Learning what is relevant for rewards via serial hypothesis testing

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

Ming Bo Cai (mcai@princeton.edu), Yael Niv (yael@princeton.edu)
Princeton Neuroscience Institute and Department of Psychology, Princeton University
Princeton, NJ 08540, United States

Abstract

Living in a world where any object bears features in many dimensions, it is crucial but also challenging for humans to figure out what dimensions are relevant for rewards. How do humans learn from trial and error to obtain rewards when multiple (or an unknown number of) dimensions need to be taken into account, and feedback is probabilistic? In this work, we designed a paradigm tailored to study such complex but naturalistic scenarios. In the experiment, participants configured three-dimensional stimuli by selecting features for each dimension and received probabilistic feedbacks. Participants demonstrated learning, selecting more rewarding features over the course of a game. To investigate their learning process, we compared three classes of learning models: a Bayesian model, reinforcement learning models and serial hypothesis testing models, and found evidence supporting the latter. This suggests that when facing complex learning scenarios with a great number of possible rules, people tend to actively test one hypothesis at a time, as opposed to evaluating all the possibilities or learning values of all features incrementally.

Keywords: reinforcement learning; representation learning; serial hypothesis testing

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 much a person likes it (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 a few times trying out different combinations, they will hopefully figure out what dimensions are relevant for obtaining a tasty coffee and which are not. Learning about relevance is useful as it helps the agent make better decisions, as well as allocate limited resources to the useful information (e.g. buying the cheaper brand if it does not affect the flavor).

Finding the dimensions relevant for a task, however, can be challenging: the outcomes may be stochastic, so learning requires aggregating over multiple experiences; 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, here we developed a "build-your-own-stimulus" task, aiming to study probabilistic reward learning about multiple (or even an unknown number of) relevant dimensions.

Experiment

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 is relevant for reward; being a relevant dimension means that one feature within this dimension makes a stimulus more rewarding.

To earn rewards and learn what are the most rewarding features in the relevant dimensions, we asked participants to configure stimuli by choosing what features to use in each dimension (Figure 1). They could also choose to not select features on any of the dimensions; in that case, a random feature would be selected by the computer. The participant would then see the resulting stimulus and receive probabilistic reward feedback (one or zero points): the more rewarding features the stimulus contained, the higher the probability of reward. Participants' goal was to earn as many points as possible over the course of each game.

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.

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

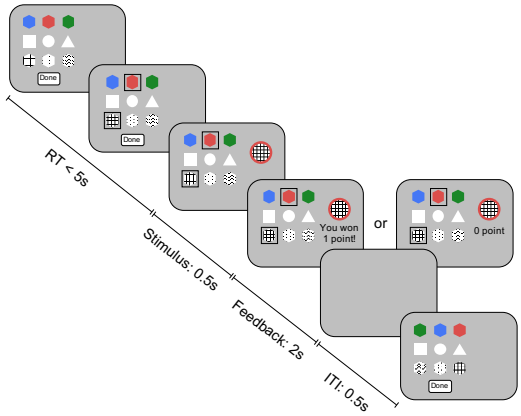

Figure 1: **The build-your-own-stimulus task.** The participant builds a stimulus by choosing one feature out of three for each of three dimensions (features chosen are marked with black squares); they can also decide not to choose a feature for any of the dimensions. After hitting "done", the stimulus shows up on the screen, with random features selected for any dimension that the participant did not make a choice on (in this example, "circle" is randomly selected for the color dimension). Then the reward feedback is shown on the screen.

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 informed about the reward probabilities: if one dimension is relevant, they will get a point 80% of the time if their stimulus contains the rewarding feature, and 20% of the time otherwise; if two dimensions are relevant, they will get a point 80%, 50% or 20% of the time, if the stimulus contains two, one or zero rewarding features; if three dimensions are relevant, they will get a point 80%, 60%, 40% or 20% of the time for three, two, one or zero rewarding features. In "known" games, they were instructed on the number of relevant dimensions. Participants were never told which dimensions were relevant or which features were more rewarding.

Learning performance. Across all six game types, participants' performance improved over the course of a game (Fig. 2). Games were harder (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 ($p = .002$, repeated measures ANOVA), but not for one or two relevant dimensions.

