Model descriptions

May 27, 2020

1 2AV + Count

- The following models including "2AV" in their model names are slight variations of the SARSA algorithm for model-free learning (Rummery and Niranjan, 1994). We changed first-action learning to include two model-free q-values representing action-value updates. The first action-value represents the prediction of which second-stage one will arrive in, each of which has its own value depending on how rewarded it has been in the recent past. The second first-stage action value represents the prediction that the first-stage action will be rewarded after the second-stage choice. Indeed, this less traditional model of the two-step task was used in Gillan et al. (2016). Separating these first-stage action values in turn removes the requirement for an eligibility trace. All models use the Bellman equation to derive model-based action values.

1.0.1 Variables

Below, t=time, s=state, a=action. At stage one, two images appeared, one of which could be selected with a given action. The image was always chosen with a certain action that did not change across the task. Each action led to two possible states (determined by a transition matrix), wherein each state had two unique images. At this second stage, an image again is selected with a given action. Note only an "s" subscript is used for second-stage action values, since one could either transition to state 2 or 3. The selection of this image would lead to a monetary reward determined by a latent probability that drifted across the task (see Gillan et al., 2016).

R = reward

$T = \begin{bmatrix} P(s_1, a_1, s_2) & P(s_1, a_2, s_2) \\ P(s_1, a_1, s_3) & P(s_1, a_2, s_3) \end{bmatrix}$ Transition matrix

M = one-hot vector indicating which first-stage action was previously taken.

- First stage:

$Q_{MF0_{t,a}}$ = First-stage action value predicting value at second stage.

$Q_{MF1_{t,a}}$ = First-stage action value predicting reward after second stage.

$Q_{MB_{t,a}}$ = Model-based value of action 1

- Second stage:

$Q_{MF2_{t,s,a}}$ = Second-stage action value predicting reward.

1.0.2 Free Parameters

All alpha parameters below were drawn from Beta priors that spans the 0-1 range.

α_1 = learning rate for both updates on first-stage action values.

α_2 = learning rate for for update on second action value.

All beta parameters below were drawn from Gamma priors that spans all positive real numbers.

β_{MF0} = inverse temperature for Q_{MF0} at first stage.

β_{MF1} = inverse temperature for Q_{MF1} at first stage.

β_{MB} = inverse temperature for Q_{MB} at first stage.

β_{st} = strength of perseveration at first stage. This multiplies the M vector, which the previously enacted first-stage action.

β_{MF2} = inverse temperature for Q_{MF2} at second stage.

1.0.3 Learning computations

- Updating the transition matrix:

Each trial, a transition counter is updated. For example if state1, action1 led to state 2 once, and on the next transition, the same transition occurs, the counting matrix would be updated as follows:

$T_{counting} = \begin{bmatrix} 1+1 & 0 \\ 0 & 0 \end{bmatrix}$

T can be one of two matrices at and given trial $T_1 = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$ or $T_2 = \begin{bmatrix} 0.3 & 0.7 \\ 0.7 & 0.3 \end{bmatrix}$ at any given trial.

This is determined by the $T_{counting}$ matrix. When $T_{counting}(1,1) + T_{counting}(2,2) > T_{counting}(1,2) + T_{counting}(2,1)$, then T_1 is used.

- Updating action-values

$Q_{MF0_{t+1,a}} = Q_{MF0_{t,a}} + \alpha_1(Q_{MF2_t} - Q_{MF0_{t,a}})$

$Q_{MF1_{t+1,a}} = Q_{MF1_{t,a}} + \alpha_1(R - Q_{MF1_{t,a}})$

$Q_{MF2_{t+1,s,a}} = Q_{MF2_{t,s,a}} + \alpha_2(R - Q_{MF2_{t,s,a}})$

Model-based q-values are then computed via the Bellman equation:

$Q_{MB_{t+1}} = P(s_2|s_1,a_i) \max_{a\in\{1,2\}} Q_{MF2}(s_2,a_i) + P(s_3|s_1,a_i) \max_{a\in\{1,2\}} Q_{MF2}(s_3,a_i).$

1.0.4 Decision computations

First-stage action:

$P(a,s_1) \propto e^{(\beta_{MF0}Q_{MF0} + \beta_{MF1}Q_{MF1} + \beta_{MB}Q_{MB} + \beta_{st}M)}$

Secon-stage action:

$P(a, s_2) \propto e^{(\beta_{MF2} Q_{MF2})}$

2 2AV + LR

R = reward

$T = \begin{bmatrix} P(s_1, a_1, s_2) & P(s_1, a_2, s_2) \\ P(s_1, a_1, s_3) & P(s_1, a_2, s_3) \end{bmatrix}$ transition matrix

M = one-hot vector indicating which first-stage action was previously taken.

