Portfolio 1

Anita Kurm: 201608652

The beta-binomial model for inferring rates

In general terms, the **beta-binomial model** presented in Lee & Wagenmakers (2014) is a formal representation of an agent's behaviour in a task that has discrete outcomes which can occur with a certain rate . One example of such task is aim practice, where a professionally trained agent hits the target with probability of 0.9 and correspondingly misses it with probability of 0.1, whereas an amateur would hit or miss it with the equal probability of 0.5. The beta-binomial model offers us a way to infer what kind of agent we are dealing with without actually knowing it and just by observing their performance over multiple trials. More accurately, the model offers a way to infer what kind of rate of aiming ability that agent had.

Following notations by Lee & Wagenmakers (2014) , the rate inference can be achieved by representing the agent's number of hits **k** out of **n** number of trials as a Binomial probability distribution. This distribution is appropriate in this case since that is the kind of distribution our discrete outcome of the task (either hit or miss) can shape over multiple trials. Knowing that the number of hits **k** out of **n** trials depended on the aiming ability rate of the agent, we can model the rate as **a parameter of the Binomial distribution** – further referred to as **theta** θ. The notation is presented below:

$$k \sim \mathrm{Binomial}(\theta, n)$$

Even though the aiming ability rate θ is unknown and is to be inferred using the model, we can still restrict the search range for it to plausible values for a rate. A convenient way to communicate that to the model is to use a **Beta prior** for **θ**, since its range from 0 to 1 is appropriate for a rate and its parameters **alpha** and **beta** are intuitive in the context of the task at hand. More specifically, **alpha** parameter can represent count of successes and **beta** parameter can represent count of misses known before we collected the agent's performance data. If we know that before this aim practice throughout last training the agent has hit 125 targets and missed 3, we can communicate to the model that most likely in this aim practice the hit rate will be pretty close to 1 by giving it a prior of Beta(125, 3). Both alpha and beta can be given values of 1 in the case where we don't have that prior information and want all values from 0 to 1 have equal plausibility in the eyes of the model. The notation for the uninformed prior for hit rate **θ** can then be written down as follows:

$$\theta \sim \mathrm{Beta}(1, 1)$$

Graphically, Lee & Wagenmakers (2014) present this full beta-binomial model as shown if Figure 1 (left), whereas a plate notation and the used Jags code is presented in Figure 1 (right) :

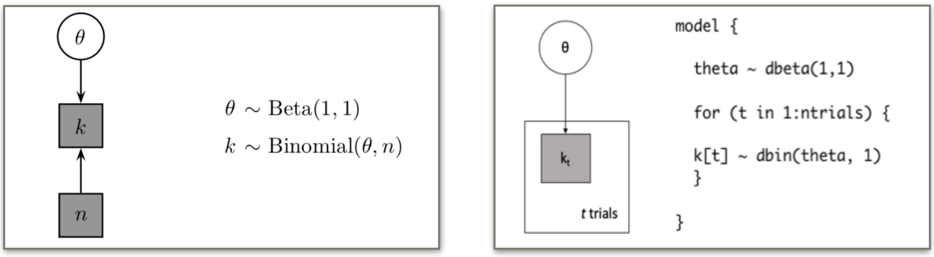

Figure 1. Graphical representations of the Beta-Binomial model and Jags code.

The learning model

In the beta-binomial model I just described, the aiming ability rate is assumed to be pre-determined and static. I.e. with every new trial of aiming at the target, the underlying aiming ability and therefore probability to hit the target stays the same. Since realistically practice has an effect on the aiming ability, a better way to model this task would be to allow the aiming ability to improve to a certain extent with every new trial. In the learning model, this property of the agent is modelled by updating theta on every trial using a certain learning rule with parameter **alpha.** The parameter alpha could be any value from 0 to 1 and represent a constant learning rate of the agent. Utilising an arbitrary learning rule for update of theta, the model presented in Figure 2 was used for simulation of a learning agent in this paper. Different agents could have different learning rates and different starting theta values (different aiming skill level prior to learning at trial 1), and the model could be used to retrieve these properties of agents. The prior for the first theta value and representation of the observable agent performance **k** remain the same as in the beta-binomial model.

