Maximum-likelihood estimation of the combined-controller model in Gillan et al. (2016)

Joaquín Rapela

September 28, 2021

Abstract

Here I outline the implementation of a maximum-likelihood algorithm to estimate the parameters of the combined-controller model described in pages 19 and 20 of Gillan et al. (2016), and I present parameters of this model fitted to behavioral data.

1 Introduction

In Section 2 I describe the estimation method, in Section 3 I mention computer implementation details, in Section 4 I show results of the application of this estimation method to behavioral data, and I conclude in Section 5.

2 Estimation method

I estimated the parameters of the combined-controlled model described in pages 19 and 20 of Gillan et al. (2016) using maximum-likelihood (Eq. 1).

$$
\begin{aligned}
\theta_{ML} &= \arg\max_{\theta} \quad \log P(\{r_t, c_{1,t}, s_t, c_{2,t}\}_{t=1}^{T}|\theta) \\
&= \arg\max_{\theta} \sum_{t=1}^{T} \log P(r_t, c_{1,t}, s_t, c_{2,t}|\theta) \\
&= \arg\max_{\theta} \sum_{t=1}^{T} \{ \log P(r_t|s_t, c_{2,t}) + \\
&\qquad \log P(c_{2,t}|s_t, \beta^{stage2}, \alpha) + \\
&\qquad \log P(s_t|c_{1,t}) + \\
&\qquad \log P(c_{1,t}|\alpha, \beta^{MB}, \beta^{MF0}, \beta^{MF1}, \beta^{stick}) \} \\
&= \arg\max_{\theta} \sum_{t=1}^{T} \{ \log P(c_{2,t}|s_t, \beta^{stage2}, \alpha) + \\
&\qquad \log P(c_{1,t}|\alpha, \beta^{MB}, \beta^{MF0}, \beta^{MF1}, \beta^{stick}) \}
\end{aligned} \tag{1}
$$

where $\theta \in \{\alpha, \beta^{MB}, \beta^{MF0}, \beta^{MF1}, \beta^{stick}\}$. $P(c_{2,t}|s_t, \beta^{stage2}, \alpha)$ and $P(c_{1,t}|\alpha, \beta^{MB}, \beta^{MF0}, \beta^{MF1}, \beta^{stick})$ are given in page 19 of Gillan et al. (2016) as

$$
\begin{aligned}
P(c_{2,t}|s_t, \beta^{stage2}, \alpha) &= K_{2,t} \quad \exp\left(\beta^{stage2} Q_t^{stage2}(s_t, c_{2,t})\right)\\
P(c_{1,t}|\alpha, \beta^{MB}, \beta^{MF0}, \beta^{MF1}, \beta^{stick}) &= K_{1,t} \,(\, \beta^{MB} Q_t^{MB}(c_{1,t}) +\\
&\qquad \beta^{MF0} Q_t^{MF0}(c_{1,t}) + \beta^{MF1} Q_t^{MF1}(c_{1,t}) +\\
&\qquad \beta^{stick} I(c_{1,t} = c_{1,t-1})\,)
\end{aligned}
$$

3 Computer implementation details

I maximized the parameters in Eq. 1 by constrained gradient ascent using Python and the *minimize* function of the *scipy* library with method *L-BFGS-B*. Gradients were automatically computed using *PyTorch's Autograd*. All parameters were constrained to the interval $[-5.0, 5.0]$ with the exception of α that was constrained to the interval $[0.1, 1.0]$. Parameters of different subjects were estimated in parallel using a SLURM computer cluster. Python code to estimate the parameters of individual models and SLURM scripts to estimate these models in parallel are available here.

4 Estimation results

I estimated the parameters of the above model using behavioral data from 253 subjects, each performing 200 trials of the two-stage decision task described in Gillan et al. (2016). Figure 1 shows box plots and values of all estimated parameters of all subjects.

