Learning what is relevant for rewards via value-based serial hypothesis testing

Mingyu Song (mingyus@princeton.edu), Yael Niv (yael@princeton.edu)
Princeton Neuroscience Institute, Princeton University
Princeton, NJ 08540, United States

Ming Bo Cai (mingbo.cai@ircn.jp)
International Research Center for Neurointelligence, (WPI-IRCN), UTIAS, The University of Tokyo
Bunkyo City, Tokyo 113-0033, Japan

Abstract

Learning what is relevant for reward is a ubiquitous and crucial task in daily life, where stochastic reward outcomes can depend on an unknown number of task dimensions. We designed a paradigm tailored to study such complex scenarios. In the experiment, participants configured three-dimensional stimuli by selecting features for each dimension and received probabilistic feedback. Participants selected more rewarding features over time, demonstrating learning. To investigate the learning process, we tested two learning strategies, feature-based reinforcement learning and serial hypothesis testing, and found evidence for both. The extent to which each strategy was engaged depended on the instructed task complexity: when instructed that there were fewer relevant dimensions (and therefore fewer reward-generating rules were possible) people tended to serially test hypotheses, whereas they relied more on learning feature values when more dimensions were relevant. To explain the behavioral dependency on task complexity and instructions, we tested variants of the value-based serial hypothesis testing model. We found evidence that participants constructed their hypothesis space based on the instructed task condition, but they failed to use all the information provided (e.g. reward probabilities). Our current best model can qualitatively capture the difference in choice behavior and performance across task conditions.

Keywords: representation learning; reinforcement learning; serial hypothesis testing; active learning

Introduction

When interacting with a multidimensional environment, it is crucial to figure out what dimensions are relevant for obtaining rewards. For example, when purchasing coffee beans, a collection of decisions needs to be made including the brand, the packaging, the origin of the beans, the level they are roasted, etc. Among these dimensions, some determine the flavor of the coffee and how enjoyable it is (e.g. the origin and the roast level), while others (e.g. the brand and packaging) may matter less. An inexperienced coffee drinker can be clueless when facing these decisions, but after trying out different combinations, they will hopefully figure out what dimensions are relevant and which are not. Learning about relevance helps the agent make better decisions and allocate limited resources to useful information.

Determining the dimensions relevant for a task, however, can be challenging: outcomes may be stochastic, so learning requires aggregating over multiple experiences, and the number of relevant dimensions is often unknown, leaving learners uncertain as to whether they have fully learned. Few studies have considered both complexities (but see (Choung et al., 2017; Duncan et al., 2018)). Instead, in most multidimensional reinforcement learning (RL) tasks (Niv et al., 2015; Marković et al., 2015; Wunderlich et al., 2011), only one dimension of a stimulus is relevant for reward, and participants are explicitly informed so; in category learning tasks, rules often involve multiple dimensions, but they are often deterministic by design (Ballard et al., 2017; Mack et al., 2016). Therefore, we developed a task aimed at studying probabilistic reward learning about multiple (or even an unknown number of) relevant dimensions.

The "build-your-own-stimulus" task

In this task, stimuli are characterized by features in three dimensions: color ({red, green, blue}), shape ({square, circle, triangle}) and texture ({plaid, dots, waves}). In each game, a subset of the three dimensions was relevant for reward, meaning that one feature (compared to the other two) in each of these dimensions made stimuli more rewarding.

To earn rewards and figure out the most rewarding features (abbreviated as "rewarding features" from here on) in the relevant dimensions, participants were asked to configure stimuli by selecting features for each dimension (Figure 1). They could also leave any dimension empty, in which case the computer would randomly select a feature in that dimension. The participant then saw the resulting stimulus and received probabilistic reward feedback (one or zero points) based on the number of rewarding features in the stimulus (Table 1). Participants' goal was to earn as many points as possible over the course of each game.

Table 1: Probability of reward in different types of games, depending on number of rewarding features in the stimulus

Game type	# rewarding features			
	0	1	2	3
1D-relevant	20 %	80%	–	–
2D-relevant	20 %	50%	80%	–
3D-relevant	20 %	40%	60%	80%

Each game had 1-3 relevant dimensions (corresponding to 1D, 2D and 3D-relevant conditions), and this number was either known or unknown to participants ("known" and "unknown" conditions), resulting in six game types in total.

