmelli
dcosmelli@uc.cl

Received: *27 January 2017*
Accepted: *04 August 2017*
Published: *08 September 2017*

Citation:
Moënne-Loccoz C, Vergara RC, López V, Mery D and Cosmelli D (2017) Modeling Search Behaviors during the Acquisition of Expertise in a Sequential Decision-Making Task. Front. Comput. Neurosci. 11:80. doi: 10.3389/fncom.2017.00080

1. INTRODUCTION

Whether you are preparing breakfast or choosing a web link to click on, decision making processes in daily life usually involve sequences of actions that are highly dependent on prior experience. Consider what happens when you interact with an ATM machine: you have to go through a series of specific button presses (i.e., actions) that depend on whether you are interested in, for instance, drawing money or consulting your account balance (i.e., the outcome). Despite some

commonalities, which sequence you use will depend on the specific ATM brand you are dealing with, while previous exposure will determine which behavioral strategy you deploy. Maybe you cautiously explore the available choices and hesitate before pressing each button; maybe this is the same machine you have used for the last year, so you deftly execute a well practiced sequence of actions to draw some cash.

Sequential choice situations such as the ATM example are pervasive in everyday behavior. Not surprisingly, its importance for the understanding of human decision making in real-world scenarios has been recognized for long time, as a wealth of studies attest (Rabinovich et al., 2006; Gershman et al., 2014; Otto et al., 2014; Cushman and Morris, 2015; see Walsh and Anderson, 2014 for review). Yet, despite its importance, the issue of how expertise is developed in the context of sequential choice situations remains still under-explored. While the study of optimal strategies or courses of action to solve sequential choice scenarios is a fundamental aim of such studies (Alagoz et al., 2009; Friedel et al., 2014; Schulte et al., 2014; Sepahvand et al., 2014), it is still necessary to better understand how agents learn and acquire such strategies as they interact with the world (Fu and Anderson, 2006; Acuña and Schrater, 2010; Sims et al., 2013).

Human-computer interfaces offer a privileged scenario to study the development of expertise in sequential decision making processes. As we learn to use an interface, isolated exploratory actions turn into expert goal-directed sequences of actions by repeatedly testing and learning to adjust behavior according to the outcome (Solway and Botvinick, 2012). Furthermore, such scenarios are particularly well adapted to sampling behavioral data in varying degrees of ecological validity. They also allow for testing different ways to use such behavioral data to make the interface more responsive. Indeed, in addition to contributing to a better understanding of the psychological and neurobiological mechanisms underlying sequential decision making, taking into account how people learn to interact with novel systems could have relevant consequences for adaptive interface design. Here we tackle the question of expertise acquisition in a sequential decision making scenario. We aim to discover if individuals can be described in terms of specific behavioral strategies while they learn to solve the task and, if so, whether this can be used to predict future choices.

Several approaches have been developed to address the problem of modeling behavior in sequential decision making scenarios. The Reinforcement Learning (RL) paradigm has been successfully extended to model behavior during sequential choice tasks (Dayan and Niv, 2008; Acuña and Schrater, 2010; Dezfouli and Balleine, 2013; Daw, 2014; Walsh and Anderson, 2014). In general terms, RL techniques aim at finding a set of rules that represent an agent's policy of action given a current state and a future goal by maximizing cumulative reward. Because actions are chosen in order to maximize reward, it is necessary to assign value to the agent's actions. Reward schemes work well when gains or losses can be estimated (e.g., monetary reward). However, in many of our everyday interactions, reward in such an absolute sense is difficult to quantify. Accordingly, the accuracy of an arbitrary reward function could range from perfect guidance to totally misleading (Laud, 2004).

To overcome the difficulty of defining a reward function, the Inverse Reinforcement Learning based approaches try to recover a reward function from execution traces of an agent (Abbeel and Ng, 2004). Interestingly, this technique has been used to infer human goals (Baker et al., 2009), developing into methods based on Theory of Mind to infer peoples' behavior through Partially Observable Markov Decision Process (POMDP) (Baker et al., 2011). Although, these techniques are important steps in the area of plan recognition, they usually focus on which is the best action an agent can take given a current state, rather than determine high-level patterns of behavior. Also, they consider rational agents who think optimally. However, in learning scenarios, optimal thinking is achieved through trial and error, rather than being the de facto policy of action.

Alternative modeling approaches exist that do not require defining a reward function to determine behaviors. Specifically, Markov models have been adapted to analyze patterns of behavior in computer interfaces, such as in web page navigational processes (Ghezzi et al., 2014; Singer et al., 2014), were no simple, unitary reward is identifiable. In general terms, Markov models are aimed at modeling the stochastic dynamics of a system which undergoes transitions from one state to another, assuming that the future state of the system depends only on the current state (Markov property). For example, a Markov chain of navigational process can be modeled by states representing content pages and state transitions representing the probability of going from one page to another (e.g., going from the login page to the mail contacts or to the inbox). Possible behaviors or cases of use can then be extrapolated from the structure of the model (see Singer et al., 2014 for an example). In general, however, these behaviors are highly simplified descriptions of the decision-making process because they do not consider the rationale behind the user's actions, and only focus on whether the behavioral pattern is frequent or not. Accordingly, if the user scrolls down the page searching for a specific item in some order and then makes a decision, the psychological processes behind his or her actions are ignored.

An interesting extension of the simpler Markov models, which aims to capture the processes underlying decision making behavior, are the Hidden Markov Models (HMM) (Rabiner, 1989). In a HMM, the states are only partially observable and the nature of the underlying process is inferred only through its outcomes. This relationship between states and outcomes allows modeling a diversity of problems, including the characterization of psychological and behavioral data (Visser et al., 2002; Duffin et al., 2014). Of special interest is the use of HMMs to model the strategic use of a computer game interface (Mariano et al., 2015). Using sets of HMMs, Mariano and collaborators analyze software activity logs in order to extrapolate different heuristics used by subjects while they discover the game's rules. Such heuristics, which are extrapolated a posteriori, are represented by hidden states composed by the grouping of actions and the time taken to trigger such actions. These are then used to identify patterns of exploratory behavior and behaviors representative of the mastery of the game. Interestingly, such heuristics show a good adjustment with self-reported strategies used by the participants throughout the task (Mariano et al., 2015).

Generally speaking, however, Markov models use individual actions to represent hidden states (such as click this or that icon) and more complex high-level behavioral heuristics (such as policies of action) are only inferred after the experimental situation or setup (see Mariano et al., 2015). This represents a potential limitation if we are to build interfaces that are responsive to the learning process as it unfolds. Indeed, throughout the acquisition of different skills, there is abundant evidence that humans and other animals rely on grouping individual actions into more complex behavioral strategies such as action programs or modules to achieve a certain goal (Marken, 1986; Manoel et al., 2002; Matsuzaka et al., 2007; Rosenbaum et al., 2007). This can happen while individual actions remain essentially unchanged and only the way they are organized changes. For instance, it has been shown that throughout the development of sensorimotor coordination, individual movements progressively become grouped into sequences of movements, as the child becomes adept at controlling goal-directed actions (Bruner, 1973; Fischer, 1980).

If expertise development and skill acquisition implies the hierarchical organization of individual actions into complex, high-level behavioral strategies, taking this into account could represent a relevant line of development for behavioral modeling. Recently, this hierarchical structure of behavior has been considered in some approaches, such as Hierarchical Reinforcement Learning (HRL) (Botvinick, 2012; Botvinick and Weinstein, 2014). The idea behind HRLs is to expand the set of actions available to an agent to include a set of extended high-level subroutines which coordinate multiple low-level simple actions that otherwise the agent would have to execute individually. An interesting consequence of this is that it could potentially aid in reducing the dimensionality of the problem, a critical issue in behavioral modeling (Doya and Samejima, 2002; Botvinick, 2012; Shteingart and Loewenstein, 2014). Furthermore, if flexibility in membership between low-level actions and behaviors exist, modeling behaviors in a probabilistic way could help in better capturing the dynamical nature of expertise acquisition.

In this work we are interested in modeling the behavior of users confronted with a sequential decision-making task with limited feedback, in a way that sheds light into potentially relevant learning strategies. While our short term goal is to be able to consistently reproduce search behaviors that individual users exhibit, our long term goal is to inform the construction of interfaces that can adapt to different learning curves. In the following Method Section, we will first present the overall rationale for the approach. We will then describe the experimental task used to test the model and the characteristics of the participants. The bulk of the section is then dedicated to presenting the modeling framework in detail. Results are then presented mainly in terms of the performance of the model for different types of participants as well as in terms of its capacity to predict their behavior. We end with a discussion of our approach as well as its limitations and potential further developments.

2. METHODS

2.1. Rationale of the Approach

We present here a HMM-based approach that capitalizes on the hierarchical structure of behavior to model the performance of individuals as they develop expertise in a sequential decision making task that is structured as a 4-level Binary Decision Tree (BDT). We use the HMM structure to infer the distribution of probabilities of a modular set of pre-defined stereotyped high-level strategies (represented by hidden states), while observing the outcome of the user's actions. As such, this approach is reminiscent of type-based methods where one focuses on a pre-specified group of behaviors among all possible behaviors (Albrecht et al., 2016). Such "types" of behavior can be hypothesized based on previous knowledge of the interaction or on the structure of the problem. In our case, pre-defined strategies (i.e., decision-making policies) are selected in order to cover a series of increasingly efficient behaviors in the context of the BDT: from random unstructured exploration to goal-directed actions driven by feedback and knowledge of the task's structure. Additionally, this approach allows us to use the modular architecture as a Mixture of Experts (see Doya and Samejima, 2002 for a similar idea). This enables us to model the evolution of the user's behavior while simultaneously asking the experts about the user's most probable next choice. As a consequence, it is possible to evaluate the model also in terms of it's capacity to predict future actions, which would be desirable in the context of adaptive interface design.

We test this approach using a simple computational interface game where an underlying concept-icon mapping must be discovered in order to complete the task. As mentioned above, the game is structured as a four-level BDT with limited feedback (see **Figure 1**; see also Fu and Anderson, 2006, p. 198 for a structurally similar design). Each node of the tree represents a decision, and the links between nodes represent the consequences of each decision. The depth of the tree represents the number of sequential choices that are needed to achieve a specific goal. This scenario captures sequential decision-making situations with limited feedback that are pervasive in real world human-computer interface interaction. Furthermore, it allows us to follow the development of expertise as the users discover the rules of the game and deploy different search strategies to solve the task.

