Task Report

Fitting and Comparison of Behavioral Models

Niusha Mirhakimi[1]

Analysis of behavioral data requires a good understanding of task designing and functionalities of the human body, especially circuits or target regions of the brain that we aim to study. With the lack of a good understanding, fitting and comprising of models are blinded and just a few measurements are available to apply in choosing the top model; however, no helpful explanation could be proposed for why the model works good or even be able to detect possible errors. In the procedure of analysis, there are some important steps like, designing a model based on the task, interpretability of the model, simulations (esp. parameter recovery), model validation, and model recovery that are skipped due to the lack of information.

Models:

- **Random Model:** It is important to know whether the subject was engaged in the task or not. Because of the boringness of psychophysics tasks, there is always a chance that the participant did not focus on the task properly and she/he just randomly chose one of the available options. Bernoulli's probability model helps to find out whether it happened to a subject or not. This model has one free parameter (b) and do not use rewards' information (this is how randomness is modeled).

$$p_t^1 = b \quad and \quad p_t^2 = 1 - b$$

- **Rescorla Wagner:** In this model, the agent learns from the feedback reward and updates his belief about the chosen action. Whether using point estimation or probability distribution estimation, we can use the following equation. In the first case, the previous amount of it is used in updating the rule. And in the second case, the mean of distribution supposed as Q could be used in the equation. In the updating, the agent learns from prediction error $(r - Q_t^k)$ (which could be negative or positive). The importance of this error in updating is represented by α (learning rate) which is between 0 and 1.

$$Q_{t+1}^k = Q_t^k + \alpha(r_t - Q_t^k)$$

The probability of choosing an action comes from preference-based policy, using the soft-max function. In this equation β named inverse temperature and model how slow or fast probabilities change by the alternation of the Q-values. Based on literature, range 0 to 5 is a good range to find the optimum β. This model has two free parameters (α, β) .

$$p_t^k = \frac{\exp(\beta Q_t^k)}{\sum_{i=1}^{K} \exp(\beta Q_t^i)}$$

[1] Niusha Mirhakimi, University of Tehran, Tehran, Iran

- **Noisy Win-Stay-Lose-Shift:** WSLS is one of the models used in binary outcome tasks. During this strategy, participants stay by picking the same options on the next trial if they were rewarded and switch by picking another option if they were not rewarded. [2]

$$p(a_t = a_{t-1} | r(a_{t-1}) is\ a\ win) = p(stay|win)$$
$$p(shift|win) = 1 - \ p(stay|win) = p(a_t \neq a_{t-1} | r(a_{t-1}) i\ s\ a\ win)$$

$$p(shift|lose) = p(a_t \neq a_{t-1} | r(a_{t-1})\ is\ a\ lose)$$
$$p(stay|lose) = 1 - \ p(shift|lose) = p(a_t = a_{t-1} | r(a_{t-1})\ is\ a\ lose)$$

This is how WSLS is modeled. This model has two free parameters p_{sw}, p_{sl} which are assumed to be constant in the whole task. Here, for the sake of simplicity, one free parameter is assumed [1].

$$p^{k_t} = 1 - \frac{\epsilon}{2}\ if (a_{t-1} = k\ and\ r_{t-1} = 1)\ OR\ (a_{t-1} \neq k\ and\ r_{t-1} = -1)$$
$$p^{k_t} = \frac{\epsilon}{2}\ if (a_t - 1 \neq k\ and\ r_{t-1} = 1)\ OR\ (a_{t-1} = k\ and\ r_{t-1} = -1)$$

- **Rescorla Wagner + choice kernel**: This model is an extended version of Rescorla Wagner model . The model tries to capture the tendency for people to repeat their previous actions [1]. For modeling this phenomenon, assumed subjects specify kernel choice for each option.

$$CK_{t+1}^k = CK_t^k + \alpha_c (a_t^k - CK_t^k)$$
$$p_t^k = \frac{\exp\left(\beta Q_t^k + \beta_c CK_t^k\right)}{\sum_{i=1}^K \exp\left(\beta Q_t^i + \beta_c CK_t^i\right)}$$

This model has four free parameters. Although in general, simple models are much appreciated - because of the ease to track and the knowledge of their behavior and better generalization- sometimes complex models describe a task better and cause tremendous improvement.