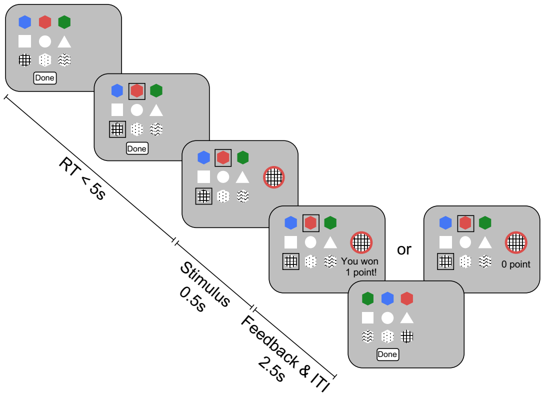

Figure 1: **The build-your-own-stimulus task.** Participants built stimuli by selecting a feature in each of 0-3 dimensions (marked by black squares). After hitting “Done”, the stimulus showed up on the screen, with features randomly determined for any dimension without a selection (here, circle was randomly determined). Reward feedback was then shown.

Compared to the multidimensional RL tasks and categorization tasks in the literature where stimuli (i.e. the combination of features) are often pre-determined and where it is hard to isolate the participants’ preference over single features, this task design enables us to directly probe participants’ preference (or lack thereof) in each of the three dimensions.

Participants.

27 participants recruited through Amazon Mechanical Turk each played all six types of games (3 games of each type, 30 trials per game). Participants were told that there could be one, two or three dimensions that are important for reward, and were explicitly informed about the reward probabilities in Table 1. In “known” games, the number of relevant dimensions was instructed before the start of the game. Participants were never told which dimensions were relevant or which features were more rewarding.

Learning performance and choice behavior.

Across all six game types, participants’ performance improved over the course of a game (Figure 2A). Games were harder (i.e. participants were less able to learn all the rewarding features) as the number of relevant dimensions increased; knowing the number helped performance when three dimensions were relevant (repeated measures ANOVA: $F(1,26) = 6.826, p = .002$), but not for one or two relevant dimensions.

Participants also showed distinct choice behavior in the different game types (Figure 2B): in “known” (number of relevant dimensions) games, they systematically selected more features on each trial as more dimensions were relevant; in “unknown” games, the number of selected features was not different between game types. In post-game surveys (not shown), participants correctly identified more rewarding features in 3D-relevant games in the “known” condition than “unknown”; they also were more likely to falsely identify an irrelevant feature as relevant in 1D-relevant games in the “unknown” condition than “known”.

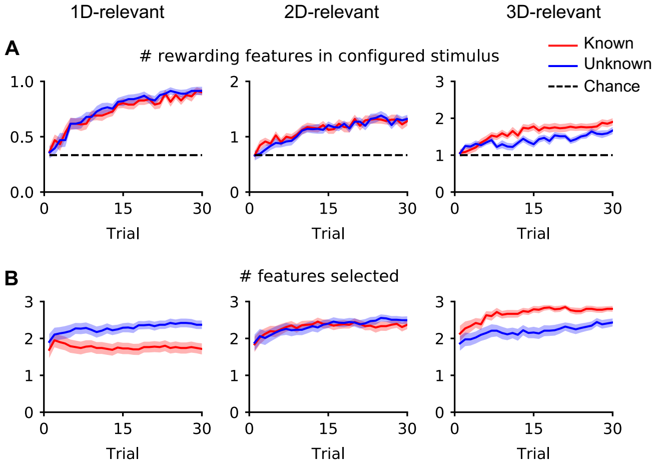

Figure 2: **Performance and choices by game type. (A)** The number of rewarding features in the configured stimulus, and **(B)** The number of features selected by the participants, over the course of 1D, 2D and 3D-relevant games (left, middle and right columns); red and blue curves represent the “known” and “unknown” conditions, respectively. Shaded areas represent 1 s.e.m. across participants. Dashed lines represent chance level for that type of game.

In sum, participants learned the task and performed better than chance, and their performance and choice behavior depended on game conditions.