2.2. Task Structure

Figure 1 presents a schematic depiction of the task interface structure that instantiates the BDT. Participants were presented with a computer screen that had an instruction of the type "$verb_{L1}$ with a $noun_{L2}$ an $adjective_{L3}$ $noun_{L4}$" (i.e., *"Feed a cat with a small fish,"* see inset in **Figure 1**). Note that because the task was performed by native Spanish speakers, the the final adjective-noun pair is inverted regarding the previous instruction example and would read *"Alimenta un gato con un pez pequeño."* After clicking anywhere on this first screen, participants were confronted with the first binary choice (*level 1* of the BDT) with two icons located 2.4° to each side of a central fixation spot. Each of the variable words in the instruction had a fixed mapping to an

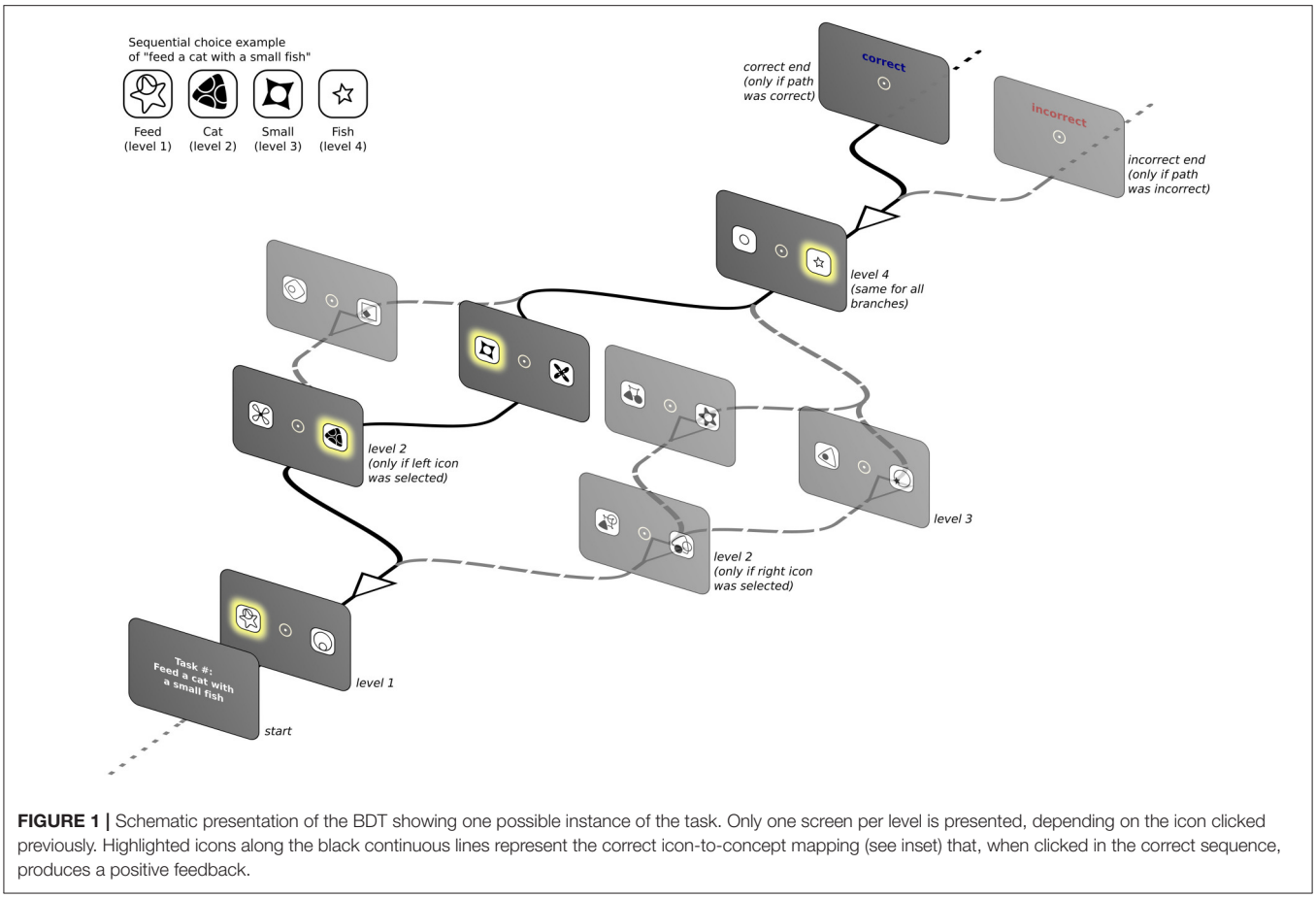

FIGURE 1 | Schematic presentation of the BDT showing one possible instance of the task. Only one screen per level is presented, depending on the icon clicked previously. Highlighted icons along the black continuous lines represent the correct icon-to-concept mapping (see inset) that, when clicked in the correct sequence, produces a positive feedback.

abstract icon and level of the BDT (as indexed by subscripts in the instruction example above), but participants were not informed of this fact. They were instructed to click using the computer mouse on one of the two icons to proceed to the next screen where the next set of two icons was presented (*level 2* of the BDT). The overall arrangement of the icons remained constant and only their identity changed according to the specific task mapping. This branching structure was repeated until level 4 was reached. The last level was always the same regardless of the branch because only two possible adjectives were used: "small (pequeño)" or "large (grande)." Once subjects clicked an icon in level 4 they received feedback informing them whether the path they had chosen was correct. In the case of a negative feedback, subjects had no way of knowing, a priori, at which level they had made the wrong choice. Therefore, they had to discover the mapping based exclusively on their exposure to successive iterations of the task and the feedback received at the end of each chosen path. Each instruction (corresponding to a single task instance) was presented repeatedly until the participant was able to choose the correct path 5 times in a row before moving onto the next possible instruction. After 15 successive wrong answers, participants were asked whether they wanted to move onto the next task. If they refused, after 10 further wrong answers they were asked again. If they decided to persist in the same task, they only had 5 additional chances to find the correct path else they were forced to move onto the next one. The maximum time allowed to complete the entire task was set to 40 min.

2.3. Participants

Twenty-two participants were recruited (12 females) of ages ranging from 20 to 32 years old, with a mean age of 26 $\pm$ 3 years (mean $\pm$ *SD*). All participants reported normal or corrected-to-normal vision and no background of neurologic or psychiatric conditions. Participants performed on average 162 $\pm$ 51 trials (mean $\pm$ *SD*) during the course of the experiment (range 89–325). Trials are defined as repetitions of an instance of the task, from the instruction to the feedback after the sequence of choices. Each task instance contains on average 10.79 $\pm$ 1.47 trials (mean $\pm$ *SD*).

The study was approved by the Ethics Committee of the School of Psychology of the Faculty of Social Sciences, Pontificia Universidad Católica de Chile. All participants gave written informed consent. Twenty-two subjects participated in this study and the nature of the task was explained to all upon arrival to the Laboratory. All experiments were performed in the Psychophysiology Lab of the School of Psychology of the same University. Participants sat in a dimly illuminated room, 60 cm away from a 19-inch computer screen with a standard computer mouse in their right hand. All participants were right handed. Prior to staring the task, subjects were fitted with a 32-electrode

Biosemi ActiveTwo © digital electroencephalographic (EEG) system, including 4 electrooculographic (EOG) electrodes, two of them placed in the outer canthi of each eye and two above and below the right eye. Continuous EEG was acquired at 2,048 Hz and saved for posterior analysis. Throughout the task, participants were instructed to maintain fixation on a central spot in order to avoid eye-movement related artifacts in the EEG data. We do not report here the analysis of electrophysiological recordings and will limit ourselves to the behavioral data. All stimuli were presented around fixation or 2.4° to each side of the central fixation spot. This ensured that, despite the instruction to maintain fixation, all participants could easily see all stimuli and perform the task without difficulty in perceptual terms.

2.4. Modeling Framework

2.4.1. Low-level Actions vs. Strategies

The simplest action that can be taken on the BDT is to click one of the two possible icons of the binary choice. We will therefore consider these two as the only low level actions for the model. What such low level actions mean or represent, in terms of learning of the BDT, depends on whether the participant is using them in some systematic way to obtain positive feedback (i.e., a strategy). We consider such systematic combination of low-level actions the high-level strategies of the model.

Different strategies can be used to explore and learn the structure of the BDT. In the following we model four high-level decision-making policies in the form of well- defined search strategies, that account for increasingly sophisticated ways to solve the task:

1. **Random Search Strategy:** If the participant displays no systematic use of low level actions, we label this as random behavior. In other words, overt actions seem to be unrelated to the task's demands so that we can only assume ignorance regarding the underlying decision making strategy.
2. **Spatial Search Strategy:** If the participant shows evidence of acting based exclusively on information regarding the spatial layout of the BDT, regardless of the identity of the presented icons (for instance, by choosing to explore from the leftmost to the rightmost branch), we label this as a spatial behavior. When clicking based on spatial features, the participant iteratively discards paths of the BDT so that complete knowledge can be obtained only when the 16 paths of the BDT are correctly recognized. Accordingly, as paths share common information, the learning curve of a user invested exclusively in this strategy will grow exponentially as the search space becomes smaller.
3. **Generative Search Strategy:** If the participant shows evidence of considering the identity of individual icons to guide her choice of actions to reach positive feedback, we label this as a generative behavior. In other words, it implies a first level of successful mapping between current task instruction and the specific BDT instance that is being explored (i.e., when the participant learns that a given icon means a given concept). When clicking based on generative relationships, the participant discards subtrees of the BDT where it is not possible to reach positive feedback. Complete knowledge of the BDT can be obtained when the 16 icons are correctly mapped. All the generative relationships can be learned by being exposed to positive feedback in the 8 paths that contains all of them. Therefore, the learning curve of this strategy is represented by a sigmoid function.
4. **Discriminative Search Strategy:** Here the participant uses a generative model, but adds the ability to learn and relate the negative form of a concept-icon relationship (i.e., learn A by a generative association and then label the neighbor as not-A). In other words, the discriminative search strategy is one that predicts concepts that have not yet been seen in the scope of positive feedback. Concepts are deduced from the context and the understanding of the rules of how the interface works. When clicking based on discriminative relationships, the participant can prune the BDT subtrees more aggressively to obtain positive feedback. As in the generative case, complete knowledge of the BDT can be obtained when the 16 icons are correctly mapped. However, all discriminative relationships can now be learned by being exposed to positive feedback in the 4 paths that contains all of them. Accordingly, the learning curve of this strategy is represented by a sigmoid function that is steeper than in the generative case.

Each of the above models has an initial domain of action that corresponds to the set of actions that can be performed on the BDT according to the strategy's rules. Once exploration of the interface is underway, the initial domain of action of each strategy will necessarily change. This can happen because the user learns something about the specific task instance he is currently solving, or because he learns something about the overall structure of the interface. It is therefore necessary to define criteria that, according to each strategy's rules, allow one to update their domain of action depending on local (task-instance) and global (task-structure) knowledge. Local updates criteria will coincide with the rules of the spatial strategy for all systematic strategies, because according to our hierarchical definition, the simplest way to discard places of the BDT systematically is using spatial information. Conversely, for global updates—and for the sake of simplicity—we will define knowledge in terms of optimal behavior, (i.e., learning places or concepts of the task instances by repeating the correct path 5 times in a row). **Table 1** presents the formal definition of the domain of action D_i and both updating schemes for each strategy s_i.

To track the BDT knowledge of the specific strategy s_i, we define α_i as a measure of what still needs to be mapped (place or concept) of the BDT at a given time t :

$$\alpha_i(t) = 1 - \frac{|R_i(t)|}{|G_i(t)|} \tag{1}$$

where $R_i(t)$ is the set of learned choices, $G_i(t)$ is the set of all distinct choices in the strategy's domain, and $|\cdot|$ is the number of elements of a given set. This parameter is the complement of the specific learning curve of each strategy. Specifically, $\alpha_i = 1$ indicates complete lack of knowledge about the interface, and $\alpha_i = 0$ indicates full knowledge.

Note that high-level strategies evolve according to the user's iterative interactions with the task. For instance, if the participant

TABLE 1 | Formal definitions of the strategies are presented according to their initial domain, domain of action updates (task instance learning), and knowledge updates (task structure learning).

