MiniConfMethods

ciranka

December 2021

Additional Methods: Modelling

In order to better understand the specific mechanisms of learning that change over lifespan, we conducted a series of model-based analyses using RL models. We first compared models in their ability to predict out-of-sample choices and their ability to simulate human-like learning curves across the various age groups Next, we analyzed the parameters of the winning model to understand how three different psychological mechanisms of learning change over the lifespan: generalization (λ; Eq. 2), uncertainty directed exploration (β; Eq. 3), and random exploration (τ; Eq. 4). Lastly, we conducted model simulations across a wide range of plausible parameter values, in order to construct a fitness landscape of different learning strategies. We then compared the developmental trajectory of our participants to that of a stochastic optimization algorithm.

RL models

We compare two main categories of RL models (see Methods for details). A *Bayesian Mean Tracker* (BMT) was chosen as a traditional RL model that learns reward estimates for each option independently, and can be interpreted as probabilistic variant of the classic Rescorla-Wagner model [**?**, **?**], which updates beliefs about reward expectations as a function of prediction error. In contrast, we use *Gaussian process* (GP) regression [**?**] as a model of value generalization, where observations of past rewards also generalize to novel options, proportional to distance, where closer options exhibit a larger influence. Thus, whereas the BMT defaults to a prior for options that have not yet been observed, the GP can form predictive generalizations about novel options based on past observations.

Both the BMT and GP models use Bayesian principles, which allows them to compute posterior predictions about the rewards for some option $\mathbf{x}$ (encoding Cartesian coordinates on the grid) conditioned on previously observed data $\mathcal{D}_t = \{\mathbf{X}_t, \mathbf{y}_t\}$ of choices $\mathbf{X}_t = [\mathbf{x}_1, \ldots, \mathbf{x}_t]$ and reward observations $\mathbf{y}_t = [y_1, \ldots, y_t]$ at time t:

$$p(r_t(\mathbf{x})|\mathcal{D}) \sim \mathcal{N}(m_t(\mathbf{x}), v_t(\mathbf{x})). \tag{1}$$

The posterior predictions in Eq. 1 are Gaussian distributions for both models, allowing them to be characterized by posterior mean $m_t(\mathbf{x})$ and posterior variance $v_t(\mathbf{x})$ (see Eqs. 7-8 and Eqs. - for details)..

And whereas the BMT posterior reward estimates depend solely on previous observations of the same option, the GP predictions use a kernel function to describe how observations from one option generalize to another as a function of their distance. Specifically, we use a Radial Basis Function (RBF) kernel to model the covariance structure of rewards,

$$k(\mathbf{x}, \mathbf{x}') = exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right), \tag{2}$$

described as an exponentially decaying function of the squared Euclidean distance between options on the grid. Put simply, nearby options are assumed to have similar rewards, with the level of similarity decreasing exponentially over increased distances. The lengthscale λ is treated as a free parameter describing the rate at which generalization decays, which we use to model the extent to which participants generalize. Larger estimates of λ thus correspond to stronger generalization over greater distances.

In order to predict choices, we combine BMT and GP as two candidate learning models, together with three different behavioral policies (described below) to produce a total of six different models. *Upper Confidence Bounds* (UCB) sampling describes a policy by using a weighted sum of expected mean reward $m(\mathbf{x})$ and uncertainty $v(\mathbf{x})$ to estimate a valuation for each option $\mathbf{x}$:

$$UCB(\mathbf{x}) = m(\mathbf{x}) + \beta\sqrt{v(\mathbf{x})}. \tag{3}$$

β is the exploration parameter, which determines to what extent uncertainty is valued positively, thus capturing the trade-off between exploring uncertain options vs. exploiting known high-value options. We use estimates of β to quantify the level of uncertainty-directed exploration in participants. We can also decompose UCB sampling into two separate policies. A *Greedy Mean* (GM) policy only values exploiting known high-value options, $GM(\mathbf{x}) = m(\mathbf{x})$, whereas a *Greedy Variance* (GV) policy only values exploring options with high variance $GV(\mathbf{x}) = \sqrt{v(\mathbf{x})}$. These two greedy policies are used to show that the predictions of the UCB are more than the sum of its parts.