A hybrid of two learning systems

There is extensive evidence supporting the existence of two learning systems in representation learning (Ashby & Maddox, 2005; Radulescu, Niv, & Ballard, 2019): an incremental learning system that learns the value of stimuli based on feedback from trial-and-error experiences, and a rule-based learning system that explicitly represents possible rules and evaluates them. Both learning strategies have been observed in tasks similar to the current one. For instance, in probabilistic reward learning tasks, people seem to learn via trial-and-error to identify relevant dimensions, and gradually focus their attention onto the rewarding features in those dimensions (Niv et al., 2015; Marković et al., 2015; Wunderlich et al., 2011). In contrast, in some types of categorization tasks, people seem to evaluate the probability of all possible rules via Bayesian inference, with a prior belief favoring simpler rules (Ballard et al., 2017). Inspired by these prior works, we test and compare both learning strategies.

Reinforcement learning model

First, we consider a feature-based reinforcement learning (RL) model, similar to the feature RL with decay model in

(Niv et al., 2015). It learns the values of nine features using Rescorla-Wagner updating, with separate learning rates for features that were selected by the participant ($\eta = \eta_s$) and those that were randomly determined ($\eta = \eta_r$). Values for the features not in the current stimulus s_t are decayed towards zero with a factor $d \in [0,1]$. η_s, η_r and d are free parameters.

$$V_t(f_{i,j}) = \begin{cases} V_{t-1}(f_{i,j}) + \eta(r_t - ER(c_t)), \text{ if } j = s_t^i \\ d \cdot V_{t-1}(f_{i,j}), \text{ if } j \neq s_t^i \end{cases} \quad (1)$$

where i and j index dimensions and features, respectively.

At decision time, the expected reward (ER) for each choice c is calculated as the sum of its feature values:

$$ER(c) = \sum_i V(f_{i,c^i}), \quad (2)$$

with the average value of all three features used for dimensions with no selected features.

The choice probability is then determined based on $ER(c)$ using a softmax function, with β as a free parameter:

$$P(c) = \frac{e^{\beta \cdot ER(c)}}{\sum_{c'} e^{\beta \cdot ER(c')}}. \quad (3)$$

Rule learning models

Unlike the value-based strategy that learns values for each feature independently and combines them additively at choice time, the rule-based strategy directly evaluates combinations of features. We considered each specification of the relevant dimension(s) and the corresponding rewarding feature(s) as a "rule". For "unknown" games, there were 63 possible rules in total; for "known" games, the total reduced to 9, 27 and 27 for 1D, 2D and 3D-relevant conditions, respectively.

There is little consensus on how people learn which rule is correct. One possibility is to consider all candidate rules, and use Bayes' rule to evaluate how likely each of them is; we term this a "Bayesian rule learning model". This model optimally utilizes feedback information to learn about candidate rules[1], and can serve as a reference model. However, Bayesian inference is computationally expensive and has a high memory load. A simpler alternative is serial hypothesis testing, with the assumption that people only test one rule at a time: if the evidence supports their hypothesis, they continue with that rule; otherwise, they switch to a different one, until the correct rule is found.

Bayesian rule learning model maintains a probabilistic belief distribution over all possible rules (denoted by h for hypotheses). After each trial, the belief distribution is updated according to Bayes' rule:

$$P(h|c_{1:t}, r_{1:t}) \propto P(r_t|h, c_t)P(h|c_{1:t-1}, r_{1:t-1}). \quad (4)$$

At decision time, the expected reward for each choice is calculated using the entire belief distribution:

$$ER(c) = \sum_h P(h)P(r|h,c). \quad (5)$$

The expected reward is then used to determine the choice probability as in Equation 3.

Serial hypothesis testing (SHT) models assume the participant is testing one hypothesis at any given time. We do not directly observe what hypothesis the participant is testing, and must infer that from their choices. We do so by using the change point detection model in (Wilson & Niv, 2012). The detailed math of this approach is beyond the page limit, but the basic idea is to infer the current hypothesis using all the choices the participant made so far (in the current game) and their reward outcomes (together denoted by $D_{1:t-1}$). Different variants of the model differ in the assumptions they make about the hypothesis testing and switching policies (i.e., whether to switch hypotheses and which next hypothesis to switch to, respectively; these two policies together determine the transition from the hypothesis on the last trial to the current one), and the choice policy (the probability of choice given the current hypothesis). Given this generative model of choices, we use Bayes' rule to calculate the posterior probability distribution over the current hypothesis h_t: $P(h_t|D_{1:t-1})$, and use this to predict the choice:

