Supplementary Material

Additional details of multi-step-ahead adaptive estimation of psychometric thresholds through dynamic programming

The Bayesian adaptive estimation of psychometric thresholds is extended to use longer-horizon look-ahead optimization by dynamic programing through the following steps.

1. Define the Bayesian model

Give stimulus *d*, the Bayesian model of response *y* and underlying psychometric threshold θ in a two-alternative-forced-choice detection task is defined by the joint probability distribution:

$$p(y,\theta \mid d) = p(y \mid \theta, d)\, p(\theta),$$

where

$$p(y \mid \theta, d) = \begin{cases} \Psi_\theta(d) & \text{if } y = 1 \\ 1 - \Psi_\theta(d) & \text{if } y = 0 \end{cases} \quad \text{and} \quad p(\theta) \propto \exp\left(-\frac{(\theta - \mu)^2}{2\sigma^2} \right).$$

Here, the psychometric function $\Psi_\theta(d)$ is given in Equation 1 and the prior distribution $p(\theta)$ is defined as a diffuse Gaussian distribution by selecting values of μ and σ that are suitable for the task domain.

2. Define the stimulus, parameter, and state spaces

The application of constrained backward induction to Bayesian adaptive estimation of thresholds involves representing three distinct variables on discretized continuums: (a) stimulus intensity *d*, (b) threshold θ, and (c) posterior distribution of θ. The scales of *d* and θ depend on a task domain. In our simulation study, assuming the task of detecting visual objects with varying contrasts, possible values of d and θ were sampled, respectively, from 0.05% to 98% and from 0.1% to 90%, and stored in 120-by-1 and 100-by-1 vectors. Possible shapes of the posterior distribution of θ were modeled parametrically by Equation 4 and its parameter space was discretized on a three-dimensional grid with a total of 23,331 grid points. The bounds, resolution, and sampling scheme of the grid were carefully determined by referring to the results of threshold estimation in empirical and simulated settings. With these key variables discretized,

grid-based computations are employed to evaluate numerical integration and maximization required in the following steps.

3. Precompute the matrix of local state transitions and rewards

From the standpoint of dynamic programming, Bayesian posterior updating is the transition of states from a trial to the next. Because all possible posteriors (including the prior at the start of a measurement session) are approximated on a discretized space, any transition from a given posterior to another can be mapped within the space, given a response to a chosen stimulus. Since stimuli are selected from 120 possible values and responses are binary (correct or incorrect detection), all possible transitions can be represented in a 23,331 × 120 × 2 matrix. Here, each of matrix elements denotes the resulting posterior (1 to 23,331) that is updated from one in the preceding trial (denoted by the position, 1 to 23,331, on the first dimension of the matrix), given a response (either 0 or 1) to a stimulus (1 to 120). This transition matrix is computed as follows. For each of all discretized states (i.e., approximate posteriors), denoted state Θ_i, the next state Θ_j given stimulus *d* and response *y* is determined by

$$j = \arg\min_h D_{\mathrm{KL}}\left(p(\theta \mid y, d) \,\|\, p_{\Theta_h}(\theta) \right)$$

where

$$p(\theta \mid y, d) \propto p(y \mid \theta, d)\, p_{\Theta_i}(\theta).$$

Here, $p_{\Theta_h}(\theta)$ is the probability distribution corresponding to state Θ_h in the state space and $D_{\mathrm{KL}}(p \,\|\, q)$ denotes the Kullback-Leibler (KL) divergence from distribution *p* to distribution *q*. To interpret, given state Θ_i, stimulus *d* and response *y*, the next state Θ_j is determined as the one in the state space that is closest to the posterior $p(\theta \mid y, d)$ in the sense of KL divergence.

