Algorithmic specifications for the MonkeyCheck models

Maxime Maheu

May 16, 2015

1 Algorithmic modules

1.1 Time pressure

The time pressure module is simply a sigmoid function of the number of trials since the beginning of the blocks.

$$\tau_i = \frac{1}{1 + \exp(-\lambda i - \theta)} \tag{1}$$

Where i is the trial number, λ is the slope of the curve and θ the trial number at which the pressure begins to push the behavior toward a check.

1.2 Weighted accumulator

This module simply accumulates the number of correct trials since the beginning of the block. The accumulation is weighted by an exponential decay such that the participation of the very last trial is far beyond the ones of older trials.

$$\kappa_i = \frac{\sum_{i=1}^{N} y_i . \exp(\gamma.(i-1))}{\sum_{i=1}^{N} \exp(\gamma.(i-1))} \tag{2}$$

Where $y_i = \{0, 1\}$ depending on whether the main-task trial was correct or not. Note also that κ is normalized such that it only can evolve between 0 and 1. Furthermore there is no forgetting *per se* in the model. However the exponential decay here, initially implemented to inhibit post-error checking, also account for the memory since the older trials have very low weights (almost 0).

1.3 Information seeking module

Up to now, there is no effect of the previous check on deciding whether to check or not. The aim of this new module is to consider that the probability of checking is function of both the distance to the previous check $i_{C_{\text{last}}}$ and the gauge size g_{last} displayed during this previous check.

$$\phi_i = 1 - \exp((i - i_{C_{\text{last}}}).\alpha.\log(g_{\text{last}})) \tag{3}$$

Note that there is an information-sensitivity rate α reflecting the speed at which the information one can get from the last check becomes obsolete. We also assume a linear relationship between the lost of information and the probability of checking again.

2 Simulation

Each of these algorithmic module can be combined with respect to one an other.

$$\mathcal{M}_i^{(1)} = \tau_i(\theta, \lambda) \tag{4}$$
$$\mathcal{M}_i^{(2)} = \kappa_i(\gamma) \tag{5}$$
$$\mathcal{M}_i^{(3)} = \kappa_i(\gamma).\tau_i(\theta, \lambda) \tag{6}$$
$$\mathcal{M}_i^{(4)} = \tau_i(\theta, \lambda).\phi(\alpha) \tag{7}$$
$$\mathcal{M}_i^{(5)} = \kappa_i(\gamma).\phi(\alpha) \tag{8}$$
$$\mathcal{M}_i^{(6)} = \kappa_i(\gamma).\tau_i(\theta, \lambda).\phi(\alpha) \tag{9}$$

In brackets are the free parameters of each module. Each of these models are able to derive the probability of checking in each trial. To decide whether the model wants to check or not, we simply use a softmax rule as follows.

$$C_i^* = \frac{\beta.\exp(p(c)_i)}{\beta.\exp(p(c)_i) + \beta.\exp(p(\bar{c})_i)} \tag{10}$$

3 Fit

In oder to fit the models to the effective behavior, we will need 3 things:

- the vector of right and wrong answer y,
- the trials in which the monkey performed a check,
- the gauge filling rule (so that we can give the model g_{last} at any trial).

Contrary to most of the decision-making models, the fit of these algorithms cannot relie on an implicit rule (e.g. accuracy since the last reversal) driving the dependent variable we are interested in (the checking). Nevertheless, we can still manage to perform the models' fitting procedure in three separate ways as follows.

$$C_i^* \sim \frac{1}{1+\exp(\beta_0+\beta_1.p(c)_i)} \tag{11}$$

$$\arg\min \sum_{i=1}^{N}(C_i - C_i^*)^2 \tag{12}$$

$$\arg\min \sum_{i=1}^{N}(C_i - p(c)_i)^2 \tag{13}$$

Where C is the array of checks performed by the monkey and C^* the ones performed by the model.

Note also that these are *meta* models since they can only predict the checking behavior. In other words, they need to know the type I information (outcomes in the main task) to perform their predictions.

Finally, by computing a BIC for each model, one can get the best model: the one with (i) the few number of free parameter(s) and (ii) that best explains the data. Interestingly, each model represents a particular computational scenario which combined 1, 2 or 3 computational module(s).