Models

In multidimensional RL tasks, people have been shown to learn via trial-and-error to identify relevant dimensions, and to gradually focus their attention onto the rewarding features in those dimensions (Niv et al., 2015; Marković et al., 2015; Wunderlich et al., 2011). In a series of studies with one relevant dimension (Niv et al., 2015; Leong et al., 2017), an RL model that updates values of chosen features based on reward prediction errors and decays the values of unchosen features was found to fit best to the participants' behavior, compared to alternative models such as Bayesian inference and serial hypothesis testing. In categorization tasks, in contrast, people seem to use a Bayesian rule-learning strategy, evaluating the probability of all possible rules via Bayesian inference, with a prior belief favoring simpler rules (Ballard et al., 2017).

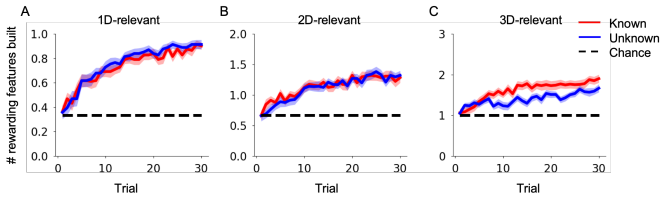

Figure 2: **Learning curves by game type.** The number of rewarding features in the configured stimuli over the course of 1D, 2D and 3D-relevant games; red and blue curves represent the "known" and "unknown" conditions, respectively. Shaded areas represent 1 s.e.m. across participants. Dashed lines represent chance level for that type of game.

Here we ask how people learn about what is relevant for reward in a more complex scenario, and whether and how their strategies are affected by (1) the number of relevant dimensions, and (2) whether they know the number. Inspired by prior work, we consider three classes of models: Bayesian rule-learning, reinforcement learning (RL) and serial hypothesis testing (SHT).

The Bayesian rule-learning model

This model maintains a probabilistic belief distribution over all possible rules. A rule specifies the relevant dimension(s) and the corresponding rewarding feature(s). For "unknown" games, there are 63 possible rules in total; for "known" games, it reduces to 9, 27 and 27 for 1D, 2D and 3D-relevant conditions, respectively. At the start of a game, the belief distribution is initialized uniformly. After each trial, it is updated according to Bayes' rule:

$$P(h|a_{1:t}, r_{1:t}) \propto P(r_t|h, a_t)P(h|a_{1:t-1}, r_{1:t-1}), \quad (1)$$

where h represents one hypothesis (i.e. rule), a_t and r_t are the choice and outcome on trial t, and the likelihood term $P(r_t|h, a_t)$ is given by the instructions.

The Bayesian model predicts choices by calculating the expected reward for each choice given the belief distribution:

$$ER(a) = \sum_h P(h)P(r|h, a), \quad (2)$$

The expected values are then put through a softmax function to determine the choice probability, with an additional cost term proportional to the number of features selected by the participant (representing the motor cost of selecting a feature):

$$P(a) = \frac{e^{\beta(ER(a) - c \cdot \sum_i \delta_i(a))}}{\sum_{a'} e^{\beta(ER(a') - c \cdot \sum_i \delta_i(a'))}}, \quad (3)$$

where $\delta_i(a)$ is an indicator function for whether or not a feature was selected on the ith-dimension for choice a. The Bayesian model thus has two free parameters: β and c.

Reinforcement learning models

The feature RL model. This model learns the value of nine features ($f_{i,j}$, where i indexes dimension and j indexes feature in that dimension) using a Rescorla-Wagner update rule,

$$V_t(f_{i,j}) = V_{t-1}(f_{i,j}) + \eta(r_t - V_{t-1}(f_{i,j})), \tag{4}$$

with one learning rate ($\eta = \eta_s$) for features the participant selects and another ($\eta = \eta_r$) for randomly selected features. Features not in the current stimulus are not updated.

The expected reward for each choice a is then the sum of its feature values:

$$ER(a) = \sum_i V(f_{i,a^i}) \tag{5}$$

where $a = (a^1, a^2, a^3)$, with a^i being the choice on the i-th dimension. If no feature is chosen on a dimension ($a^i = \text{null}$), the average of the three features on that dimension is used: $V(f_{i,a^i=\text{null}}) = \frac{1}{3}\sum_j V(f_{i,j})$. The choice is then determined in the same way as in the Bayesian model (Equation 3). This model has four free parameters: η_s, η_r, β and c.