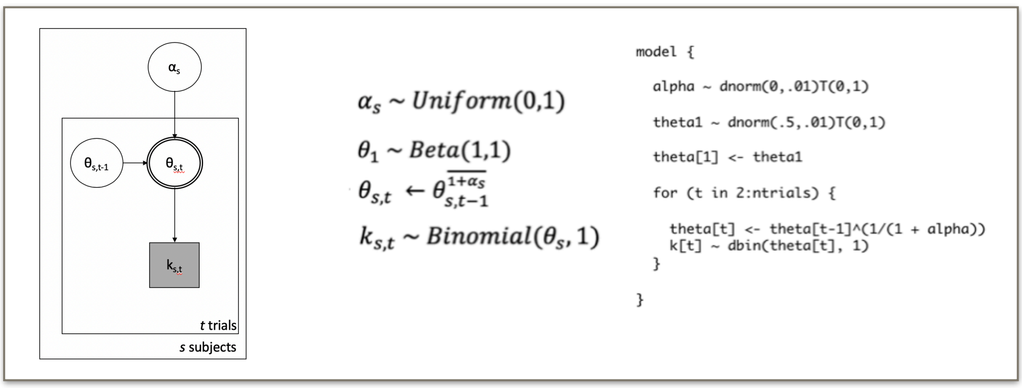

Figure 2. Graphical representation of the learning model and its Jags code.

Parameter recovery

Parameter recovery is a model evaluation procedure, where we first simulate data with known parameter values and then test whether the model is capable to infer these exact parameter values from the simulated data. If the model infers these values accurately and precisely, the posterior distribution of the inferred parameter values will be centred around the true parameter value we used to simulate the data. When that happens, we can conclude that the model works as we expected it to. On a single case basis like presented in Figure 3a, we can evaluate the Beta-Binomial model's theta parameter recovery visually as just has been described. However, for reliability it is better to repeat the parameter recovery procedure on a variety of true parameter values, and just compare the true values to maximum aposterior across different simulations like in Figure 3b.

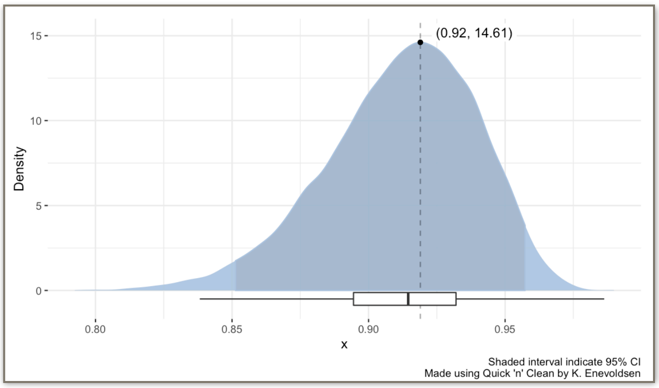

Figure 3a. Theta parameter recovery from the beta-binomial model. Data was simulated using $\theta = 0.9$, which the posterior captures pretty accurately.

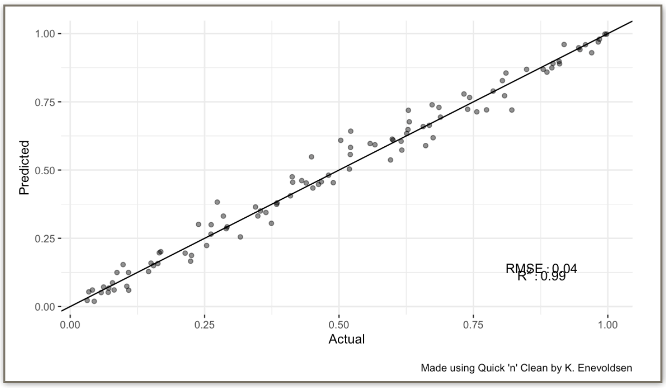

Figure 3b. Theta parameter recovery from the beta-binomial model over 100 simulations of different true values of theta. Across the whole range of θ, the maximum aposterior points corresponded closely to values of true θ

Parameter recovery seemed to work pretty well on the Beta-Binomial model. The same procedure of multiple simulation and inference was completed on the learning model, where we could recover the starting θ (further in trials θ was changing until reaching the ceiling) and the learning rate alpha. The results are presented in Figure 4.