$$P(c_t|D_{1:t-1}) = \sum_{h_t} P(c_t|h_t)P(h_t|D_{1:t-1}) \quad (6)$$

Various choices can be made regarding the three policies, and the hypothesis space. As a baseline, we allow all $N_h = 63$ hypotheses in the hypothesis space, and consider the following hypothesis testing policy: On each trial, the participant estimates the reward probability of the hypothesis on last trial. With a uniform Dirichlet prior, this is equivalent to counting how many times they have been rewarded since they started testing this hypothesis. The estimated reward probability is then compared to a soft threshold θ to determine whether to stay with this hypothesis or to switch to a different one:

$$Pr(\text{stay}) = \frac{1}{1 + e^{-\beta_{\text{stay}}\left(\hat{P}_{\text{reward}} - \theta\right)}}, \quad (7)$$

where $\hat{P}_{\text{reward}} = \frac{\text{reward count} + 1}{\text{trial count} + 2}$ is the estimated probability of reward for hypothesis h_{t-1}, and β_{stay} and θ are free parameters. If the participant decides to switch, the model assumes that they will randomly switch to any other hypothesis:

$$P(h_t) = \begin{cases} Pr(\text{stay}), \text{ if } h_t = h_{t-1} \\ \frac{1}{N_h - 1}\left(1 - Pr(\text{stay})\right), \text{ if } h_t \neq h_{t-1} \end{cases} \quad (8)$$

Participants' choices are assumed to be aligned with their hypotheses most of the time, with a free-parameter lapse rate of λ.

We call this model the **random-switch SHT model**.

[1] We note that this model is nevertheless not strictly optimal, even with no decision noise, as it maximizes the reward on the current trial, but not the total reward.

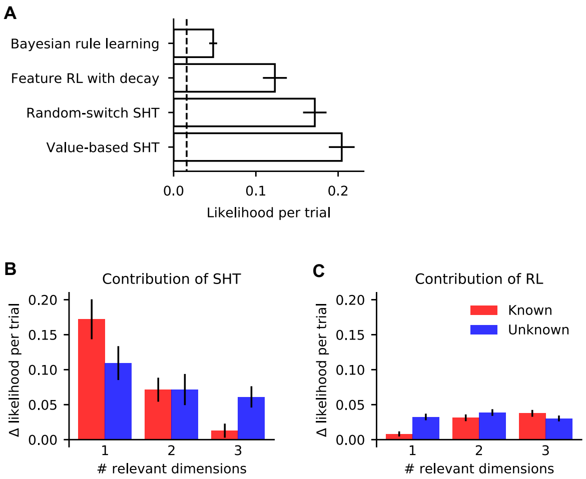

Figure 3: **Model comparison supports both learning strategies.** **(A)** Geometric average likelihood per trial for each model. Higher values indicate better model fits. Dashed lines indicate the chance level. **(B, C)** The difference in likelihood per trial between the hybrid value-based SHT model and **(B)** the feature RL with decay model (i.e. the contribution of serial hypothesis testing in the hybrid model), or **(C)** the random-switch SHT model (i.e. the contribution of feature value learning), by game type. Error bars represent 1 s.e.m. across participants.

Hybridizing the two learning systems

So far we have considered the two learning systems separately. However, they are not necessarily exclusive. We thus consider a hybrid model by incorporating the feature values into the switch policy of the serial hypothesis testing model. Rather than choosing a new hypothesis randomly, this model favors hypotheses that contain recently rewarded features. It maintains a set of feature values updated according to feature RL with decay as in Equation 1 (but with a single learning rate), and calculates the expected reward for each alternative hypothesis by adding up its feature values, similar to Equation 2 but for h instead of c. The probability of switching to $h_t \neq h_{t-1}$ is:

$$P(h_t) = (1 - Pr(\text{stay})) \frac{e^{\beta_{\text{switch}} \cdot ER(h_t)}}{\sum_{h' \neq h_{t-1}} e^{\beta_{\text{switch}} \cdot ER(h')}}, \tag{9}$$

where β_{switch} is a free parameter. We call this model the **value-based SHT model**.

Model fitting and model comparison

We fit the models using maximum likelihood estimation with the minimize function (L-BFGS-B algorithm) in Python package scipy.optimize with 10 random starting points. We performed leave-one-game-out cross-validation. Model fits were evaluated with cross-validated trial-by-trial likelihood.