- First stage:

$Q_{MF0_{t,a}}$ = First-stage action value predicting value at second stage.

$Q_{MF1_{t,a}}$ = First-stage action value predicting reward after second stage.

$Q_{MB_{t,a}}$ = Model-based value of action 1

- Second stage:

$Q_{MF2_{t,s,a}}$ = Second-stage action value predicting reward.

2.0.1 Free Parameters

All alpha parameters below were drawn from Beta priors that spans the 0-1 range.

α_1 = learning rate for both updates on first-stage action values.

α_2 = learning rate for for update on second action value.

γ = learning rate for state transitions

All beta parameters below were drawn from Gamma priors that spans all positive real numbers.

β_{MF0} = inverse temperature for Q_{MF0} at first stage.

β_{MF1} = inverse temperature for Q_{MF1} at first stage.

β_{MB} = inverse temperature for Q_{MB} at first stage.

β_{st} = strength of perseveration at first stage. This multiplies the M vector, which the previously enacted first-stage action.

β_{MF2} = inverse temperature for Q_{MF2} at second stage.

2.0.2 Learning computations

- Updating the transition matrix:

$T = \begin{bmatrix} P(s_1, a_1, s_2) & P(s_1, a_2, s_2) \\ P(s_1, a_1, s_3) & P(s_1, a_2, s_3) \end{bmatrix}$

Each trial, a transition estimate is updated with a learning rate, and probabilities are at that time normalized. For instance, if action 1 is taken and transition to state 2:

$P(s_1, a_1, s_2)_{t+1} = P(s_1, a_1, s_2)_t + \gamma(1 - P(s_1, a_1, s_2)_t)$

and

$P(s_1, a_1, s_3)_{t+1} = 1 - P(s_1, a_1, s_2)_{t+1}$

- Updating action-values

$Q_{MF0_{t+1,a}} = Q_{MF0_{t,a}} + \alpha_1(Q_{MF2_t} - Q_{MF0_{t,a}})$

$Q_{MF1_{t+1,a}} = Q_{MF1_{t,a}} + \alpha_1(R - Q_{MF1_{t,a}})$

$Q_{MF2_{t+1,s,a}} = Q_{MF2_{t,s,a}} + \alpha_2(R - Q_{MF2_{t,s,a}})$

Model-based q-values are then computed via the Bellman equation:

$Q_{MB_{t+1}} = P(s_2|s_1, a_i) \max_{a \in \{1,2\}} Q_{MF2}(s_2, a_i) + P(s_3|s_1, a_i) \max_{a \in \{1,2\}} Q_{MF2}(s_3, a_i).$

2.0.3 Decision computations

First-stage action:

$P(a, s_1) \propto e^{(\beta_{MF0} Q_{MF0} + \beta_{MF1} Q_{MF1} + \beta_{MB} Q_{MB} + \beta_{st} M)}$

Secon-stage action:

$P(a, s_2) \propto e^{(\beta_{MF2} Q_{MF2})}$

α_1 = learning rate for Q_{MF0}

α_2 = learning rate for Q_{MF1} and Q_{MB}

3 2AV LR + Counterfactual

Same as 2AV + LR except that transitions for actions not taken are updated as if the not-taken action led to the state than was not experienced for the taken action. This counterfactual inference is predicated on assumption (that was told to participants and experienced in practice) that the two actions cannot lead most often to the same state.

4 2AV + Dynamic LR

Same as 2AV + LR except here the γ decays to 0 on each trial by the following equation:

$\gamma_t = \frac{1}{\epsilon + \mathrm{N}_{action}}$

where ϵ determine the starting learning rate, and N_{action} is a tally of how many times a given action was taken.

5 2AV + Dynamic LR + Intercept

Same as 2AV + Dynamic LR except here the γ decays to a variable baseline LR, ω:

$\gamma_t = \omega + \frac{1-\omega}{\epsilon + \mathrm{N}_{action}}$

where ϵ determines time it will take to decay to baselin ω learning rate, and N_{action} is a tally of how many times a given action was taken.

6 2AV + Fixed LR

Same as 2AV + LR except here, γ is fixed across subjects.

7 2AV + Bayes

R = reward

$T = \begin{bmatrix} P(s_1, a_1, s_2) & P(s_1, a_2, s_2) \\ P(s_1, a_1, s_3) & P(s_1, a_2, s_3) \end{bmatrix}$ transition matrix

M = one-hot vector indicating which first-stage action was previously taken.

- First stage:

$Q_{MF0_{t,a}}$ = First-stage action value predicting value at second stage.

$Q_{MF1_{t,a}}$ = First-stage action value predicting reward after second stage.

$Q_{MB_{t,a}}$ = Model-based value of action 1

- Second stage:

$Q_{MF2_{t,s,a}}$ = Second-stage action value predicting reward.