Strategy (*i* index)	Rule/Definition	Description
Initialization of the domain of action (defines D_i for a given task instance starting at time t' and target path Q^*)		
Random	$D_i(t) = \{p \mid p \in G_i(t)\}$	The random strategy does not consider learning, thus its domain of action is open to all BDT locations p at any time t.
Spatial *Generative* *Discriminative*	$D_i(t = t') = \{p \mid p \notin R_i(t') \vee p \in Q^*\}$	At the begining of a given task, valid actions can be taken only in unexplored locations p or those belonging to the target path Q^*.
Generative *Discriminative*	$\forall c \mid c \in R_i(t') \wedge c \in Q^*$ $\rightarrow$ subtree(neighbor(c)) $\notin D_i(t')$	Additionaly, discard tree branches by using previous known c concep-icon relationships. subtree(c) is a function that yields c and the set of all locations below c, and neighbor(p) yields the neighbor of a given location/icon p.
Domain of action updates (updates D_i while searching for correct feedback in path Q_t at time t)		
Spatial *Generative* *Discriminative*	$\forall p \mid \text{leaf}(p) \wedge p \in Q_t \rightarrow p \notin D_i(t+1)$	The final selection of a path Q_t is always out of domain in the next iteration. leaf(p) is a function that is true for each BDT leaf (any 4th level location/icon).
	$\forall p \mid p \in Q_t \wedge \text{neighbor}(p) \notin D_i(t)$ $\rightarrow p, \text{parent}(p) \notin D_i(t+1)$	Bottom-up rule to discard parent nodes of Q_t when they have already being explore. parent(p) is a function that yields the location/icon in the level immediately above which leads to p.
Knowledge updates (updates R_i when learning the task instance of target path Q^* at time t)		
Spatial	$\forall p \mid \text{leaf}(p) \wedge p \in Q^* \rightarrow p \in R_i(t+1)$	The final selection of a learned path Q^* is always in the learned set of upcoming tasks.
	$\forall p \mid p \in Q^* \wedge \text{neighbor}(p) \in R_i(t)$ $\rightarrow p, \text{parent}(p) \in R_i(t+1)$	Bottom-up rule to set parent nodes of Q^* as learned.
Generative *Discriminative*	$\forall c \mid c \in Q_a^* \cap Q_b^* \wedge c \notin \{Q_j \mid t_a^* < j < t'_b\}$ $\wedge\, c \in \{Q_j \mid t'_b \leq j \leq t\} \rightarrow c \in R_i(t+1)$	Concept c is considered learned if there are no selection mistakes between two tasks a and b which intersect in that concept. Task a is learned at time t_a^*, and task b starts at time t'_b.

does not show evidence of learning any path, the learning curve of each strategy is a straight constant line at $\alpha_i = 1$. When feedback becomes available (i.e., when the participant reaches the end of a path producing either a correct or incorrect answer), we ask each model how such observation changes or violates its expected probabilities regarding the nature of future feedback. As long as no learning is involved, all active models will answer equally to this query. However, as evidence of learning becomes available, each model will restrict the domain of possible future actions that are consistent with what the model predicts the participant's knowledge should be. A spatial model will label as a mistake any repetition of a path that previously gave positive feedback in the context of a different instruction. A generative model will label as mistakes actions that are inconsistent with a successful icon-concept mapping for which there is prior evidence. The discriminative model inherits the restrictions imposed by the generative model, but will also consider mistakes as those actions that do not take into account not-A type knowledge that the participant should have, given the history of feedback. An important consequence of the above is that strategies can yield the probability of clicking a given icon of the BDT without further training or modeling at any moment throughout the task.

It is worth noting that defining all possible strategies to solve the BDT is not necessary. To delimit the knowledge level of the participant, only the lower and higher bounds of the problem must be defined and more strategies in-between will only increase the framework's resolution. In the BDT case used here, the lower bound is necessarily the random strategy. The upper bound is set by the discriminative strategy because it is the best possible strategy to solve the BDT task (i.e., it requires the least exposure to positive feedback). Accordingly, we make the assumption that at any given moment, the user has all strategies at his disposal but that overt behavior is best captured by a weighted mix of them. We call this level the behavioral model of the framework.

2.4.2. Behavioral Modeling

Once the high-level strategies are defined, we turn to modeling the expectation about the use of a specific strategy or a combination of them by the participant. Such

behavioral model is composed by a modular architecture of the four possible strategies, which interact between them in a HMM-like structure. This modular architecture has the advantage of modeling complex strategies as if they were a single abstract state in the behavioral model.

Formally, a HMM is defined by a finite set of hidden states, $s_1, s_2, \ldots, s_n$, and each time a relevant information arises (e.g., feedback) the system moves from one state $s(t) = s_i$ to another $s(t+1) = s_j$ (possibly the same). The transition probabilities, $P(s_i \rightarrow s_j)$, determine the probability of transiting between states: $P(s_i \rightarrow s_j) = P(s(t+1) = s_j | s(t) = s_i)$. The observable information of the process is a finite set of distinct observations, $v_1, v_2, \ldots, v_m$, and a probabilistic function of the states. Accordingly, each observation has an emission probability $P(v_k|s_i)$, $k \in \{1, \ldots, m\}$, of being seen under a state s_i. As the system must start somewhere, it is necessary to define an initial probability distribution of the states: $w_i = P(s(t=0) = s_i)$, $i \in \{1, \ldots, n\}$. Given that the sets of hidden states and observations are defined a priori, the only values to estimate are the initial distribution and the transition and emission probabilities. Each strategy therefore takes the role of a hidden state at the behavioral level, which then yields the probability distribution of each observation for each trial.

2.4.2.1. Emission probabilities

Although, the task can yield positive and negative feedback, an observer can interpret these observations in different ways depending on the situation. While positive feedback is unambiguous (a hit observation), negative feedback can have two different connotations: it is a mistake if, given previous actions, the observer is warranted to assume that the participant should have had the knowledge to avoid performing the action that produced such outcome. Observing such feedback will therefore mean evidence in favor of random behavior. Else, negative feedback is consistent with exploratory search behavior prior to the first positive feedback and, accordingly, not considered a mistake. The set of observations V is therefore defined as: $V = \{\text{mistake, explore, hit}\}$.

Since strategies are sensitive to the context, their emission probabilities change as the participant makes choices. At each sequence step, we calculate the emission probabilities within the subtree of possible future choices. Considering the last choice of the participant as the root of the subtree, we enumerate all possible future paths of actions Q, defining the following sets at step t:

$$H_i(t) = \{Q | (\forall a \in Q)[a \in D_i(t)] \wedge (\exists a \in Q)[a \notin Q^*]\} \quad (2)$$

$$H_i^*(t) = \{Q | (\forall a \in Q)[a \in Q^*]\} \quad (3)$$

where a represents a specific action needed to generate path Q, and Q^* the target path. $H_i(t)$ is the set of all paths that do not violate the strategy's rules, while simultaneously allowing the exploration of available choices. $H_i^*(t)$ is the set of paths that lead to positive feedback. Thus, emission probabilities for strategy s_i are defined as follows:

$$P(V|s(t) = s_i) = \begin{cases} \left\{0, \frac{|H_i(t)|}{|H_i(t)|+|H_i^*(t)|}, \frac{|H_i^*(t)|}{|H_i(t)|+|H_i^*(t)|}\right\}, & \text{if } |H_i(t)| > 0 \vee |H_i^*(t)| > 0 \\ \{1, 0, 0\}, & \text{otherwise} \end{cases} \quad (4)$$

The first case represents emission probabilities for those strategies that, given their rules, allow for future exploration or exploitation. In the second case, when the rules of the strategy cannot explain the current actions, the emission probabilities are fixed to explain mistakes. This is also the case for the random strategy, which is assumed when the observer has no knowledge about the participant's strategy.

2.4.2.2. Transition probabilities

It is possible, but not necessary, that a participant moves progressively through each of the increasingly complex strategies as he learns the structure of the task. Although, such progression may seem as discrete steps (i.e., first using a spatial strategy and then abandoning it altogether when conceptual knowledge becomes available), it is most likely that at any given moment of the task, the participant's strategy will fall somewhere in between, being better represented by a mix of strategy models. This is precisely the distribution captured by the behavioral model. To estimate it, it is necessary to model the interactions between strategies s_i, in terms of transition probabilities and relative weights.

To obtain the transition probabilities we use a voting scheme based on emission probabilities, where each strategy distributes its own $P(V|s_i)$ depending on the ability of the strategy to explain a specific type of observation. Thus, for each observation $v \in V$ we define the set of best explanatory strategies of v as follows:

$$B_v(t) = \{s_i | s_i \in \arg\max_{s_i} P(v|s_i)\} \quad (5)$$

Then, every step t in which the participant performs an action, transition links l_{ij}^t between strategies s_i and s_j gain votes according to the following rules:

$$l_{ij}^t = \sum_{v \in V} P(v|s_i) \begin{cases} 1, & \text{if } s_i \in B_v(t) \wedge i = j \\ \frac{1}{|B_v(t)|}, & \text{if } s_j \in B_v(t) \wedge s_i \notin B_v(t) \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

These rules represent three cases: (a) Any strategy that belongs to $B_v(t)$ strengthens its self-link, not sharing its $P(v|s_i)$ with other strategies. (b) Strategies that do not belong to $B_v(t)$ generate links to those that best explain the observation v, losing their $P(v|s_i)$ in equal parts to those that best explain v. (c) Strategies that do not explain v or do not belong to $B_v(t)$ do not receive votes for observing v.

In the case of the random strategy, its emission probabilities are fixed to $\{1, 0, 0\}$, therefore the above rules do not generate

links with other strategies in the case of exploring or exploiting the BDT knowledge. In order to overcome this limitation, we define the set of strategies that can leave the random strategy as those that do not see the current action as a mistake, plus the random strategy itself:

$$U(t) = \{s_i | P(\text{mistake}|s_i) = 0\} \cup \{s_{random}\} \tag{7}$$

Then, the links from random strategy are voted as:

$$l^t_{random\,j} = \frac{1}{|U(t)|} \begin{cases} 1, & \text{if } s_j \in U(t) \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Note that $\sum_v P(v|s_i) = 1$ for each strategy, thus the vote sharing scheme is always normalized.

The value of these links represent only what happens at the current time. The participant's actions, however, can be tracked historically by defining an observation window of the process τ, which modulates the weight of the votes over time. Therefore, at time t, the amount of cumulative votes L between strategies i and j is defined by:

$$L_{ij}(t) = \frac{\sum_{n=0}^{t} l^q_{ij}\tau(n)}{\sum_{n=0}^{t}\tau(n)} = P(s_i \to s_j) \tag{9}$$

Here we define τ as a Gaussian function with a standard deviation of σ steps, normalized to a maximum of 1 at the current time t:

$$\tau(n) = e^{\frac{-(n-t)^2}{2\sigma^2}} \tag{10}$$

The result is a set of directed interactions (i.e., transition probabilities) among different strategies, storing historical information of the participants' behavior.

2.4.2.3. Weights optimization

Weights represent the probability distribution across strategy models at each iteration. Once emission and transition probabilities are known, weights are updated by comparing the cost, in terms of probabilities, to start in some strategy and end in the best transition link that explains an observation type. This allows us to compare the transition that best represents the current state of the behavioral model (to where the system is moving) for a specific observation, and how much it costs for each model to reach that point.

