generating_learning_signals

October 2, 2020

1 Approach

We update the decision boundary (a) and the rate of evidence accumulation (v) using estimates of the reward difference between targets (ΔB) and the reward changepoint probability (Ω).

Nomenclature

Learning signals (estimates from ideal observer)

ΔB = signed belief in the reward difference between optimal & suboptimal targets or target identities

Ω = change point probability

Learning targets (decision parameters)

a = decision boundary

v = drift rate

Other parameters involved in calculating the above

σ_n^2 = variance of the generative distribution

σ_t^2 = estimated variance

ϕ = model confidence

H = hazard rate

r_t = reward observed

α = bayesian belief learning rate

δ = reward prediction error

RU = reward uncertainty

1.1 Calculation

The following algorithm is adapted from Vaghi et al., 2017 (code repo; main function used to calculate ideal observer parameters). A detailed description of their model is under Method Details –> Computational Model.

Note that there is an error in the calculation of model confidence in their manuscript (equation 8). There, model confidence (ϕ) is defined as the calculation for reward uncertainty (RU) instead of (1 - RU) (but their code is correct, and we use the correct calc. of model confidence).

(Note that this approximation of a Bayesian delta-rule model was first proposed by Nassar et al. 2010 and Vaghi et al. based their model on the reduced Bayesian observer in Mcguire et al. 2014).

1.1.1 Belief

Given that c = the chosen target and u = the unchosen target, the belief in the mean of the distribution of reward differences on the next trial is calculated as:

$$B_{t+1,c} = B_{t,c} + \alpha_t \delta_t$$

The unchosen target value decays to the pooled expected value of both targets, $E(r)$:

$$B_{t+1,u} = B_{t,u}(1 - \Omega_t) + \Omega_t E(r)$$

$$E(r) = \frac{\bar{r}_{t_0} + \bar{r}_{t_1}}{2}$$

The signed belief in the reward difference between targets is calculated as the difference in belief for targets 0 and 1:

$$\Delta B = B_{t,1} - B_{t,0}$$

For the purpose of visualization and analysis, this can be recast as the reward difference between optimal & suboptimal targets:

$$\Delta B_{t,opt} = B_{t,opt} - B_{t,subopt}$$

The learning rate of the model [α] is influenced by the change point probability [Ω] and the model confidence [ϕ]. The learning rate should be high if either 1) a change in the mean of the distribution of reward is likely [Ω is high] or 2) the estimate of the mean is highly imprecise [σ_n^2 is high]:

$$\alpha_t = \Omega_t + (1 - \Omega)(1 - \phi_t)$$

The prediction error, δ, is the difference between the model belief and the reward difference observed:

$$\delta_t = r_t - B_{t,c}$$

Estimated variance is calculated as:

$$\sigma_t^2 = \sigma_n^2 + \frac{(1 - \phi_t)\sigma_n^2}{\phi_t}$$

1.1.2 Changepoint probability

The changepoint probability is the likelihood that a new sample is drawn from the same Gaussian distribution centered about the current belief estimate of the model relative to the likelihood that a new sample is drawn from a uniform distribution. The changepoint probability will be close to 1 as the relative probability of a sample coming from a uniform distribution increases.

$$\Omega_t = \frac{U(r_{\Delta_t})H}{U(r_{\Delta_t})H + N(r_{\Delta_t}|B_{\Delta_t}, \sigma_t^2)(1-H)}$$

The hazard rate is the global probability that the mean of the distribution has changed (calculated as the sum of change points over the total number of trials).

$$H = \frac{sum(cp_{trials})}{n_{trials}}$$

The model confidence $[\phi]$ is a function of the changepoint probability $[\Omega]$ and the variance of the generative distribution $[\sigma_n^2]$. The first term is the variance when a changepoint is assumed to have occurred. The second term is the variance conditional on no changepoint (slowly decaying uncertainty). The third term is the rise in uncertainty when the model is unsure whether a changepoint has occurred. The same terms are in the denominator with an added variance term to reflect uncertainty arising from noise.

$$RU_t = \frac{\Omega_t\sigma_n^2 + (1-\Omega_t)(1-\phi_t)\sigma_n^2 + \Omega_t(1-\Omega_t)(\delta_t\phi_t)^2}{\Omega_t\sigma_n^2 + (1-\Omega_t)(1-\phi_t)\sigma_n^2 + \Omega_t(1-\Omega_t)(\delta_t\phi_t)^2 + \sigma_n^2}$$

$$\phi_{t+1} = 1 - RU$$

We propose that the belief in the relative reward for the two choices, B, updates the drift rate, v, or the speed of evidence accumulation:

$$v_{t+1} = \hat{\beta}_v \cdot B_{\Delta_t} + v_t$$

and that the change point probability, Ω decreases the decision threshold, a, or the amount of evidence needed to make a decision:

$$a_{t+1} = a_0 - \hat{\beta}_a \cdot \Omega_t$$

2 Computational process for generating parameters

0) Generate experimental parameters for each subject, varying the frequency of change points and the conflict between choices on a given trial.

1) The ideal observer calculation routine receives the same experimental parameters as given to the subjects. For each subject and for each session, extract the reward values and change points for each target from the experimental parameters.

- `expParam = pd.read_csv(experimental_parameters)`
- `p_targets = expParam.reward_p_t0, expParam.reward_p_t1`

2) Calculate derivatives to feed to the ideal observer calculation function.

- `reward_difference = expParam.reward_p_t0 - expParam.reward_p_t1`
- `H = expParam[(expParam.cp == 1)].shape[0] / expParam.shape[0] # hazard rate`
- `low = p_targets.min() # min reward value`
- `up = p_targets.max() # max reward value`
- `high = up - low # reward value range`

3) Update the Bayesian parameters for each trial within a session.

Inputs: Trial number (t), total number of trials (nTrials), reward values for each target (prob_reward_targets), the hazard rate (H), true variance of the generative reward distribution (sN), low (reward value min), up (reward value max), high (reward value range), choices (array of choices so far), B (belief in the value of each target), signed_B_diff (belief in target 1 - belief in target 0), B_diff (belief in chosen target - belief in unchosen target), lr (learning rate), rpe (reward prediction error), cpp (change point probability), MC (model confidence), epoch_length, sF (estimated variance of the generative reward distribution)).

Outputs: Updated B, signed_B_diff, B_diff, lr, rpe, CPP, MC, epoch_length, sF

```
B,signed_B_diff,B_diff,lr,rpe,CPP,MC,epoch_length,sF = update_bayesian_belief(t,
nTrials, prob_reward_targets, H, sN, low, up, high, choices, B, signed_B_diff,
B_diff, lr, rpe, CPP, MC, epoch_length, sF)
```

These updated outputs are then given as inputs for the next trial.

4) Repeat this procedure for all subjects and all sessions using multiprocessing.

All output for each session is saved as a pickled object under `data/simulated_data/`.

e.g. `sim789_reward6.pkl` would refer to the ideal observer parameters calculated for subject 789 for the session with reward code 6 (reward probability of 85% for the optimal choice and a switch in reward contingencies every 10 trials on average).

5) Select output contained within these pickled objects is saved for future analysis within a csv file (e.g. `sub-789_cond-8510_learning_signals.csv`).

–

6) If also simulating predicted changes in the drift-rate or the decision boundary as a function of the learning signals ΔB and Ω, then the drift-rate and decision boundary are updated according to the model and the learning rate specified in the multiprocessing script and the simulation class. Trial-by-trial boundary and drift-rate estimates are saved as part of the .pkl object for each session. Note that I have only included the specification for a single model for the sake of brevity / clarity.