Model comparison results are shown in Figure 3A. Among the four models, the Bayesian rule learning model, even though optimal in utilizing the feedback information, showed the worst fit to participants' choices. This is potentially due to the large hypothesis space (up to 63 hypotheses), making it implausible that participants performed exact Bayesian inference. Both the feature RL with decay model and the random-switch SHT model showed much better fit. Compared to the Bayesian model, both have lower storage and computational loads: the RL model takes advantage of the fact that different dimensions are independent and the reward probabilities are additive, by learning nine feature values individually and later combining them; the random-switch SHT model limits the consideration of hypotheses to one at a time. The hybrid value-based SHT model, which combines both learning systems, fit best, suggesting that participants used both strategies when solving this task.

Knowing that both learning systems were used in this task, the next questions is how they were combined in participants' strategies and how much each of them contributed. We address this question by comparing the hybrid model with the two component models, for each game condition separately: the additional likelihood per trial for the hybrid model as compared to each component is a proxy for the contribution of the other mechanism (Figure 3B and 3C). Our results show that participants' strategies were sensitive to task information. In "known" games, the contribution of hypothesis testing decreased with more relevant dimensions (estimated fixed effect slope -0.080 ± 0.011 in a mixed linear model with a random intercept, $p < .001$), and the contribution of value learning increased instead (estimated slope: 0.0147 ± 0.0032, $p < .001$). This suggests that when the task was known to have a smaller hypothesis space, participants were more likely to test one hypothesis at a time; whereas when the hypothesis space became larger, participants relied more on parallel learning of feature values, showing a strategic use of task information. In "unknown" games, in contrast, the contribution of both mechanisms differed less across game conditions (estimated slopes: -0.024 ± 0.010 for SHT, $p = .024$; -0.0010 ± 0.0025 for RL, $p = .674$).

Exploring the value-based serial hypothesis-testing strategy

In this experiment, human participants were sensitive to the game condition (Figure 2). Crucially, both their performance and choice behavior differed between "known" and "unknown" games with the same number of relevant dimensions. A reinforcement learning strategy that learns individual feature values is unable to predict these differences. Fortunately, the serial hypothesis testing strategy can potentially explain this finding. For instance, the hypothesis space can be constructed according to the instructed number of relevant dimensions, so that only a subset of hypotheses need to be considered in known games, potentially simplifying learning.

This having been said, the two minimalistic SHT models we have considered so far do not use game condition information, always having all 63 hypotheses in the hypothesis space. In this section, we explore the possibility of incorporating task instructions into the SHT models to capture the observed behavior differences. We also explore various choices for each of the assumptions the SHT models make (Figure 4), with the goal of explaining some more choice patterns we observed, and better accounting for participants' behavior.

To compare model variants, we use the value-based SHT model as the baseline model.

Model variants

Hypothesis space: Incorporating game information. In "known" games, participants were informed about the number of relevant dimensions, which could be used to limit the size of the hypothesis space. The extent to which people trust and follow instructions can vary. Thus, we parameterize the hypothesis space with two weight parameters w_l and w_h that were multiplied with the probability of hypotheses $P(h)$ in Equation 9 at hypothesis switch point:

$$P(h) \leftarrow \begin{cases} w_l P(h) & \text{if } D(h) < D \\ P(h) & \text{if } D(h) = D \\ w_h P(h) & \text{if } D(h) > D \end{cases} \qquad (10)$$

Here, $D(h)$ is the dimension of hypothesis h (how many rewarding features are specified in h), and D is the number of relevant dimensions according to the instruction. The baseline model can be seen as a special case of this variant with $w_l = w_h = 1$. If a participant follows the instruction exactly, $w_l = w_h = 0$. The average $P(h)$ of 1D, 2D and 3D games is used for "unknown" games.

Hypothesis testing policy: incorporating reward probability information. In the experiment, participants were informed of the reward probabilities for all game conditions, which is not used by the baseline model. One way to use such information is to calculate a "target" reward probability $RP_{\text{target}}(h|D, D(h))$, which can be achieved under the best case scenario if all features specified in the current hypothesis are rewarding (while not exceeding the instructed number of relevant dimensions D). In "known" games, we assume that participants set their threshold according to this "target" reward probability, with a free-parameter offset δ:

$$\theta = RP_{\text{target}}(h|D, D(h)) + \delta \qquad (11)$$

The intuition is that the participant should expect a higher reward probability, for example, when testing the same one-dimensional hypothesis in a 1D-known game compared to in a 3D-known game. The average RP_{target} of 1D, 2D and 3D games is used for "unknown" games.