- Transition matrices:

p_1 represents the belief that the true transition matrix is $T_1 = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$

Whereas p_2 represents the belief that the true transition matrix is $T_2 = \begin{bmatrix} 0.3 & 0.7 \\ 0.7 & 0.3 \end{bmatrix}$ at any given trial.

7.0.1 Fixed parameter

The beta prior defining evidence in favor of Transition Matrix 1 was initialized with mode=0.5

7.0.2 Free Parameters

All alpha parameters below were drawn from Beta priors that spans the 0-1 range.

α_1 = learning rate for both updates on first-stage action values.

α_2 = learning rate for for update on second action value.

κ = concentration of prior over belief in either possible transition matrix.

All beta parameters below were drawn from Gamma priors that spans all positive real numbers.

β_{MF0} = inverse temperature for Q_{MF0} at first stage.

β_{MF1} = inverse temperature for Q_{MF1} at first stage.

β_{MB} = inverse temperature for Q_{MB} at first stage.

β_{st} = strength of perseveration at first stage. This multiplies the M vector, which the previously enacted first-stage action.

β_{MF2} = inverse temperature for Q_{MF2} at second stage.

7.0.3 Learning computations

- Updating the transition matrix:

The mode (fixed) and concentration (free) of the beta distribution defining the prior belief in T1 and T2 was converted to E1 (evidence in favor of T1) and E2 (evidence in favor of T2) parameters describing the shape of the beta distribution by the following equations:

$E1 = \text{mode}(\kappa - 2) + 1.$

$E2 = (1\text{-mode})(\kappa - 2) + 1.$

The posterior of the beta prior is updated analytically:

$E1 = E1 + 1$ when common transitions predicted by T1 are experienced.

and

$E2 = E2 + 1$ when common transitions predicted by T2 are experienced.

Each time model-based action values are computed, evidence for each transition matrix is derived from the mean of the beta distribution by:

$p_1 = \frac{E1}{E1+E2}$ which represents the probability that T1 is the true transition matrix.

$p_2 = 1 - p_1.$

$Q_{MB_{t+1}} = (\text{Bellman Equation for } T_1)(p_1) + (\text{Bellman Equation for } T_2)(p_2).$

- Updating action-values

$Q_{MF0_{t+1,a}} = Q_{MF0_{t,a}} + \alpha_1(Q_{MF2_t} - Q_{MF0_{t,a}})$

$Q_{MF1_{t+1,a}} = Q_{MF1_{t,a}} + \alpha_1(R - Q_{MF1_{t,a}})$

$Q_{MF2_{t+1,s,a}} = Q_{MF2_{t,s,a}} + \alpha_2(R - Q_{MF2_{t,s,a}})$

Model-based q-values are then computed via the Bellman equation:

$Q_{MB_{t+1}} = P(s_2|s_1, a_i) \max_{a\in\{1,2\}} Q_{MF2}(s_2, a_i) + P(s_3|s_1, a_i) \max_{a\in\{1,2\}} Q_{MF2}(s_3, a_i).$

7.0.4 Decision computations

First-stage action:

$P(a, s_1) \propto e^{(\beta_{MF0}Q_{MF0} + \beta_{MF1}Q_{MF1} + \beta_{MB}Q_{MB} + \beta_{st}M)}$

Secon-stage action:

$P(a, s_2) \propto e^{(\beta_{MF2}Q_{MF2})}$

8 MB

Here, first-stage actions are only influences by model-based planning and a perseveration parameter.

R = reward

$T = \begin{bmatrix} P(s_1, a_1, s_2) & P(s_1, a_2, s_2) \\ P(s_1, a_1, s_3) & P(s_1, a_2, s_3) \end{bmatrix}$ transition matrix

M = one-hot vector indicating which first-stage action was previously taken.

- First stage:

$Q_{MF0_{t,a}}$ = First-stage action value predicting value at second stage.

$Q_{MF1_{t,a}}$ = First-stage action value predicting reward after second stage.

$Q_{MB_{t,a}}$ = Model-based value of action 1

- Second stage:

$Q_{MF2_{t,s,a}}$ = Second-stage action value predicting reward.

8.0.1 Free Parameters

All alpha parameters below were drawn from Beta priors that spans the 0-1 range.

α_1 = learning rate for both updates on first-stage action values.

α_2 = learning rate for for update on second action value.

γ = learning rate for state transitions

All beta parameters below were drawn from Gamma priors that spans all positive real numbers.

β_{MB} = inverse temperature for Q_{MB} at first stage.

β_{st} = strength of perseveration at first stage. This multiplies the M vector, which retains which action was taken most recently.

β_{MF2} = inverse temperature for Q_{MF2} at second stage.