- **Epsilon Greedy:** This model gives the a higher chance to last optimum action to occur, and gives other actions a chance to get explored. Fully Greedy policy does not give other bandits to explore and chooses bandit with the highest average reward, for resolve this dogmatism, one parameter, named epsilon is assumed which inserts some randomness to the model.

$$p^{k_t} = p(a_t = a_t^*) = 1 - e + \frac{e}{number\ of\ actions}$$

$$p^{k_t} = p(a_t \neq a_t^*) = \frac{e}{number\ of\ actions}$$

Measurments:

These two measurements calculate negative log-Likelihood, with one term extra penalty to avoid a selection of more complex models (which can easily overfit the data). Smaller AIC and BIC are more plausible, and the winner model has a minimum amount of AIC and BIC.

1. **AIC** :

$$AIC = -2log\acute{L}L + 2k_m$$

2. **BIC** :

$$BIC = -2log\acute{L}L + k_m \log(T)$$

Here, T shows number of run and k_m is numbere of free parameters of the model.

Code Implementation:

All scripts are implemented in Matlab. The fitting process is an optimization problem. To find the best-fitted parameters, we seek for ones who maximize the likelihood. Due to the importance of running time and computation complexity, calculating log-likelihood is more desirable. First, one function is implemented to compute the log-likelihood and then tried to minimize negative log-likelihood to find the biggest log-likelihood. Finding a global minimum is hardly possible, almost all algorithms reach a local minimum, to find global minima the algorithm should for different random initial points and if no changes were found in the minimum for continuous iteration, then we approximately reached the global minimum. This approach has been applied to all models.

- **AIC and BIC Analysis:**

First I made a matrix of log-likelihood , which embrace all subjects. Then I passed it to *aicbic* function in Matlab and then sort AIC and BIC outputs.

```
logL = [minlog_random, minlog_win_lose, minlog_td, minlog_eg, minlog_td_ex]
```

```
aic_ =

    1.0000    0.0500
    3.0000    0.1500
    4.0000    0.2000
    5.0000    0.2500

>> bic_

bic_ =

    1.0000    0.0500
    3.0000    0.1500
    4.0000    0.2000
    5.0000    0.2500
```

The first row of each matrix represents all models that were best among some subjects. The next row shows the percantage of subjects fitted well to the front model. Based on these 2 matrices, Rescorla Wagner, Epsilon Greedy, and Rescorla Wagner + choice kernel models have acceptable performance.

- **Mean and Standard Deviation Comparison:**

One other aspect of choosing the best model is the variation of parameters among the subjects. The model with a lower standard deviation is better. Here I report the mean and standard deviation of each model's free parameters.

I. Random Model:

```
>> b_mean

b_mean =

    0.3805

>> b_sd

b_sd =

    0.0632
```

II. Rescorla Wagner:

```
>> alpha_mean

alpha_mean =

    0.4002

>> alpha_sd

alpha_sd =

    0.2592
```

```
>> beta_mean

beta_mean =

    1.1270

>> beta_sd

beta_sd =

    0.9442
```

III. Noisy Win-Stay-Lose-Shift:

```
>> epsilon_mean

epsilon_mean =

    0.9843

>> epsilon_sd

epsilon_sd =

    0.0413
```

IV. Epsilon Greedy:

```
>> alpha_mean_eg

alpha_mean_eg =

    0.3912

>> alpha_sd_eg

alpha_sd_eg =

    0.1140
```

```
>> e_mean

e_mean =

    0.6848

>> e_sd

e_sd =

    0.1102
```