Not always testing hypothesis. The baseline model assumes that people are always testing hypotheses, which might not be true. Since we did not enforce feature selection on any of the dimensions, the participant could choose not to select any feature, and let the computer configure a random stimulus. In fact, many participants did so, especially in the beginning of games (not shown). This was potentially due to not

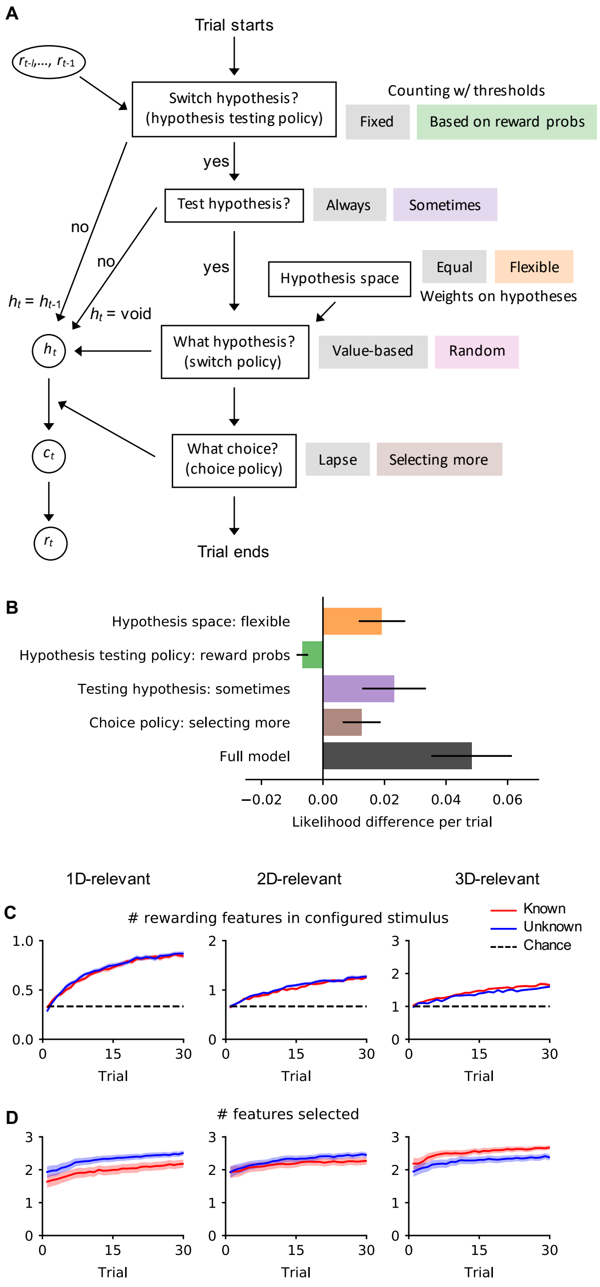

Figure 4: **Value-based serial hypothesis testing models: model variants and their comparison.** **(A)** A diagram of the SHT models. Model assumptions are presented in white boxes, accompanied by different variants on each assumption in colored boxes: in light gray are the assumptions adopted by the baseline model, and in other colors are those used in the model variants. **(B)** Difference in average likelihood per trial between variants of the SHT models and the baseline model (the value-based SHT model). All models except the full model are only different from the baseline model by one assumption as noted in the label; the full model adopts the better alternative in every assumption. Bar colors correspond to those in panel A, except for the full model (in dark gray). **(C, D)** Same as in Figure 2 but for simulation of the full model.

having a good candidate hypothesis in mind, either because little information had been obtained at the start of a game, or participants had trouble figuring out what features were good, which could happen at any point in a game. To model this inability to come up with hypotheses, we add a soft threshold on hypothesis testing: if the expected reward of the best candidate hypothesis is below a threshold θ_{test}, participants will be unlikely to test any hypothesis:

$$Pr(\text{test}) = \frac{1}{1+e^{-\beta_{\text{test}}(\max_h(ER(h))-\theta_{\text{test}})}} \quad (12)$$

β_{test} and θ_{test} are free parameters. This probability is applied to the first trial of a game and at hypothesis switch points.