The local reward of a decision on stimulus *d* made at state Θ_i is defined by the expected information gain:

$$r(\Theta_i, d) = H(\Theta_i) - \mathbb{E}_{Y|d}[H(\Theta')]$$

where $H(p)$ is the entropy of distribution *p*, Θ' denotes the state updated from the preceding state Θ_i given response *y* to stimulus *d*, and the expected value $\mathbb{E}_{Y|d}$ is with respect to the posterior predictive distribution of *y* at state Θ_i. The values of $r(\Theta_i, d)$ for all states and stimuli are also precomputed and stored in a matrix of size 23,331 × 120.

4. Backward induction

Given the above setup, a backward induction algorithm is run as follows.

Step 0. Initialize the global rewards $R_t(\Theta_i)$ and optimal stimuli $d_t^*(\Theta_i)$ for all states Θ_i (indexed from 1 to I) and all trials t (1 to T)

Step 1. FOR the last trial T only

FOR $i = 1, 2, \ldots, I$

Compute and store $R_T(\Theta_i) = \max_d r(\Theta_i, d)$ and

$d_T^*(\Theta_i) = \arg\max_d r(\Theta_i, d)$

END

END

Step 2. FOR $t = T-1, T-2 \ldots, 1$

FOR $i = 1, 2, \ldots, I$

Compute and store $R_t(\Theta_i) = \max_d \left(r(\Theta_i, d) + \mathbb{E}_{Y|d} \left[R_{t+1}(\Theta_j \mid \Theta_i, y, d) \right] \right)$

and $d_t^*(\Theta_i) = \arg\max_d \left(r(\Theta_i, d) + \mathbb{E}_{Y|d} \left[R_{t+1}(\Theta_j \mid \Theta_i, y, d) \right] \right)$, where

$R_{t+1}(\Theta_j \mid \Theta_i, y, d)$ denotes the (previously computed) global reward in trial $t+1$ at state Θ_j which will be reached by the transition from Θ_i given response y to stimulus d.

END

END

Note that, in Step 2, the values $R_{t+1}(\Theta_j \mid \Theta_i, y, d)$ are computed efficiently by using the precomputed transition matrix to determine transitions from Θ_i to Θ_j. Also, in Steps 1 and 2, the precomputed values of $r(\Theta_i, d)$ are used.

For the order-constrained condition (see Simulation Experiments in the main text), the state space is augmented with 10 possible bins from which an optimal stimulus is selected in

each trial (i.e., Cartesian product $\Theta_i \times \{1,\ldots,10\}$). Then, in computing $R_t(\Theta_i)$ and $d_t^*(\Theta_i)$ in Step 2, whenever the bin number of the next state Θ_j is greater than that of the current state Θ_i corresponding to the candidate stimulus d, the quantity inside the max operator is set to 0 so that such *d* is excluded from maximization.

5. Forward evaluation

Given the optimal decisions computed and stored through backward induction, a session of adaptive threshold estimation over *T* trials is carried out by the forward evaluation of the stored look-up table. Suppose that stimuli in each trial are to be selected by *k*-trial-ahead optimization. Then, the algorithm for running a measurement session is as simple as follows.

Set the prior distribution $p_0(\theta)$

FOR $t = 1, 2, \ldots, T$

- Determine the optimal stimulus d_t by $d_t = d_u^*(\Theta_j)$ where $u = \max(t, T-k+1)$ and $j = \arg\min_h D_{\mathrm{KL}}\left(p_{t-1}(\theta) \,||\, p_{\Theta_h}(\theta)\right)$
- Present d_t and observe the response y_t
- Update the posterior by $p_t(\theta) \propto p(y_t \mid \theta, d_t)\, p_{t-1}(\theta)$

END

Note that, in each trial, the look-up table of optimal stimuli over all states and trials (i.e., $d_t^*(\Theta_i)$ computed from backward induction) is referred to at the position of the *k*-th to last trial unless the number of remaining trials is less than *k* (i.e., $u = \max(t, T-k+1)$) and at the state Θ_j that is closest to the current, actual posterior $p_{t-1}(\theta)$ before observing y_t in the sense of KL divergence (i.e., $j = \arg\min_h D_{\mathrm{KL}}\left(p_{t-1}(\theta) \,||\, p_{\Theta_h}(\theta)\right)$).