The feature RL with decay model. This model is identical to the feature RL model, except with an additional decay parameter d that multiplies the values of features not in the current stimulus:

$$V_t(f_{i,j}) = d \cdot V_{t-1}(f_{i,j}), \text{ if } j \neq s_t^i \tag{6}$$

with s_t^i indexing the feature on dimension i of the current stimulus. This model has five free parameters: η_s, η_r, β, c and d.

Serial hypothesis testing models

In serial hypothesis testing (SHT) models, we assume the participant's choice reflects the current hypothesis they are testing, i.e. any change in choice marks a switch in the hypothesis tested. In addition to the 63 hypotheses mentioned before, we also include a hypothesis with no identified rewarding features, corresponding to not selecting any feature on any dimension.

On each trial, the participant decides whether to stay with the current hypothesis or switch to a different one, based on the estimated reward probability of the current hypothesis. Assuming a uniform Dirichlet prior, this is equivalent to counting how many times the participant was rewarded since they started testing the current hypothesis. The estimated reward probability is then compared to a soft threshold θ to determine the stay probability, with the same cost proportional to number of features selected:

$$Pr(\text{stay}) = \frac{1}{1 + e^{-\beta_{\text{stay}}\left(\frac{\text{reward count}+1}{\text{trial count}+2} - \theta\right) - c_{\text{stay}} \cdot \sum_i \delta_i(a_{t-1})}} \tag{7}$$

The switch probability is $1 - Pr(\text{stay})$. The two SHT models differ on their switch policies.

The random-switch SHT model. Here, the switch probability is uniformly distributed to all choices other than the current one, with a penalty for selecting more features. Therefore the choice policy is

$$P(a_t) = \begin{cases} Pr(\text{stay}), \text{ if } a_t = a_{t-1} \\ (1 - Pr(\text{stay})) \frac{e^{-c_{\text{switch}} \cdot \sum_i \delta_i(a_t)}}{\sum_{a' \neq a_{t-1}} e^{-c_{\text{switch}} \cdot \sum_i \delta_i(a')}}, \text{ if } a_t \neq a_{t-1} \end{cases} \tag{8}$$

This model has four free parameters: θ, β_{stay}, c_{stay} and c_{switch}.

The value-based SHT with reset model. Instead of randomly switching to any other hypothesis, this model favors recently rewarded features. It maintains a set of feature values using the Rescorla-Wagner rule (as in Equation 4), and calculates the expected reward for each alternative hypothesis as in Equation 5. The choice probability for $a_t \neq a_{t-1}$ is thus:

$$P(a_t) = (1 - Pr(\text{stay})) \frac{e^{\beta_{\text{switch}} \cdot ER(a_t) - c_{\text{switch}} \cdot \sum_i \delta_i(a_t)}}{\sum_{a' \neq a_{t-1}} e^{\beta_{\text{switch}} \cdot ER(a') - c_{\text{switch}} \cdot \sum_i \delta_i(a')}} \tag{9}$$

After switching hypotheses, feature values are reset to zero. This model has seven free parameters: θ, η_s, η_r, β_{stay}, c_{stay}, β_{switch} and c_{switch}.

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 3-fold stratified cross-validation: each fold of data contains six games (one game of each type); we fit the models to two folds of data and calculated the log likelihood on the third fold using the fit parameters. The log likelihoods of all three folds were summed to get the cross-validated log likelihood. This procedure was repeated 10 times and results were averaged to reduce noise.

Model comparison results

Across the three classes of models, the serial hypothesis testing models fit participants' behavior best, followed by RL models, and then the Bayesian rule-learning model (Fig 3A, B).

In contrast to the RL and SHT models that do not utilize task instructions, the Bayesian model exploits the complete knowledge of the task (including the game conditions and the reward probabilities). However, it did poorly in predicting participants' choices. This was potentially due to the large hypothesis space (up to 63 hypotheses), making it implausible that participants actually performed exact Bayesian inference.