The cost, in terms of probability of moving from the state s_k to s_j (which could be the same), given an observation $v \in V$, is represented by $P(s_k \to s_j)P(v|s_j)$. Note that there may be more than one best transition for the system. Accordingly, we define set of best transition links as:

$$\{(s_k, s_j) | \arg\max_{s_k, s_j \in M} P(s_k \to s_j)P(v|s_j)\} \tag{11}$$

where M represents the set of all strategy models. Equation (11) implies that the behavioral model identifies the transition from s_k to s_j as one of the most representative in case of observing v at time t. In case more than one best transition exists, the most convenient path is optimized. Then, for an initial state s_i, the cost of reaching the best transition link is defined as the path that maximizes the following probability:

$$L_{ii}(t)P(v|s_i)P(s_a \to s_b)P(v|s_b) \cdots P(s_k \to s_j)P(v|s_j) \tag{12}$$

We use L_{ii} as the prior for the initial probability $w_i(t)$, so that models with $w_i(t-1) = 0$ can be incorporated in the optimization at time t. Note that Equation (12) is equivalent to the Viterbi path constrained to a fixed observation v.

Then, the weight of strategy s_i at time $t+1$ is represented by the total cost for the set of observations V:

$$w_i(t+1) = \sum_{v \in V} w_i(t)P(v|s_i) \cdots P(s_k \to s_j)P(v|s_j)\tau_v(t) \tag{13}$$

where $\tau_v(t)$ yields the relevance of observation v in the total cost function, given an observation window τ (such as defined in Equation 10):

$$\tau_v(t) = \sum_{n=0}^{t} \tau(n) \begin{cases} 1, & \text{if } v(t=n) = v \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

The most likely observation at time step $v(t)$ is defined by consensus regarding the observation with the best average emission probabilities:

$$v(t) = \arg\max_{v \in V} \sum_{s_i \in M} P(v|s_i) \tag{15}$$

Finally, in order to obtain a probability distribution over the search strategies, each weight is normalized by the coefficient W defined as:

$$W = \sum_{i \in M} w_i(t+1) \tag{16}$$

Note that if two or more models have the same domain of action (e.g., both concept-based strategies have the same domain of action for target path Q^*), we consider only the one with greatest knowledge. Otherwise the normalization is unfair to the other models.

2.4.2.4. Learning curve

Each strategy's individual knowledge of the BDT, α_i, can be combined with its respective weight w_i to produce a mixture of basal strategies α^*. This is accomplished by a weighted sum:

$$\alpha^*(t) = \sum_{i \in M} \alpha_i(t)w_i(t) \tag{17}$$

Recall that α_i is the complement of the specific learning curve of each strategy. Therefore, the approximate learning curve of a given participant can be obtained as: $1 - \alpha^*$.

2.4.2.5. Predicting participants' choices

As the models' weights are known previous to icon selection, we can build a Mixture of Experts (Jacobs et al., 1991) where search strategy models become the experts that must answer the question "Which icon is the participant most likely to click in the next step?". Because each model can produce any of two possible actions, i.e., clicking on the left or the right icon, the most probable next choice at step t will be the one that has the largest support as expressed by the weighted sum rule of the ensemble:

$$\underset{a \in \{\text{left,right}\}}{\arg\max} \ \mu_a(t) = \sum_{i \in M} w_i(t) \begin{cases} 1, & \text{if } a \in h_i(t) \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where $h_i(t)$ represents the expert-prediction of the strategy i. Prediction can be based on dichotomous decisions or random selection. Dichotomous decisions occur when there is only one possible action in the strategy's domain of action so that it will be selected with probability equal to 1. Alternatively, when both possible actions belongs to the strategy's domain (i.e., have equal probability of being selected) or when neither of the actions belongs to the domain (i.e., the strategy is in the inactive set), a coin is tossed to choose at random (50/50 guess).

It is worth noting that the prediction capability of each strategy depends on the size of its domain of action. Strategies with wider exploratory behaviors often have larger domains, and consequently, less predictive power due to the number of paths that can be selected. This aspect is captured by the number of 50/50 guesses, because, as mentioned above, whenever the strategy has more than one equally likely possibility of action, it must choose at random. This does not mean that the strategy itself is failing to capture the participants' choice of action, but that the conditions are ambiguous enough to keep looking for the correct path.

Finally, to better visualize the search process of participants (and groups of participants), we introduce three scores. As any sequence of choices can be the consequence of different degrees of expertise (from fully random behavior to goal-directed exploitation), we define a scale where we assign points depending on the most likely observation $v(t)$ that yields the behavioral modeling (Equation 15) at each step of a sequence of choices Q.

To measure the degree of expertise for path Q, realized between time steps t_a and t_b, we define the expertise score as:

$$\text{expertise} = \frac{1}{|Q|} \sum_{t_a <= t <= t_b} \begin{cases} 1, & \text{if } v(t) = \text{hit} \\ 0.5 & \text{if } v(t) = \text{explore} \\ 0, & \text{if } v(t) = \text{mistake} \end{cases} \tag{19}$$

where expertise equal to 1 means exploitation behavior and expertise equal to 0 means completely random behavior. Values in-between represent various degrees of exploration.

If only quantifying exploration and exploitation rates, we assign points equal to 1 only if $v(t)$ match the explore/hit observation respectively:

$$\text{exploration} = \frac{1}{|Q|} \sum_{t_a <= t <= t_b} \begin{cases} 1, & \text{if } v(t) = \text{explore} \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

Likewise, exploitation score is defined as:

$$\text{exploitation} = \frac{1}{|Q|} \sum_{t_a <= t <= t_b} \begin{cases} 1, & \text{if } v(t) = \text{hit} \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

3. RESULTS

We present the results organized around four main themes: behavioral results, model parameter dependence, individual differences in the learning process, and prediction of participants' choices.

3.1. Behavioral Results

Reaction times (RT) averaged over all participants for individual repetitions of each task instance and for the overall experiment are presented in **Figure 2**. As different participants have different number of trials per task instance, we calculated a representative average of each instance through the following interpolation scheme: we estimated a bin size for each task by calculating the average number of trials per task over the group of participants. Each individual's task-related vector was then linearly interpolated in that common space. This preserves the relative time that it takes—on average—for the participants to complete each task instance, while it allows to visualize the average of the reaction time and learning curves (**Figure 4**) more naturally.

A consistent two-fold exponential structure is visible revealing the development of expertise. Each individual task instance (same instructions) takes progressively less time to solve as participants repeat it. Likewise, successive instances of the task (different instructions) take progressively less time to solve as the session unfolds. Participants take an average of 29 ± 9 min (Mean ± *SD*) to complete the entire experimental session, with a minimum of 13 min and a maximum of 40 min, which was the maximum time allowed to solve the task.

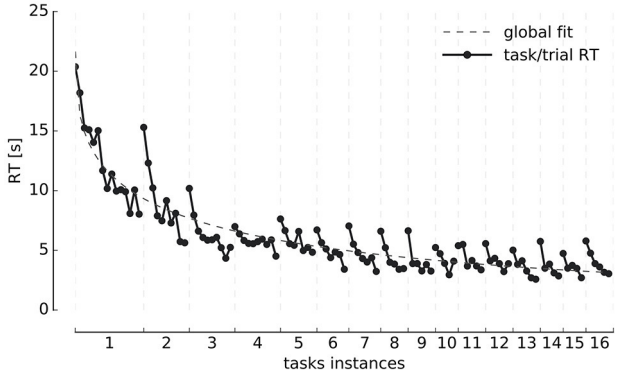

FIGURE 2 | Reaction times. Evolution of trial reaction times (y-axis) grouped by tasks instances (x-axis) and averaged over all participants. The global RT curve fit, corresponds to an exponential decay function of the type: $\lambda_1 \exp(-\lambda_2 x^{\lambda_3})$, where $\lambda_1 = 21.6$ is the starting average RT, $\lambda_2 = 0.28$, and $\lambda_3 = 0.4$. The coefficient λ_3 is necessary since the drop in time is not as steep as when $\lambda_3 = 1$ (the usual exponential decay constant).

3.2. Model Parameter Dependence

To illustrate how the modeling results are influenced by the observation window of the process τ (Equation 10), we analyze the dynamics of τ while tracking the weight of a participant's discriminative strategy (**Figure 3**).

The observation window τ affects how observations and the voting scheme impact the point of view of the observer when determining the use of a given strategy by the user. This parameter directly affects the transition probabilities of the behavioral modeling, and the weights through the temporal relevance of the observations. To test its sensitivity, we built Gaussian kernels of σ equal to 1, 12, and 20 steps, which represent 1, 3, and 5 trials/sequences of choices (see inset in **Figure 3** for a graphical representation of the kernels). These kernels cover cases between the maximum temporal weight in the current choice ($\sigma = 1$) to 50% of the temporal weight at approximately 5 trials of distance ($\sigma = 20$). The highest σ value was chosen such that it includes the temporary extension of the average number of trials to find positive feedback (8 paths), when the domain is the entire BDT.

As seen in **Figure 3**, these cases capture the general dynamics of the learning process. For example, task instances 1 and 2 are statements which belong to the same branch of the BDT. The participant uses the information learned in instance 1 to answer instance 2, what is reflected positively in the weight of concept-based strategies such as the discriminative strategy. Conversely, task instance 3 does not belong to the same branch of instances 1 and 2. This time the participant is faced with a more explorative situation, which is reflected in a decrease in the weight of the discriminative strategy. Finally, around task instances 4 and 5 the weight of the strategy is consolidated around the maximum weight, which is expected in a fully discriminative behavior, where 4 paths of positive feedback can describe the full knowledge of the BDT.

Regarding the specific sensitivity of τ across kernels, small kernels tend to react faster to changes in the style of the participant, punishing or rewarding the weight of a strategy very quickly. For example, a $\sigma = 1$ kernel produces "spikes" in **Figure 3** in tasks instance 3 when the participant goes from mistake to exploration (rewarding case). Likewise, in tasks instance 16, when participant performs incorrect actions in a context of full knowledge, the weight of the strategy is punished abruptly to match other strategies that are compatible with such behavior. On the contrary, large kernels tend to produce smoother transitions, as a more extensive history is taken into account when calculating the relevance of the current observation. For example, task instance 4 in **Figure 3** represents a shift in the strategies' weights toward discriminative behavior. As expected, the $\sigma = 1$ kernel has a steeper curve than the $\sigma = 20$ kernel around shift time. Finally, when a strategy is irrelevant in the current weights distribution, as seen from task instance 6 to 15 of **Figure 3**, distinct kernels make no difference in the strategy's final weight curve.

Although, τ can be set for different usage scenarios and types of tasks, we use $\tau(\sigma) = 20$ for modeling our data hereafter. This choice produces smoother changes in the α curves, clearly showing the process of expertise acquisition.

3.3. Individual Differences

It was possible to distinguish two groups of participants according to whether they managed to solve the task (learners, $N = 14$) or not (non-learners, $N = 8$) in the allowed time window (40 min). We consider learners (**Figure 4A**), all participants who managed to complete the 16 tasks instances, reaching full knowledge of the interface as evidenced by consistent positive feedback. Conversely, the non-learners group (**Figure 4B**), is composed by participants who did not complete the task within the time limit, or failed to reach full knowledge of the interface.