Lastly, a softmax function takes the valuation $q(\mathbf{x}_j)$ computed by the UCB, GM, or GV policies, and translates them into choice probabilities:

$$p(\mathbf{x}) = \frac{exp(q(\mathbf{x})/\tau)}{\sum_{\mathbf{x}' \in \mathcal{X}} exp(q(\mathbf{x}')/\tau)} \tag{4}$$

The temperature parameter τ controls the amount of random exploration. As $\tau \to \infty$, all options will have uniform probability, while $\tau \to 0$ corresponds to an argmax. We use estimates of τ to quantify the level of random exploration in participants, which is both distinct and recoverable (Fig.) from the uncertainty-directed exploration parameter β.

Even more detail

Bayesian mean tracker

The Bayesian mean tracker (BMT) is a type of Kalman filter, but with the assumption of time-invariant rewards, as in the experiment. For each option $\mathbf{x}$ (describing the 2-D spatial coordinates), the BMT independently defines a Gaussian prior distribution of the reward expectations

$$p\big(r_0(\mathbf{x})\big) \sim \mathcal{N}\big(m_0(\mathbf{x}), v_0(\mathbf{x})\big), \tag{5}$$

which are characterized by prior mean $m_0(\mathbf{x})$ and variance $v_0(\mathbf{x})$.

Conditioned on a set of observations $\mathcal{D}_t = \{\mathbf{X}_t, \mathbf{y}_t\}$ of choices $\mathbf{X}_t = [\mathbf{x}_1, \ldots, \mathbf{x}_t]$ and reward observations $\mathbf{y}_t = [y_1, \ldots, y_t]$ at time t, the BMT can compute a posterior distribution for the expected reward $r_{j,t}$ for each option, also in the form a Gaussian:

$$P\big(r_t(\mathbf{x}|\mathcal{D}_t\big) = \mathcal{N}\big(m_t(\mathbf{x}), v_t(\mathbf{x})\big), \tag{6}$$

characterized by posterior mean $m_{j,t}$ and posterior variance $v_{j,t}$. This posterior can be computed iteratively by updating the mean and variance following:

$$m_{t+1}(\mathbf{x}) = m_t(\mathbf{x}) + \delta_t(\mathbf{x})G_t(\mathbf{x})\big(y_t(\mathbf{x}) - m_t(\mathbf{x})\big) \tag{7}$$

$$v_{t+1}(\mathbf{x}) = v_t(\mathbf{x})\big(1 - \delta_t(\mathbf{x}))G_t(\mathbf{x})\big) \tag{8}$$

Both updates use $\delta_t(\mathbf{x}) = 1$ if option j was chosen on trial t, and $\delta_t(\mathbf{x}) = 0$ otherwise. Thus, the posterior mean and variance are only updated for the chosen option. The update of the mean is based on the prediction error $y_t(\mathbf{x}) - m_t(\mathbf{x})$ between observed and anticipated reward, while the magnitude of the update is based on the Kalman gain $G_t(\mathbf{x})$:

$$G_t(\mathbf{x}) = \frac{v_t(\mathbf{x})}{v_t(\mathbf{x} + \theta_\epsilon^2)}, \tag{9}$$

similar to the learning rate of the Rescorla-Wagner model. Here, the Kalman gain is dynamically defined as a ratio of variance terms, where $v_{j,t}$ is the posterior variance estimate and θ_ϵ^2 is the error variance, which models the level of noise associated with reward observations. Smaller values of θ_ϵ^2 thus result in larger updates of the mean.

