Programming project for course
Bayesian Statistics and Hierarchical Bayesian Modeling for Psychological Science

[Aim]

To practice model development, model estimation, and model selection in a real-world situation.

[Tool]

R/RStudio and RStan. Download all necessary files: https://git.io/fjv9F

[Background]

Your colleague Lisa is currently working on a project using the two-armed bandit reward learning task. In this experiment, participants were instructed to choose from two abstract symbols, each of which was associated with a certain reward probability (e.g., 70% and 30%). A feedback (either win or loss) was delivered at the end of each trial (Figure 1). She just finished the data collection and is now focusing on the analysis. She is aware of a commonly used learning model in such tasks, that is, the Rescorla-Wagner Model (Rescorla & Wagner, 1972; RW), and would like to fit this model with hierarchical Bayesian methods (Gelman et al., 2013). Her supervisor told her that an increasing number of researchers are using a newly developed programming language Stan (Carpenter et al., 2017) for this purpose, and encouraged her to try it out. Lisa has known R already, but she has never used Stan. She spent a few days reading the Stan documentation and coded her first Stan model. As she expected, she received some errors, and she was unable to understand them. She was told that you are now taking a course on Bayesian statistics and cognitive modeling, and therefore, she came to you for some help with the Stan code.

A few days later, her supervisor asked her to implement a variant of the Rescorla-Wagner model with deferential learning rates, one for reward and one for punishment (RP). She did not fully understand the equation (Figure 2), so she came to you again, asking if you could help her implement this model in Stan.

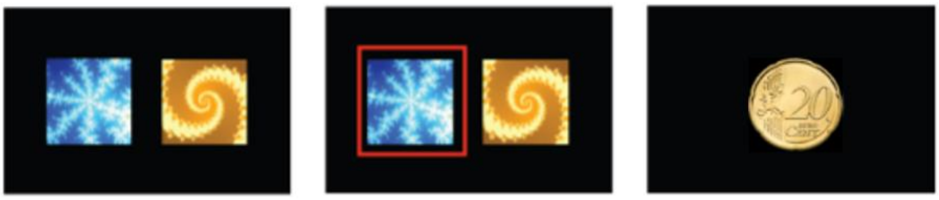

Figure1. Example trial of the decision-making task.

$$V_{c,t} = \begin{cases} V_{c,t-1} + \eta^{\text{rew}}\left(O_{t-1} - V_{c,t-1}\right), \text{ if } O_{t-1} > 0 \\ V_{c,t-1} + \eta^{\text{pun}}\left(O_{t-1} - V_{c,t-1}\right), \text{ if } O_{t-1} < 0 \end{cases}$$

Figure2. Reward-punishment model

[Task]

1. Fix the potential bugs/errors in the “rw. stan” model file, and fit this model.
2. Implement the two-learning-rate model in “rp.stan”, and fit this model.
3. Compare these two models using widely applicable information criterion (WAIC).

[Format]

Please follow the folder structure introduced during the lectures, i.e., /programming_project/{data, scripts, outputs}. Put all data in “data”, all *.R and *.stan files in “scripts”, and all stanfit objects in “outputs”. In the root folder (i.e., “programming_project”), complete the report named “short_summary.docx”, by filling in the WAIC value for each model, and deciding on the winning model.

[Submission]

Submission can be made at any time BEFORE the deadline. When submitting, make a zip file, name it as “lastname_matriculatenumber.ZIP”, and upload it to Moodle.

[Scoring]

Ten (10) points will be given for successful results submitted in due time. Six (6) points will be given for unsuccessful results submitted in due time. 0.5 point per day will be deducted for over-due submission.

References:

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., ... & Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of statistical software*, *76*(1).

Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis*. Chapman and Hall/CRC.

Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Classical Conditioning II: Current Research and Theory, 2*, 64–99.