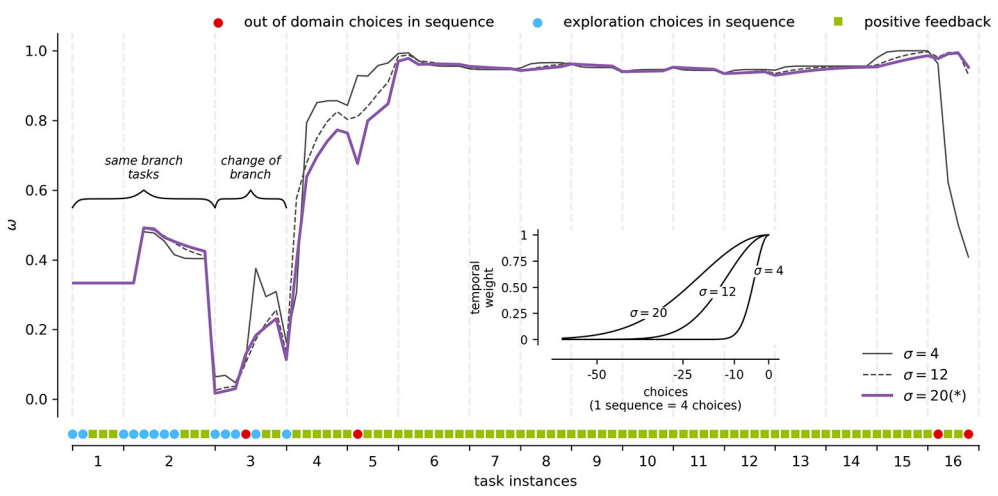

FIGURE 3 | Framework parameters modulation. An example for a participant's discriminative strategy weight (y-axis) for each task instance (x-axis) at different τ values (see inset). The inset x-axis presents the distance in choices from the current time and the y-axis the respective kernel weight. The curve marked by (*) represents the σ used for modeling the rest of results presented in the section.

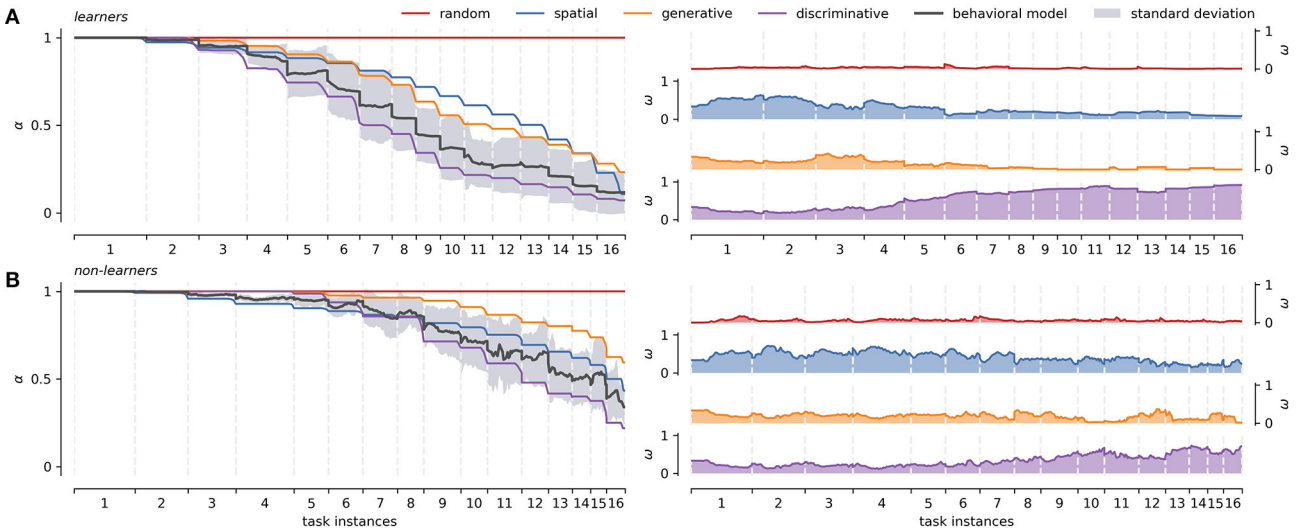

FIGURE 4 | Learners and non-learners average strategies across tasks. **(A)** Right panel: characterization of the strategy models averaged over all participants (66.7%) that managed to solve the sixteen different instances of the task (x-axis). Left panel: average weights of each strategy models over the same group of participants. **(B)** Right panel: characterization of the strategy models averaged over all participants (33.3%) that didn't fully learn the task. Left panel: average weights of each strategy models over the same group of participants. The behavioral modeling represents the averaged individual approximation (basal strategies weighting) and is shown with its standard error across tasks. Dispersion of α curves for basal strategies is not presented to facilitate visualization. The latter can be seen in greater detail in **Figure 5** below. The same interpolation scheme as in **Figure 2** is used.

For the learners group, the left panel in **Figure 4A** shows the distinctive shape of each strategy as they reach full knowledge. The right panel in **Figure 4A** shows the average weight of each strategy model across the 16 different task instances. Recall that the absence of learning under the random strategy is represented as a straight, constant line at $\alpha = 1$. In the spatial model, as learning progresses, the number of attempts necessary to encounter positive feedback decreases exponentially. Finally, generative and discriminative models are represented by increasingly steep sigmoid functions, as complete knowledge of the BDT can be obtained with less positive feedback exposure. It is worth noting that the definition of learning for concept-based strategies (no selection mistakes between two consecutive paths with the same target concept), causes that not all learning curves reach zero for all participants. This is mainly because not all concepts can be checked for the learning condition in one run of the task. The average group-level behavioral modeling remains between the space delimited by the generative and discriminative strategies, following more closely the discriminative strategy. The right panel in **Figure 4A** shows that the weight of the discriminative strategy starts outperforming the rest of the strategies around task instance 4. Before that, the spatial strategy is used to locate positive feedback. Generative strategy has a short transition period between spatial and discriminative strategies around task 3, loosing prominence quickly after that.

In contrast, non-learners tend to go through successive task instances without obtaining positive feedback (i.e., they are prompted to continue after persistent mistakes, see Methods Section). Accordingly, in this group all strategies suffer substantially regarding knowledge completion, never reaching full knowledge (**Figure 4B** left panel). In general, non-learners do not generate as much explicit knowledge as learners do, focusing on spatial exploration over conceptual mapping in order to find positive feedback. This is reflected in the fact that the spatial α curve surpasses the generative α curve. Importantly, however, some implicit knowledge seems to be generated based on the tasks that they do manage to complete. This would explain why the discriminative strategy gains weight toward the end of the task (**Figure 4B** right panel task instance 9). In this group, the average behavioral modeling moves between spatial and discriminative strategies.

Figure 5 presents the deployment of strategies and their corresponding weights for representative cases of the two groups. In these cases, each strategy reaches different levels of knowledge depending on the participants' context. A highly proficient learner, **Figure 5A**, usually goes from random to discriminative strategies progressively, as seen in the weights panel. Few mistakes are committed, and the learning curve $(1 - \alpha^*)$ approximates a sigmoidal shape. A less proficient learner such as the one shown in **Figure 5B**, initially goes through a similar process, but also through periods of difficulty in solving the task (e.g., from task instances 7 to 9 in **Figure 5B**). The corresponding learning curve presents a sigmoid shape interrupted by intervals of evidence for spatial and random strategies, which is caused by explorations and mistakes. These intervals have different durations and extensions, finally disappearing when the discriminative strategy regains dominance. A non-learner, as the case shown in **Figure 5C**, typically has a similar starting strategy distribution as learners. However, instead of converging to the discriminative strategy, the participant falls back mainly to the

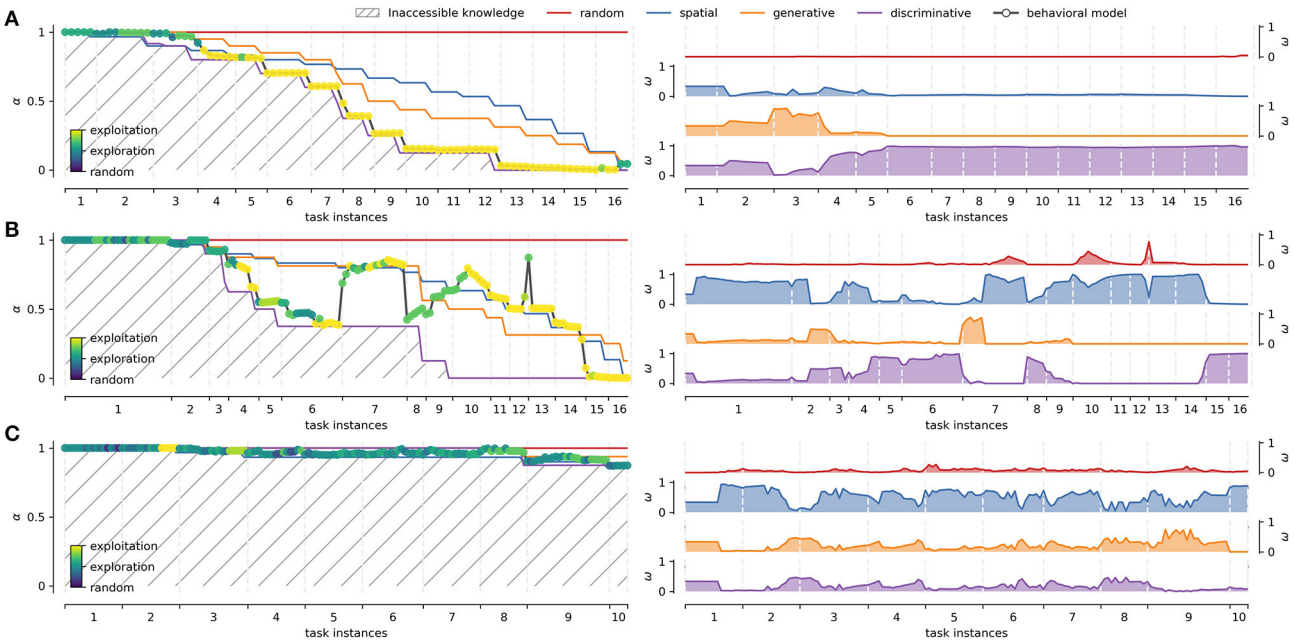

FIGURE 5 | Study cases. Three representative individual cases according to task performance. **(A)** Highly proficient learner. **(B)** Less proficient learner. **(C)** Non-learner. The left panel of each case shows the deployment of basal strategies in terms of α during the overall duration of the experiment (x-axis). Colored dots represent different degrees of expertise used to explore/exploit the interface knowledge (Equation 19). The right panels of each case shows the evolution of the corresponding weights for each strategy across task instances. The average of the basal strategy weights yield α^* for the behavioral modeling.

spatial strategy. As they do not appear to learn most of the concepts, they must search for positive feedback in less efficient ways during the whole task.