Choice policy: selecting more than hypothesizing. In the baseline model, participants are assumed to make choices aligned with their current hypothesis, unless they have a lapse. In the experiment, however, we observed an overall tendency to select more features than instructed (Figure 2B). This was not surprising as there was no cost for selecting more features. In fact, it is strictly optimal to always make selections on all dimensions, as there is always a best feature (at least equally good as the other two) within each dimension according to the participant's mental model. Thus, we allow in the model for participants to select more features than what are in their current hypothesis, and parameterize the extent to which they do so with a free parameter k: if c_t is more complex than or the same as h_t, $P(c_t|h_t) \propto (1-\lambda)e^{k(D(c_t)-D(h_t))}$; otherwise, $P(c_t|h_t) \propto \lambda$.

Model comparison results

All the variants improved model fits, except for the hypothesis testing policy that incorporates the reward probability information (Figure 4B). These results suggest that participants made use of the task instructions to form their hypothesis space, but may not have used the reward probability information in evaluating the hypotheses. When there was not enough evidence for a good hypothesis, they would not engage in hypothesis testing. They also seemed to choose more features than what they had in their hypotheses.

The full model, which takes the better alternative on all assumptions, was able to qualitatively capture the dependency of both choices and performance on task condition (model simulation: Figure 4C and 4D; data: Figure 2). The effects predicted by the model, however, are smaller than those observed in the data, including the performance difference between "known" and "unknown" conditions for 3D-relevant games, and the choice differences between "known" and "unknown" conditions for 1D- and 3D-relevant games.

Summary

We designed a novel "build-your-own-stimulus" task to study probabilistic reward learning of multi-dimensional stimuli when the underlying rules involve multiple or an unknown number of relevant dimensions. Participants were able to learn over time and performed above chance level across all task conditions. Their strategies and performance were sensitive to both the complexity of the underlying rules and whether they were given this information explicitly. We found evidence for the use of two learning strategies, namely feature-value learning and serial hypothesis testing. When deciding which one to rely on more, participants took advantage of the task complexity information. We also explored various ways the value-based serial hypothesis testing models can incorporate task instructions, and different model assumptions. The current best model is able to qualitatively capture how choice behavior and performance depended on task conditions.

Acknowledgment

This work was supported by the National National Institute of Drug Abuse (Grant R01DA042065) and Army Research Office (Grant W911NF-14-1-0101). MBC was supported by World Premier International Research Center Initiative (WPI), MEXT, Japan.

References

Ashby, F. G., & Maddox, W. T. (2005). Human category learning. *Annu. Rev. Psychol.*, *56*, 149–178.

Ballard, I., Miller, E. M., Piantadosi, S. T., Goodman, N. D., & McClure, S. M. (2017). Beyond reward prediction errors: Human striatum updates rule values during learning. *Cerebral Cortex*, *28*(11), 3965–3975.

Choung, O.-h., Lee, S. W., & Jeong, Y. (2017). Exploring feature dimensions to learn a new policy in an uninformed reinforcement learning task. *Scientific reports*, *7*(1), 17676.

Duncan, K., Doll, B. B., Daw, N. D., & Shohamy, D. (2018). More than the sum of its parts: a role for the hippocampus in configural reinforcement learning. *Neuron*, *98*(3), 645–657.

Mack, M. L., Love, B. C., & Preston, A. R. (2016). Dynamic updating of hippocampal object representations reflects new conceptual knowledge. *Proceedings of the National Academy of Sciences*, *113*(46), 13203–13208.

Marković, D., Gläscher, J., Bossaerts, P., O'Doherty, J., & Kiebel, S. J. (2015). Modeling the evolution of beliefs using an attentional focus mechanism. *PLoS computational biology*, *11*(10), e1004558.

Niv, Y., Daniel, R., Geana, A., Gershman, S. J., Leong, Y. C., Radulescu, A., & Wilson, R. C. (2015). Reinforcement learning in multidimensional environments relies on attention mechanisms. *Journal of Neuroscience*, *35*(21), 8145–8157.

Radulescu, A., Niv, Y., & Ballard, I. (2019). Holistic reinforcement learning: the role of structure and attention. *Trends in cognitive sciences*.

Wilson, R. C., & Niv, Y. (2012). Inferring relevance in a changing world. *Frontiers in human neuroscience*, *5*, 189.

Wunderlich, K., Beierholm, U. R., Bossaerts, P., & O'Doherty, J. P. (2011). The human prefrontal cortex mediates integration of potential causes behind observed outcomes. *Journal of neurophysiology*, *106*(3), 1558–1569.