5 Final comments

The parameters estimates in Figure 1 have a large variability. This could reflect that the subjects behavioral strategies were very variable. Alternatively, it could indicate that 200 trials for each subject were not enough data to sufficiently constrain the parameters estimates. The latter problem could be ameliorated by estimating the parameters of all subjects simultaneously using a hierarchical model, as in Gillan et al. (2016).

Another possible explanation for the large variability of model estimates is that there is a bug in the estimation program. I implemented this program with little time and I did not tested its functionality thoroughly with, for example, test cases I neither compared the previous parameters estimates with those of the mixed effect model in the R library *lme4*. In addition, the code has not documentation.

The parameters of all 253 subjects were estimated in less than three minutes in our computer cluster. However, the code has not been optimized for speed or memory consumption. In particular, some for loops in the current code could be replaced by more efficient batch matrix multiplications. Also, the current code stores the complete value functions for all trials in memory, and a more efficient implementation could only store the value functions of a single trial.

I used constrained maximization to optimize the model likelihood function, in order to constrain the α parameter in the $[0, 1]$ interval, and to avoid extreme values of the β parameters. The implementation of this constrained maximization was not trivial. To save implementation time, I used PyTorch's Autograd, instead of computing gradients manually. However, PyTorch does not provide bounded optimization methods in its `torch.optim` package. So I used the `scipy.optimize` package and its bounded optimization method `L-BFGS-B`, with gradients computed by PyTorch.

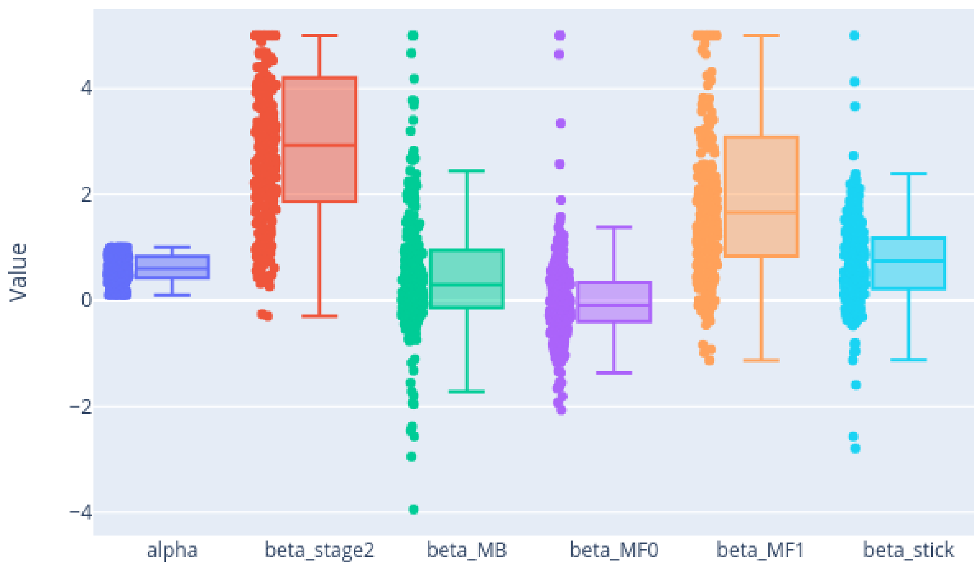

Figure 1: Box plots and values of all estimated parameters of all subjects. Click on the figure to see its interactive version.

A nice feature of the code is that it is not tied to the given file format structure. Different file formats could be used by just changing the default parameters provided in the script `doMLforDawRLmodel_scipy_LBFGSB.py`. In general, I prefer not to hard-code constants, but to use a large set of default parameters in my code. A side benefit of this choice was that I could run the estimations in the SLURM cluster by just adding two shell scripts, and without having to change a single line of the Python code.

Finally, I think that visualizations are important in statistical data analysis and I find that the interactive version of Figure 1 properly summarizes the estimation results.

References

Gillan, C. M., Kosinski, M., Whelan, R., Phelps, E. A., and Daw, N. D. (2016). Characterizing a psychiatric symptom dimension related to deficits in goal-directed control. *Elife*, 5:e11305.