3.4. Predicting Participants' Choices

In order to better understand the actions of the participants, we tracked the emission probabilities of exploration (Equation 20) and exploitation (Equation 21), comparing them with the actual performance of the participant in the task. **Figure 6** leftmost panel shows these comparisons for learners and non-learners. In the case of learners, the model identifies complementary curves for exploration and exploitation (exploration + exploitation $\approx$ 1). Learners tend to be more explorative at first, but their performance falls between exploration and exploitation. From task instance 4 onwards, the performance of learners is similar to what the model predicts as exploitation. After that, exploration behavior tends to disappear and exploitation increases. This is congruent with **Figure 4A** left panel, where learners begin to use the discriminative strategy more consistently from task instance 4. In contrast, non-leaners do not have complementary exploration and exploitation curves. This is because of the random behavior during the search of positive feedback. Although the actual behavior of non-learners is similar to the exploitation curve, the performance level remains under 50%. This is probably because they only map the easiest levels of the BDT (level 1 and 4), while exploring the rest of the tree in each task instance.

To compare the performance of our behavioral model, we use a Mixture of Experts (Equation 18) to predict the participant's most likely next choice. Prediction outputs two values: a next possible action and the number of 50/50 guesses (see Section Methods), normalized by the experts' weights. The same concepts apply to individual strategies, so that the prediction capabilities can be compared by the prediction output of each strategy. **Figure 6** center and rightmost panels compare choice prediction capabilities and the number of 50/50 guesses for both learners and non-leaners. At the beginning, when the knowledge of the BDT is still poor, we expect a high 50/50 guess rate with a consequently low prediction accuracy. In both groups, prediction accuracy starts at around 25% (two choices). This is because, on average, it is not difficult to guess the first and four level icons at random, regardless of it being a participant or an observer. From then on, learners progressively improve their performance, reaching values close to 90% halfway through the experiment, and near 100% toward the end of the task. The number of 50/50 guesses follows the previous trend, stabilizing near zero around task instance 10. In the case of non learners, the number of 50/50 guesses is quite high, only decreasing around task instance 9 and converging to two 50/50 guesses choices. The initial prediction capabilities of two choices is maintained throughout the course of the task, and only rises to about three choices toward the end of the task.

In both groups the leading single strategy is the discriminative, which is followed closely by the behavioral model. This is consistent with the results presented in **Figure 4**, where the weight of the discriminative strategy follows the critical moments where it differs from the rest of the strategies (task instance 4 for learners and 9 for non-learners). This suggests that the core

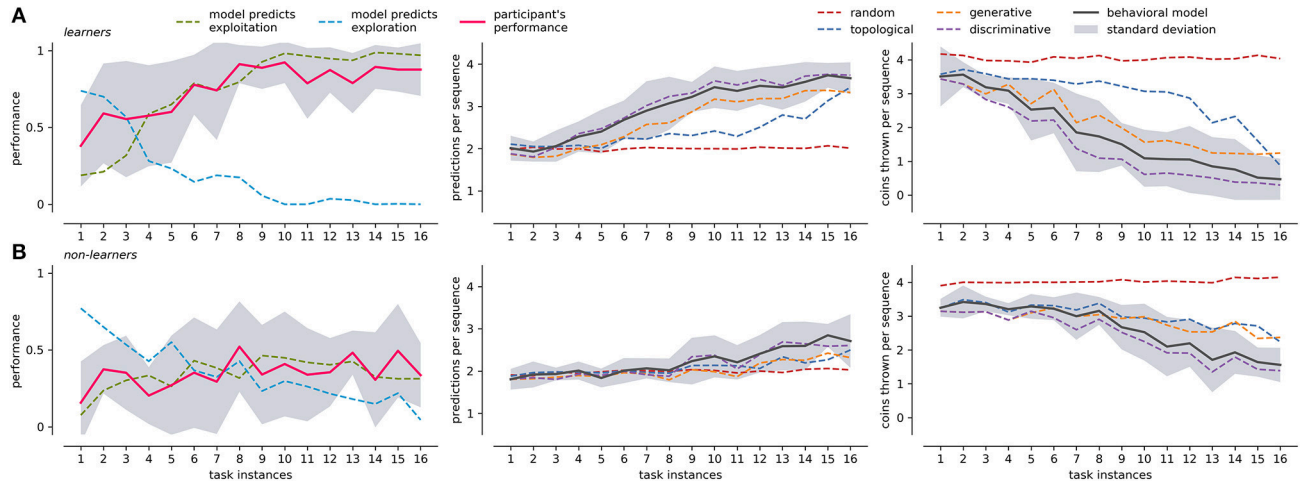

FIGURE 6 | Participants performance, framework prediction, and 50/50 guesses averaged across tasks. **(A)** Presents results for the learners group: the leftmost panel shows the participant's averaged performance (y-axis), compared with exploration/exploitation performance as predicted by the behavioral modeling; the center panel present the average number of predicted choices (y-axis) in a sequence for behavioral model and individual strategies; the rightmost panel present the number of 50/50 guesses (y-axis left) for center panel. The x-axis for all panels is the average across the 16 different tasks. **(B)** Presents the equivalent measures of **(A)** for non-learners. The corresponding standard deviation of participants' performance and behavioral modeling is shown in gray. Standard deviation for the rest of the curves is not presented to facilitate visualization.

rules of the BDT mapping are learned through this mechanism. However, the discriminative strategy is not able to fully explain the behavior of the participants in all the stages of expertise acquisition (and the other strategies do not explain it either). This may be due to the fact that before the critical point of the discriminative strategy, participants are not as systematic as the strategies propose (or it is necessary to define different search strategies to explain such behavior). In the case of learners, the behavioral model predicts more exploratory behavior (**Figure 6** leftmost panel) before that point. In other words, the modeling framework predicts that there will be a greater amount of exploration, independent of the exact way in which it would take place.

4. DISCUSSION

We have presented here a modeling framework which, on the basis of stereotyped, well-defined search strategies, is able to track the development of expertise in a sequential decision-making task. We have tested the framework using a novel BDT task that has the potential to provide a well-defined computer interface scenario for the study of expertise acquisition in real world situations. By following individual learning processes, when confronted with users that are able to learn the task, our model quickly reaches a point where it can predict the users' most likely next choice. This suggest that the framework presented here could play a role in the development of adaptive interfaces where sequential choices are required to reach a desired outcome.

We organize the following discussion around four main issues: the modeling framework choice and its consequences for both the tracking of the learning process and for adaptive interface design; the structure of the sequential decision making task; the criteria for temporal and spatial information encoding; and the issue of individual differences among users.

4.1. Modeling Framework

As outlined in the introduction, a host of approaches exist to deal with the problem of modeling decision making behavior in general (Sukthankar et al., 2014), and sequential decision making in particular (Walsh and Anderson, 2014). Here we have chosen to follow an HMM-based approach that, in line with type-based methods, relies on a set of pre-defined strategies to model a potentially infinite set of behaviors (Albrecht et al., 2016). This approach reduces the dimensionality of the problem while at the same time aims to provide some insight into the learning and expertise acquisition process.

While the approach presented here is indeed inspired by the HMM structure, we do not use traditional algorithms to determine the parameters of the HMM (e.g., BaumWelch, see Rabiner, 1989 for more details). This is because, in our task, emission and transition probabilities are dynamic consequences of the participant's actions. Therefore, we can ask the strategies for the emission probabilities, because they directly operationalize possible outcomes (as a consequence of their formal definition). Transition probabilities are, in contrast, a more open problem because they define how the HMM is connected (i.e., its structure), which can itself change as learning happens. To deal with this issue, we define transition probabilities in terms of the comparative changes of emission probabilities between strategies. This is because, as discussed above, strategies formalize actual domains of action so that their comparison allows us to determine toward which strategy it is better to transit in order to best explain the current choice made by the participant. Finally, in the case of weight optimization, we use a

formulation which is equivalent to finding the Viterbi path in a HMM (also described in Rabiner, 1989).

As our results show, the modeling approach presented here was able to consistently follow the performance of individual users as they became familiarized with the task. In those that were able to discover the underlying icon-concept mapping and fully solve the task (learners), the model was able to match their performance as early as the fourth instance of the task. More importantly for the question of how learning proceeds in a sequential decision making situation, the use of pre-defined strategies suggests that learners and non-learners are likely to use different approaches to the problem: while learners dwell on spatial, brute-search behavior only during a short while and then rapidly start using concept-mapping knowledge (mostly in discriminative terms), non-learners persist in spatial searches and only well into the task do they start showing behavior of consistent icon-concept mappings.

However, to the extent that the pre-defined strategies represent one possible set of high-level behaviors among many, our approach cannot provide definite evidence that users actually use such strategies. In this sense, it may well be the case that a different modeling approach (or a different subset of strategies) might capture equally well the user's behavior. This is, of course, a pervasive problem in behavioral modeling when users are free to tackle the problem at hand in whatever way they choose. Approaches that combine modeling with subjective reports such as Mariano et al. (2015) are therefore an interesting extension and worth exploring further. Indeed, Beta Process Hidden Markov Models (BP-HMM) (Fox et al., 2009) such as the one used by Mariano et al. (2015), consider libraries of states. Here, one pattern of states, represented by an HMM, is active at current time. This approach captures the idea of multiple patters of HMM structures, but the interpretation of such patterns in terms of the user's behavior is necessarily an exercise that has to be done a posteriori. Nevertheless, the pre-defined strategies that we chose here are informed by the structure of the problem and represent increasingly efficient ways to solve it, thus representing a valid alternative in the context of type-based approaches while being more sensitive to potential differences in the way users approach the task. Other modeling techniques, such as Interactive Partially Observable Markov Decision Processes (I-POMDP) (Gmytrasiewicz and Doshi, 2005), also rely on a similar approach by using predefined intuitions to solve the task (i.e., types of behaviors/strategies), thus allowing to capture the learning of users more efficiently and faster.

A final point regarding the use of a subset of high-level behaviors is worth noting: for instance, as shown in **Figure 6** it could be argued that considering the discriminative strategy is enough to yield comparable results in terms of tracking the behavior of those who learn the task. This is expected given that by the fourth iteration, users should have been exposed to most of the discriminative knowledge necessary to fully map de problem. Yet, if we chose this approach, we would have no insight into potentially relevant strategies that unfold throughout the learning processes.

In addition to the previous considerations, an a posteriori validation of the approach is its capacity to predict the behavior of individual users. When it comes to studying the agent's interaction with a user, prediction inference is usually performed over the users' goals (Oh et al., 2011; Ramírez and Geffner, 2011). Such models identify the goal and the necessary output actions to fulfill that specific goal. In contrast, we specify the goal and our inference is about the way in which the user perform the actions in a search process. The strategies defined here represent policies of actions that could be optimal during different stages of learning. In this sense, prediction performance is not in and by itself, the primary measure of the model's capacity. For instance, even with poor prediction performance (e.g., **Figure 6B** center panel), the behavioral model can follow the actual performance of the participants based on the exploitation score. In addition, the number of 50/50 guesses can be considered an estimation of how good the prediction could be. As such, this approach (and extensions of it in terms, for instance, of a different set of strategies) could represent a viable approach to inform the construction of flexible interfaces that can adapt to the different moments of the learning curve of their users.

4.2. Task Structure

There are several types of sequential tasks studied in the literature (Ruh et al., 2010; Diuk et al., 2013; Friedel et al., 2014; Huys et al., 2015). In this context, decision trees present an interesting scenario, because they naturally embody the most basic structures of a sequential decision-making situation (Fu and Anderson, 2006) while allowing for a clear description of the task structure. Here we chose a BDT design to instantiates a simple case of sequential action in Human-Computer Interactions (HCI).