8.0.2 Learning computations

- Updating the transition matrix:

$T = \begin{bmatrix} P(s_1, a_1, s_2) & P(s_1, a_2, s_2) \\ P(s_1, a_1, s_3) & P(s_1, a_2, s_3) \end{bmatrix}$

Each trial, a transition estimate is updated with a learning rate, and probabilities are at that time normalized. For instance, if action 1 is taken and transition to state 2:

$P(s_1, a_1, s_2)_{t+1} = P(s_1, a_1, s_2)_t + \gamma(1 - P(s_1, a_1, s_2)_t)$

and

$P(s_1, a_1, s_3)_{t+1} = 1 - P(s_1, a_1, s_2)_{t+1}$

- Updating action-values

$Q_{MF0_{t+1,a}} = Q_{MF0_{t,a}} + \alpha_1(Q_{MF2_t} - Q_{MF0_{t,a}})$

$Q_{MF1_{t+1,a}} = Q_{MF1_{t,a}} + \alpha_1(R - Q_{MF1_{t,a}})$

$Q_{MF2_{t+1,s,a}} = Q_{MF2_{t,s,a}} + \alpha_2(R - Q_{MF2_{t,s,a}})$

Model-based q-values are then computed via the Bellman equation:

$$Q_{MB_{t+1}} = P(s_2|s_1, a_i) \max_{a \in \{1,2\}} Q_{MF2}(s_2, a_i) + P(s_3|s_1, a_i) \max_{a \in \{1,2\}} Q_{MF2}(s_3, a_i).$$

8.0.3 Decision computations

First-stage action:

$P(a, s_1) \propto e^{(\beta_{MB} Q_{MB} + \beta_{st} M)}$

Secon-stage action:

$P(a, s_2) \propto e^{(\beta_{MF2} Q_{MF2})}$

9 1AV Model

R = reward

$T = \begin{bmatrix} P(s_1, a_1, s_2) & P(s_1, a_2, s_2) \\ P(s_1, a_1, s_3) & P(s_1, a_2, s_3) \end{bmatrix}$ Transition matrix

- First stage:

$Q_{MF1_{t,a}}$ = First-stage action value.

$Q_{MB_{t,a}}$ = Model-based value of action 1

M = one-hot vector indicating which first-stage action was previously taken.

- Second stage:

$Q_{MF2_{t,s,a}}$ = Second-stage action value predicting reward.

9.0.1 Free Parameters

All parameters below were drawn from Beta priors that spans the 0-1 range.

α_1 = learning rate for both updates on first-stage action values.

α_2 = learning rate for for update on second action value.

λ = eligibility trace

ω = weight on model-based control

All beta parameters below were drawn from Gamma priors that spans all positive real numbers.

β_1 = inverse temperature for Q_{MF0} at first stage.

β_2 = inverse temperature for Q_{MB} at first stage.

st = perseveration parameter

9.0.2 Learning computations

- Updating the transition matrix:

Each trial, a transition counter is updated. For example if state1, action1 led to state 2 once, and on the next transition, the same transition occurs, the counting matrix would be updated as follows:

$T_{counting} = \begin{bmatrix} 1+1 & 0 \\ 0 & 0 \end{bmatrix}$

T can be one of two matrices at and given trial $T_1 = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$ or $T_2 = \begin{bmatrix} 0.3 & 0.7 \\ 0.7 & 0.3 \end{bmatrix}$ at any given trial.

This is determined by the $T_{counting}$ matrix. When $T_{counting}(1,1) + T_{counting}(2,2) > T_{counting}(1,2) + T_{counting}(2,1)$, then T_1 is used.

- Updating action-values

$Q_{MF1_{t+1,a}} = Q_{MF1_{t,a}} + \alpha_1(Q_{MF2_t} - Q_{MF_{t,a}})$

$Q_{MF1_{t+2,a}} = Q_{MF1_{t+1,a}} + \alpha_1(R - Q_{MF1_{t+1,a}})$

$Q_{MF2_{t+1,s,a}} = Q_{MF2_{t,s,a}} + \alpha_2(R - Q_{MF2_{t,s,a}})$

Model-based q-values are then computed via the Bellman equation:

$Q_{MB_{t+1}} = P(s_2|s_1, a_i) \max_{a \in \{1,2\}} Q_{MF2}(s_2, a_i) + P(s_3|s_1, a_i) \max_{a \in \{1,2\}} Q_{MF2}(s_3, a_i).$

Q values for 1-stage actions are integrated in the following way:

$Q_{integrated} = \omega(Q_{MB}) + (1 - \omega)(Q_{MF1})$

9.0.3 Decision computations

First-stage action:

$P(a, s_1) \propto e^{\beta_1[Q_{integrated} + st(M)]}$

Secon-stage action:

$P(a, s_2) \propto e^{\beta_2 Q_{MF2}}$

10 1AV+ LR model

Same as Daw model except state transition estimates are learned in the same way as 2AV + LR.