The RL models, in contrast, take advantage of the facts that different dimensions are independent and the reward probabilities are additive. The models reduce both the storage and computational loads by learning nine feature values. The feature RL model predicted data better than the Bayesian model; additionally, the decay mechanism greatly helped model fits, consistent with findings in a related task on the 1D-known condition (Niv et al., 2015). The decay mechanism means that experiences in the far past will not affect the current choice,

which Niv et al. (2015) interpreted as "forgetting". It can also be seen as a way to reduce the storage load: the values of features not considered for a while are decayed to zero, and do not need to be memorized by the agent.

Extending this idea of only using recent history to the extreme gives rise to the SHT models: when deciding whether to stay with the current hypothesis or to switch, the agent only utilizes evidence within the current hypothesis window and discards all previous experiences. The SHT models fit at least as well as the feature RL with decay model, even with a random-switch policy. The model fits further improved when we allowed the SHT model to learn a set of feature values and use them to guide the switching of hypotheses. Interestingly, the value-based SHT with reset model requires no less memory and computation compared to the feature RL model, but fit behavior better. This suggests that the poor fit of the feature RL model was not due to limits on storing and updating all feature values, but rather due to participants' learning strategies being closer to actively testing one hypothesis at a time, compared to learning feature values in parallel. The feature RL with decay model accounted for data relatively well potentially because it "mimics" the SHT by favoring recent experiences.

Broken down for the six game conditions, the best model (value-based SHT with reset) could best predict 1D-relevant games, and less so for 2D or 3D-relevant games (Figure 3C). Additionally, its advantage over the feature RL with decay model was most prominent in 1D-relevant games, especially 1D-known games (Figure 3D). Together, these results indicate that participants' strategies were more hypothesis-based when the game was easier (i.e. fewer rewarding features to search for), and less so when the game was harder; in fact, the evidence for serial hypothesis testing was similar to that for incremental learning in 3D-known games.

Conclusions and Future Directions

We developed a novel task to investigate how people learn about relevance in a multidimensional environment, and found evidence for a serial hypothesis testing strategy.

Future directions include (1) deriving the optimal strategy for this task: participants were able to perform above chance-level but were still far from ceiling performance; knowing the optimal strategy will help understand and evaluate human performance; (2) developing models that incorporate task instructions: apart from the Bayesian model, all other models did not make use of task instructions (e.g. game condition, reward probabilities), and were thus unable to explain differences in behavior under different game conditions (e.g. Figure 2C).

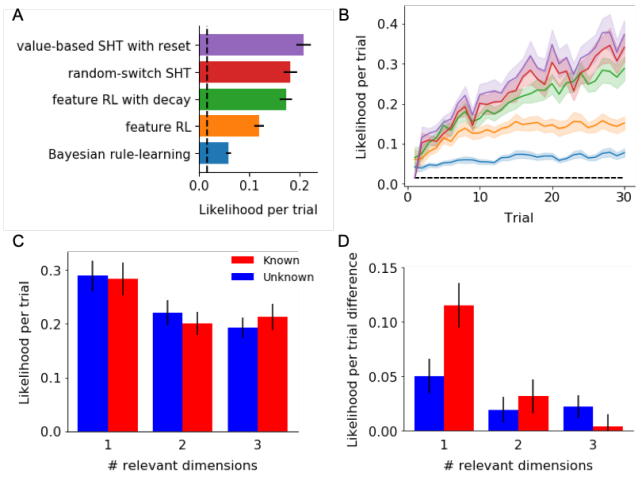

Figure 3: **Model comparison results. (A)** Likelihood per trial for all the models. Higher values indicate better model fit. **(B)** Likelihood per trial over the course of the game. Colors are as in A. **(C)** Likelihood per trial of the best-fitting value-based SHT with reset model, and **(D)** the difference between this model and the feature RL with decay model, for each of the six conditions. Error bars represent 1 s.e.m. across participants. Dashed lines indicate chance level. Likelihood per trial is calculated as the geometric mean of trial-wise cross-validated likelihood.

References

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.

Leong, Y. C., Radulescu, A., Daniel, R., DeWoskin, V., & Niv, Y. (2017). Dynamic interaction between reinforcement learning and attention in multidimensional environments. *Neuron*, *93*(2), 451–463.

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.

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.