An important aspect of our BDT design is the feedback structure. Specifically, participants have to reach the end of each branch before obtaining information about the appropriateness of previous decisions. In this kind of limited feedback scenario, a *credit assignment problem* appears (Fu and Anderson, 2008; Walsh and Anderson, 2011, 2014). This means that participants have to learn how to use the consequences of their actions to assign value to different parts of the sequential choice. In our task, participants face two main types of credit assignment problems: on the one hand, negative feedback is not enough in and by itself to discover at which point of the BDT wrong decisions were made. On the other hand, positive feedback can only be used by the participant to generate specific icons-concepts mappings through successive exposures to different instances of the task.

One could argue that immediate feedback interactions (such as interfaces characterized by labeled icons or explicit icon-concept mappings) is the dominant mode of HCI interaction and should therefore be the target for behavioral modeling. This would have the advantage of sidestepping the credit assignment problem. However, limited feedback scenarios such as the one used here have the advantage of making the task more difficult (Walsh and Anderson, 2011), therefore revealing the learning process more clearly. This has obvious advantages in terms of making the process of expertise acquisition by different users explicit and thus available for modeling efforts. Moreover, given the nature of our behavioral model, which deals with relationships among strategies, immediate feedback scenarios

could be considered as a particular case for the framework. When interfaces are built on the basis of labeled icons, only two strategies are required: spatial and discriminative: the spatial strategy is necessary to map the physical layout and arrangement of information throughout the interface; the discriminative strategy, on the other hand, allows one to relate complementary or neighboring icons to future (alternative) task instances. The generative strategy, in contrast, is unnecessary because the primary icon-concept mapping is explicit from the start.

4.3. Temporal and Spatial Encoding Criteria

Related to the credit assignment problem, another important issue arises when modeling behavior, especially in sequential decision making situations. This issue, which has received much less attention from modeling studies, pertains the criteria that one sets in order to determine which observations count as relevant information (Behrens et al., 2007). Such criteria can be of temporal nature (for instance, how much of past experience we consider when planning future actions) or spatial (for example, how much of the BDT's structure is remembered and used for making decisions).

In our work, the function-parameter τ determines the influence of past outcomes on the behavioral modeling. Here we have chosen a continuous Gaussian kernel (Equation 10) with a peak at the current time. This choice aims to capture underlying short term memory processes whereby current items have a higher probability of influencing behavior than those encountered previously (Cowan, 2008). Whether by interference of novel task information or due to temporal decay, current information cannot not persist indefinitely and therefore it is necessary to consider the dynamics of its causal influence on ongoing decision making (Lewandowsky and Oberauer, 2009; Barrouillet et al., 2012). The choice of the Gaussian kernel could be a target for improvement, eventually considering the possibility of adapting it to individual users' profiles. Nevertheless, similar exponential metrics, including Gaussian kernels have been used to model complex memory processes and therefore represents a viable first choice (Brown et al., 2007).

In addition to the temporal problem, a spatial navigation problem arises that is related to the localization of feedback (Madl et al., 2015). To localize the correct path, as well as to propagate credit to previous choices, it is necessary for the model to encode some type of memory of the user's actions (Fu and Anderson, 2008). In our work, each strategy has two types of rules: a global memory associated to learning the overall BDT structure according to that particular strategy across task instances, and a local memory, which is associated with narrowing the domain of action leading to correct feedback in the current instance of the task. Importantly, such encoding does not forget landmarks that have been encountered by the participant and in this sense does not represent actual memory processes. However, this is in principle not necessary because we are dealing with a highly restricted situation in which a full mapping of the task can happen within one experimental session. Indeed, we use a task repetition criterion to ensure learning and make forgetting more difficult (Corazzini et al., 2008). Eventually, when dealing with more complex search scenarios, taking into account that the user might forget a previously encountered location might be necessary (Baumann et al., 2011; Liverence and Scholl, 2015). Nevertheless, we account for apparent reductions in the user's knowledge of the BDT structure using the random model to penalize actions that, according to the model's encoding of the user's previous actions, represent mistakes in landmark choices.

4.4. Individual Differences among Users

One of our main results, which is related to the structure of the task, is the notorious difference among participants in terms of performance. Beyond the evident interest this has in terms of underlying psychological and brain mechanisms, from our perspective, this represents an opportunity to contrast the performance of participants of different skill levels when dealing with search problems (Sacchi and Burigo, 2008). Our task reveals that a standard group of participant is not uniform and can be separated in learners and non-learners, as a first level of analysis. These two groups differ not only in terms of whether they manage to complete the task, but also in terms of which strategies they prefer.

Such differentiation is relevant when we want to distinguish experts from non-experts. When dealing with a problem-solving situation, experts and non-experts differ on a number of dimensions. For instance, they differ in how they search for information (Barrick and Spilker, 2003; Hershler and Hochstein, 2009), make decisions (Connors et al., 2011; Gorman et al., 2015) or pay attention (Schriver et al., 2008), among others. Taking into account these differences is a critical step toward a more natural and truly adaptive HCI, as the user's needs could be focused more accurately according to their level of expertise. For example, being able to identify the strategy that is currently being deployed could enable the interface to display contents accordingly. Likewise, it could be in principle possible to speed the learning processes with unfamiliar interfaces by showing specific types of interface-hints depending on the user's history of interactions. In the case of non-experts, one could provide contextual cues to facilitate the development or discovery of strategies that have proven successful for experts. Alternatively, because the model is capable of detecting quite robustly when a user is not learning (i.e., when the random strategy dominates), the interface could choose to display challenging options or hints in order to "wake up" the user in such cases.

It is notorious that individual differences in behavior have rarely been the target for computational modeling or, in some cases, even treated as a nuisance (Karwowski and Cuevas, 2003). This is all the more surprising given the importance that recognizing individual differences can have for such disparate but fundamental domains as educational interventions (Detterman and Thompson, 1997; Phillips and Lowenstein, 2011; Melby-Lervag et al., 2012) or human performance studies (Van Dongen, 2006; Parasuraman and Jiang, 2012; Goel et al., 2013). We surmise that the framework presented here could be used to tackle this issue because it is built upon modular strategies whose combination can capture a diversity of behaviors, even among learners (see for instance **Figure 5A** vs. **Figure 5B** in which two different types of learners can be clearly distinguished). We believe that much more research is needed on the issue of

modeling behavior in a way that takes into account the specifics of individuals, in addition to average cases or proof of concept approaches (Smith et al., 2014).

AUTHOR CONTRIBUTIONS

CM participated in the experimental design, performed the experiment and acquired, analyzed the data and contributed with the interpretation of the results and the writing of the manuscript. DC designed the experiment, contributed with experimental and data recording equipment, contributed to the interpretation of the results and wrote and edited the manuscript. RV contributed with the analysis of the data and with the interpretation of the results as well as revising critically the manuscript. VL contributed to the experimental design and interpretation of the results, participating in the writing and revision of the manuscript. DM contributed with the interpretation of the data, the formalization of the model, and the critical revision of the manuscript.

FUNDING

This work was supported by CONICYT National Ph.D. grant number 21110823 to CM, FONDECYT National postdoctoral grant number 3160403 to RV, FONDECYT grant 1130758 to DC, and FONDECYT grant 1150241 to VL.

REFERENCES

Abbeel, P., and Ng, A. Y. (2004). "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the Twenty-First International Conference on Machine Learning* (Banff, AB).

Acuña, D. E., and Schrater, P. (2010). Structure learning in human sequential decision-making. *PLoS Comput. Biol.* 6:e1001003. doi: 10.1371/journal.pcbi.1001003

Alagoz, O., Hsu, H., Schaefer, A. J., and Roberts, M. S. (2009). Markov decision processes: a tool for sequential decision making under uncertainty. *Med. Decis. Making* 30, 474–483. doi: 10.1177/0272989X09353194

Albrecht, S. V., Crandall, J. W., and Ramamoorthy, S. (2016). Belief and truth in hypothesised behaviours. *Artif. Intell.* 235, 63–94. doi: 10.1016/j.artint.2016.02.004

Baker, C. L., Saxe, R., and Tenenbaum, J. B. (2009). Action understanding as inverse planning. *Cognition* 113, 329–349. doi: 10.1016/j.cognition.2009.07.005

Baker, C. L., Saxe, R., and Tenenbaum, J. B. (2011). "Bayesian theory of mind: modeling joint belief-desire attribution," in *Proceedings of the Cognitive Science Society* (Boston, MA), 2469–2474.

Barrick, J. A., and Spilker, B. C. (2003). The relations between knowledge, search strategy, and performance in unaided and aided information search. *Organ. Behav. Hum. Decis. Process.* 90, 1–18. doi: 10.1016/S0749-5978(03)00002-5

Barrouillet, P., De Paepe, A., and Langerock, N. (2012). Time causes forgetting from working memory. *Psychon. Bull. Rev.* 19, 87–92. doi: 10.3758/s13423-011-0192-8

Baumann, O., Skilleter, A. J., and Mattingley, J. B. (2011). Short-term memory maintenance of object locations during active navigation: which working memory subsystem is essential? *PLoS ONE* 6:e19707. doi: 10.1371/journal.pone.0019707

Behrens, T. E. J., Woolrich, M. W., Walton, M. E., and Rushworth, M. F. S. (2007). Learning the value of information in an uncertain world. *Nat. Neurosci.* 10, 1214–1221. doi: 10.1038/nn1954

Botvinick, M., and Weinstein, A. (2014). Model-based hierarchical reinforcement learning and human action control. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 369:20130480. doi: 10.1098/rstb.2013.0480

Botvinick, M. M. (2012). Hierarchical reinforcement learning and decision making. *Curr. Opin. Neurobiol.* 22, 956 – 962. doi: 10.1016/j.conb.2012.05.008

Brown, G. D. A., Neath, I., and Chater, N. (2007). A temporal ratio model of memory. *Psychol. Rev.* 114, 539–576. doi: 10.1037/0033-295X.114.3.539

Bruner, J. S. (1973). Organization of early skilled action. *Child Dev.* 44, 1–11.

Connors, M. H., Burns, B. D., and Campitelli, G. (2011). Expertise in complex decision making: the role of search in chess 70 years after de groot. *Cogn. Sci.* 35, 1567–1579. doi: 10.1111/j.1551-6709.2011.01196.x

Corazzini, L. L., Thinus-Blanc, C., Nesa, M.-P., Geminiani, G. C., and Peruch, P. (2008). Differentiated forgetting rates of spatial knowledge in humans in the absence of repeated testing. *Memory* 16, 678–688. doi: 10.1080/09658210802286931

Cowan, N. (2008). What are the differences between long-term, short-term, and working memory? *Prog. Brain Res.* 169, 323–338. doi: 10.1016/S0079-6123(07)00020-9

Cushman, F., and Morris, A. (2015). Habitual control of goal selection in humans. *Proc. Natl. Acad. Sci. U.S.A.* 112, 13817–13822. doi: 10.1073/pnas.1506367112

Daw, N. D. (2014). "Advanced reinforcement learning," in *Neuroeconomics, 2nd Edn.*, eds P. W. Glimcher and E. Fehr (San Diego, CA: Academic Press), 299–320. doi: 10.1016/B978-0-12-416008-8.00016-4

Dayan, P., and Niv, Y. (2008). Reinforcement learning: the good, the bad and the ugly. *Curr. Opin. Neurobiol.* 18, 185–196. doi: 10.1016/j.conb.2008.08.003

Detterman, D. K., and Thompson, L. A. (1997). What is so special about special education? *Am. Psychol.* 52, 1082–1090.