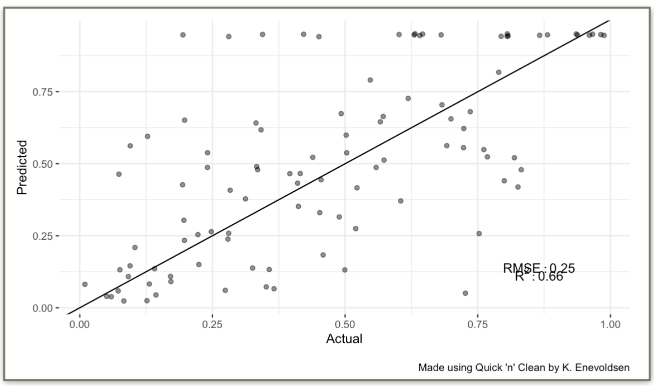

Figure 4a. Starting theta parameter recovery from the learning model over 100 simulations. Parameter recovery becomes less reliable compared to the beta-binomial model

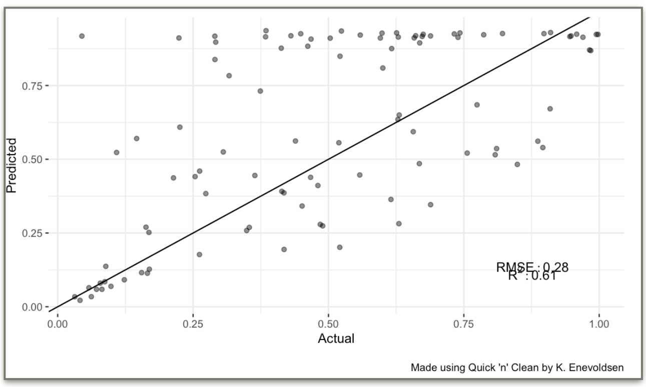

Figure 4b. Alpha parameter recovery from the learning model over 100 simulations. Parameter recovery seems to be not very reliable

Model recovery

Model recovery is another way to evaluate and compare models, where we simulate data using a certain model's parameters and principles and then compare whether the true model or its false alternative is closer to the inferred from simulated data model. The comparison is done via DIC (deviance information criterion), where the model with least DIC is considered a better fit to the simulated data. The result of multiple comparisons of different simulations can be presented in a confusion matrix, like the one in Figure 5.

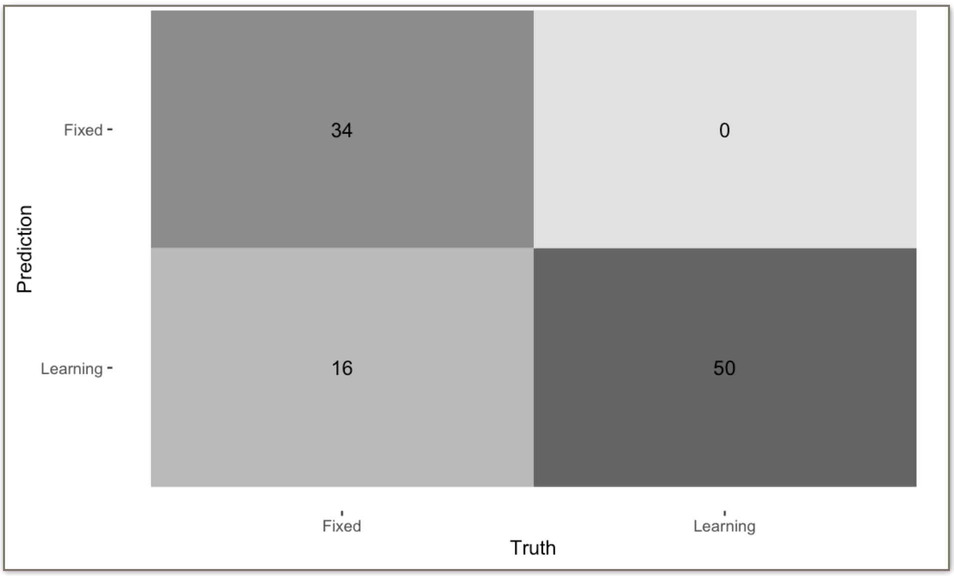

Figure 5. Confusion matrix of 50 simulations. 'Truth' represents the model that data was generated with and 'Prediction' represents which model had the smallest DIC on that data. 'Fixed' corresponds to the Beta-Binomial model

Ideally, the model is not interchangeable with other models and would almost always give the best inference given its data, like in the case of both models presented in Figure 5 (where the learning model was retrieved in all cases, and less so in the case of beta-binomial model). In general both models were very simplistic, but held up in parameter and model recovery.

References:

Lee, M. D., & Wagenmakers, E. J. (2014). Bayesian Cognitive Modeling: A Practical Course. Cambridge University Press.

Skewes, Joshua (2020). Cognitive Modelling course slides