Gaussian process regression

Gaussian process (GP) regression[**?**] provides a non-parametric Bayesian framework for function learning. We use the GP to infer a value functions $f : \mathcal{X} \rightarrow R^n$ mapping input space $\mathcal{X}$ (all possible options on the grid) to a real-valued scalar outputs (reward expectation). The GP performs this inference in a Bayesian manner, by first defining a prior distribution over functions $p(f)$, which is assumed to be multivariate Gaussian:

$$f \sim \mathcal{GP}\left(m\left(\mathbf{x}\right),\, k\left(\mathbf{x},\mathbf{x}'\right)\right), \tag{10}$$

with the prior mean $m(\mathbf{x})$ defining the expected output of input $\mathbf{x}$, and with covariance defined by the kernel function $k(\mathbf{x},\mathbf{x}')$, for which we use an RBF kernel (Eq. 2). Per convention, the prior mean is often set to zero, as in our model, without loss of generality[**?**].

Conditioned on a set of observations $\mathcal{D}_t = \{\mathbf{X}_t, \mathbf{y}_t\}$, the GP computes a posterior distribution $p(f(\mathbf{x}_*)|\mathcal{D}_t)$ for some new input $\mathbf{x}_*$, which is also Gaussian, with posterior mean and variance defined as: $\mathrm{m}(\mathrm{x}_*|\mathcal{D}_t) = \mathbf{k}_{*,t}^\top(\mathbf{K} + \sigma_\epsilon^2\mathbf{I})^{-1}\mathbf{y}_t$ $v(\mathbf{x}_*|\mathcal{D}_t) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_{*,t}^\top(\mathbf{K} + \sigma_\epsilon^2\mathbf{I})^{-1}\mathbf{k}_{*,t}$ $\mathbf{k}_{*,t} = k(\mathbf{X}_t, \mathbf{x}_*)$ is the covariance matrix between each observed input and the new input $\mathbf{x}_*$ and $\mathbf{K} = k(\mathbf{X}_t, \mathbf{X}_t)$ is the covariance matrix between each pair of observed inputs. $\mathbf{I}$ is the identity matrix and σ_ϵ^2 is the observation variance, corresponding to assumed i.i.d. Gaussian noise on each reward observation.

Model cross-validation

Combining the two learning models (BMT and GP) with the three policies (UCB, GM, and GV), produced a total of six models. Each model was fit using leave-one-round-out cross validation for each participant individually. A maximum likelihood estimate (MLE) was iteratively obtained on a training set with a single round omitted, and then used to make predictions on the held-out round. Model fits are described using the out-of-sample prediction accuracy, while individual participant parameter estimates are based on aggregating over the cross-validated MLEs.

Figure **??**a reports model fits in terms of a pseudo-R^2, which compares the negative log likelihood (nLL) for each model k against a random model:

$$R^2 = 1 - \frac{log\mathcal{L}(M_k)}{log\mathcal{L}(M_{random})} \tag{11}$$

Protected Exceedance Probability

We used a hierarchical Bayesian model selection framework [**?**, **?**] to compute the protected exceedance probability (pxp). Intuitively, one can imagine an urn containing differently colored marbles, representing a distribution of different models in a population. The probability of each model in the population is thus assumed to be fixed but unknown. $pxp(m_k)$ defines the probability that model

k is more frequent in the population than all other models under consideration, after correcting for chance. We first describe the posterior probability $p(m_k|\mathbf{y})$ of model k given data $\mathbf{y}$ (i.e., model evidence), modeled as a Dirichlet distribution, where the parameters are estimated based on the summed cross-validated nLL for each model and participant using variational Bayes. This allows us to compute the exceedance probability (xp)

$$xp(m_k) = p(r_{m_k} > r_{m_{k' \neq k}}|\mathbf{y}) \tag{12}$$

defining the posterior probability that the frequency r of model k is larger than that of all other models under consideration.

Protected exceedance probability corrects for chance using the Bayesian Omnibus Risk (BOR), which is the posterior probability that all models are equally frequent:

$$pxp(m_k) = pxp(m_k)(1 - BOR) + \frac{BOR}{K} \tag{13}$$

with K indicating the total number of models considered[**?**].