Dezfouli, A., and Balleine, B. W. (2013). Actions, action sequences and habits: evidence that goal-directed and habitual action control are hierarchically organized. *PLoS Comput. Biol.* 9:e1003364. doi: 10.1371/journal.pcbi.1003364

Diuk, C., Schapiro, A., Crdova, N., Ribas-Fernandes, J., Niv, Y., and Botvinick, M. (2013). "Divide and conquer: hierarchical reinforcement learning and task decomposition in humans," in *Computational and Robotic Models of the Hierarchical Organization of Behavior*, eds G. Baldassarre and M. Mirolli (Berlin; Heidelberg: Springer), 271–291.

Doya, K., and Samejima, K. (2002). Multiple model-based reinforcement learning. *Neural Comput.* 14, 1347–1369. doi: 10.1162/089976602753712972

Duffin, E., Bland, A. R., Schaefer, A., and De Kamps, M. (2014). Differential effects of reward and punishment in decision making under uncertainty: a computational study. *Front. Neurosci.* 8:30. doi: 10.3389/fnins.2014.00030

Fischer, K. W. (1980). A theory of cognitive development: the control and construction of hierarchies of skills. *Psychol. Rev.* 87:477.

Fox, E., Jordan, M. I., Sudderth, E. B., and Willsky, A. S. (2009). "Sharing features among dynamical systems with beta processes," in *Advances in Neural Information Processing Systems* (Vancouver, BC), 549–557.

Friedel, E., Koch, S. P., Wendt, J., Heinz, A., Deserno, L., and Schlagenhauf, F. (2014). Devaluation and sequential decisions: linking goal-directed and model-based behaviour. *Front. Hum. Neurosci.* 8:587. doi: 10.3389/fnhum.2014.00587

Fu, W.-T., and Anderson, J. (2008). Solving the credit assignment problem: explicit and implicit learning of action sequences with probabilistic outcomes. *Psychol. Res.* 72, 321–330. doi: 10.1007/s00426-007-0113-7

Fu, W.-T. T., and Anderson, J. R. (2006). From recurrent choice to skill learning: a reinforcement-learning model. *J. Exp. Psychol. Gen.* 135, 184–206. doi: 10.1037/0096-3445.135.2.184

Gershman, S. J., Markman, A. B., and Otto, A. R. (2014). Retrospective revaluation in sequential decision making: a tale of two systems. *J. Exp. Psychol. Gen.* 143, 182–194. doi: 10.1037/a0030844

Ghezzi, C., Pezzè, M., Sama, M., and Tamburrelli, G. (2014). "Mining behavior models from user-intensive web applications," in *Proceedings of the 36th International Conference on Software Engineering* (Hyderabad), 277–287.

Gmytrasiewicz, P. J., and Doshi, P. (2005). A framework for sequential planning in multi-agent settings. *J. Artif. Intell. Res.* 24, 49–79. doi: 10.1613/jair.1579

Goel, N., Basner, M., Rao, H., and Dinges, D. F. (2013). Circadian rhythms, sleep deprivation, and human performance. *Prog. Mol. Biol. Transl. Sci.* 119, 155–190. doi: 10.1016/B978-0-12-396971-2.00007-5

Gorman, A. D., Abernethy, B., and Farrow, D. (2015). Evidence of different underlying processes in pattern recall and decision-making. *Q. J. Exp. Psychol.* 68, 1813–1831. doi: 10.1080/17470218.2014.992797

Hershler, O., and Hochstein, S. (2009). The importance of being expert: top-down attentional control in visual search with photographs. *Atten. Percept. Psychophys.* 71, 1478–1486. doi: 10.3758/APP.71.7.1478

Huys, Q. J. M., Lally, N., Faulkner, P., Eshel, N., Seifritz, E., Gershman, S. J., et al. (2015). Interplay of approximate planning strategies. *Proc. Natl. Acad. Sci. U.S.A.* 112, 3098–3103. doi: 10.1073/pnas.1414219112

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Comput.* 3, 79–87.

Karwowski, W., and Cuevas, H. M. (2003). Considering the importance of individual differences in human factors research: no longer simply confounding noise. *Proc. Hum. Factors Ergon. Soc. Annu. Meet.* 47, 1082–1086. doi: 10.1177/154193120304700908

Laud, A. D. (2004). *Theory and Application of Reward Shaping in Reinforcement Learning*. Ph.D. thesis, University of Illinois at Urbana-Champaign.

Lewandowsky, S., and Oberauer, K. (2009). No evidence for temporal decay in working memory. *J. Exp. Psychol. Learn. Mem. Cogn.* 35, 1545–1551. doi: 10.1037/a0017010

Liverence, B. M., and Scholl, B. J. (2015). Object persistence enhances spatial navigation: a case study in smartphone vision science. *Psychol. Sci.* 26, 955–963. doi: 10.1177/0956797614547705

Madl, T., Chen, K., Montaldi, D., and Trappl, R. (2015). Computational cognitive models of spatial memory in navigation space: a review. *Neural Netw.* 65, 18–43. doi: 10.1016/j.neunet.2015.01.002

Manoel, E. D. J., Basso, L., Correa, U. C., and Tani, G. (2002). Modularity and hierarchical organization of action programs in human acquisition of graphic skills. *Neurosci. Lett.* 335, 83–86. doi: 10.1016/S0304-3940(02)01102-3

Mariano, L., Poore, J., Krum, D., Schwartz, J., Coskren, W., and Jones, E. (2015). Modeling strategic use of human computer interfaces with novel hidden markov models. *Front. Psychol.* 6:919. doi: 10.3389/fpsyg.2015.00919

Marken, R. S. (1986). Perceptual organization of behavior: a hierarchical control model of coordinated action. *J. Exp. Psychol. Hum. Percept. Perform.* 12:267.

Matsuzaka, Y., Picard, N., and Strick, P. L. (2007). Skill representation in the primary motor cortex after long-term practice. *J. neurophysiol.* 97, 1819–1832. doi: 10.1152/jn.00784.2006

Melby-Lervag, M., Lyster, S.-A. H., and Hulme, C. (2012). Phonological skills and their role in learning to read: a meta-analytic review. *Psychol. Bull.* 138, 322–352. doi: 10.1037/a0026744

Oh, J., Meneguzzi, F., and Sycara, K. (2011). "Probabilistic plan recognition for intelligent information agents," in *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence* (Rome), 281–287.

Otto, A. R., Skatova, A., Madlon-Kay, S., and Daw, N. D. (2014). Cognitive control predicts use of model-based reinforcement learning. *J. Cogn. Neurosci.* 27, 319–333. doi: 10.1162/jocn_a_00709

Parasuraman, R., and Jiang, Y. (2012). Individual differences in cognition, affect, and performance: behavioral, neuroimaging, and molecular genetic approaches. *Neuroimage* 59, 70–82. doi: 10.1016/j.neuroimage.2011.04.040

Phillips, D. A., and Lowenstein, A. E. (2011). Early care, education, and child development. *Annu. Rev. Psychol.* 62, 483–500. doi: 10.1146/annurev.psych.031809.130707

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 257–286.

Rabinovich, M. I., Huerta, R., and Afraimovich, V. (2006). Dynamics of sequential decision making. *Phys. Rev. Lett.* 97:188103. doi: 10.1103/PhysRevLett.97.188103

Ramírez, M., and Geffner, H. (2011). "Goal recognition over pomdps: inferring the intention of a pomdp agent," in *Twenty-Second International Joint Conference on Artificial Intelligence* (Barcelona).

Rosenbaum, D. A., Cohen, R. G., Jax, S. A., Weiss, D. J., and Van Der Wel, R. (2007). The problem of serial order in behavior: Lashley's legacy. *Hum. Mov. Sci.* 26, 525–554. doi: 10.1016/j.humov.2007.04.001

Ruh N., Cooper, R. P., and Mareschal, D. (2010). Action selection in complex routinized sequential behaviors. *J. Exp. Psychol. Hum. Percept. Perform.* 106, 99–114. doi: 10.1037/a0017608

Sacchi, S., and Burigo, M. (2008). Strategies in the information search process: interaction among task structure, knowledge, and source. *J. Gen. Psychol.* 135, 252–270. doi: 10.3200/GENP.135.3.252-270

Schriver, A. T., Morrow, D. G., Wickens, C. D., and Talleur, D. A. (2008). Expertise differences in attentional strategies related to pilot decision making. *Hum. Factors* 50, 864–878. doi: 10.1518/001872008X374974

Schulte, P. J., Tsiatis, A. A., Laber, E. B., and Davidian, M. (2014). Q-and a-learning methods for estimating optimal dynamic treatment regimes. *Stat. Sci.* 29, 640–661. doi: 10.1214/13-STS450

Sepahvand, N. M., Stöttinger, E., Danckert, J., and Anderson, B. (2014). Sequential decisions: a computational comparison of observational and reinforcement accounts. *PLoS ONE* 9:e94308. doi: 10.1371/journal.pone.0094308

Shteingart, H., and Loewenstein, Y. (2014). Reinforcement learning and human behavior. *Curr. Opin. Neurobiol.* 25, 93–98. doi: 10.1016/j.conb.2013.12.004

Sims, C. R., Neth, H., Jacobs, R. A., and Gray, W. D. (2013). Melioration as rational choice: sequential decision making in uncertain environments. *Psychol. Rev.* 120, 139–154. doi: 10.1037/a0030850

Singer, P., Helic, D., Taraghi, B., and Strohmaier, M. (2014). Detecting memory and structure in human navigation patterns using markov chain models of varying order. *PLOS ONE* 9:e102070. doi: 10.1371/journal.pone.0102070

Smith, T., Henning, R., Wade, M., and Fisher, T. (2014). *Variability in Human Performance*. Boca Raton, FL: Taylor & Francis.

Solway, A., and Botvinick, M. M. (2012). Goal-directed decision making as probabilistic inference: a computational framework and potential neural correlates. *Psychol. Rev.* 119, 120–154. doi: 10.1037/a0026435

Sukthankar, G., Geib, C., Bui, H. H., Pynadath, D., and Goldman, R. P. (2014). *Plan, Activity, and Intent Recognition: Theory and Practice*. Waltham, MA: Newnes.

Van Dongen, H. P. A. (2006). Shift work and inter-individual differences in sleep and sleepiness. *Chronobiol. Int.* 23, 1139–1147. doi: 10.1080/07420520601100971

Visser, I., Raijmakers, M. E. J., and Molenaar, P. C. M. (2002). Fitting hidden markov models to psychological data. *Sci. Program.* 10, 185–199. doi: 10.1155/2002/874560

Walsh, M., and Anderson, J. (2011). Learning from delayed feedback: neural responses in temporal credit assignment. *Cogn. Affect. Behav. Neurosci.* 11, 131–143. doi: 10.3758/s13415-011-0027-0

Walsh, M. M., and Anderson, J. R. (2014). Navigating complex decision spaces: problems and paradigms in sequential choice. *Psychol. Bull.* 140, 466–486. doi: 10.1037/a0033455

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.