V. Rescorla Wagner + choice kernel:

```
>> alpha_exd_mean

alpha_exd_mean =

    0.4175

>> alpha_ex_sd

alpha_ex_sd =

    0.2814
```

```
>> beta_ex_mean

beta_ex_mean =

    1.0495

>> beta_ex_sd

beta_ex_sd =

    0.9086
```

```
>> alpha_exd_k_mean

alpha_exd_k_mean =

    0.3405

>> alpha_ex_k_sd

alpha_ex_k_sd =

    0.3221
```

```
>> beta_ex_k_mean

beta_ex_k_mean =

    0.7662

>> beta_ex_k_sd

beta_ex_k_sd =

    1.4863
```

Results:

The best log-likelihood of subjects across iteration has been depicted to ensure that the algorithm reached the global minimum. Also, the free parameters' distribution of each model has been shown.

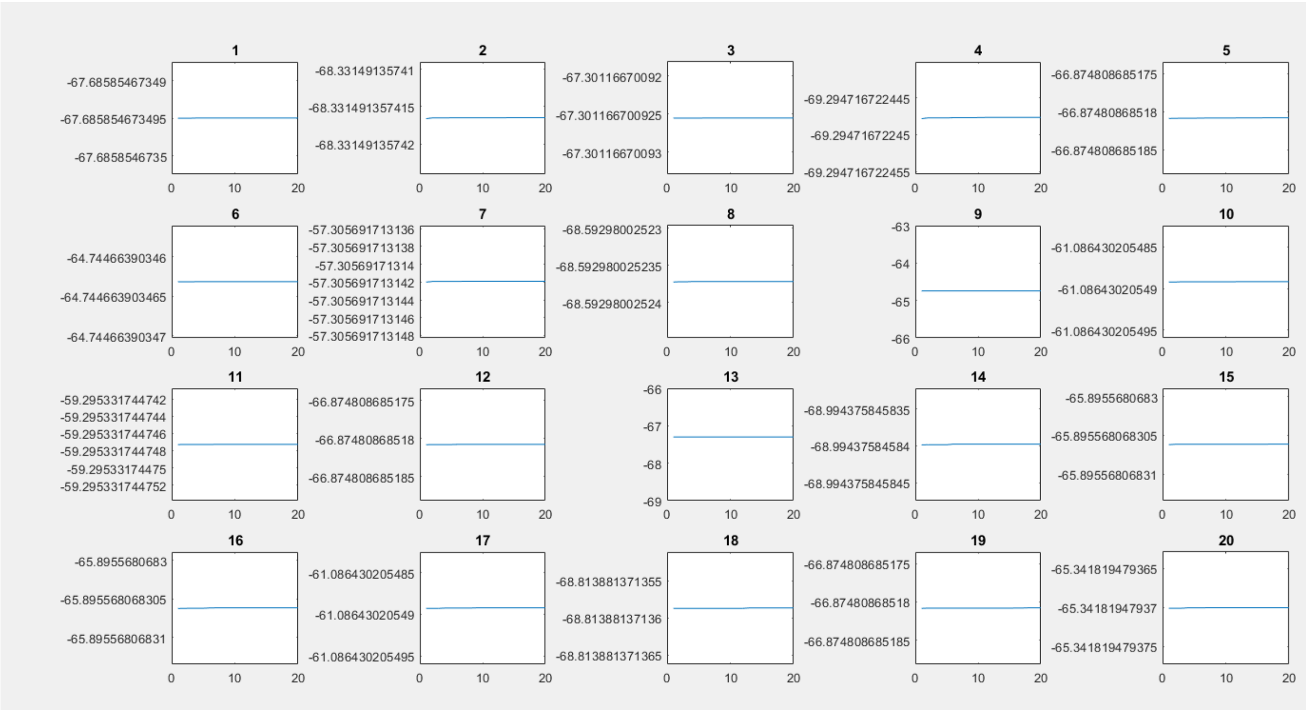

Figure 1- Random Model Log-Likelihood

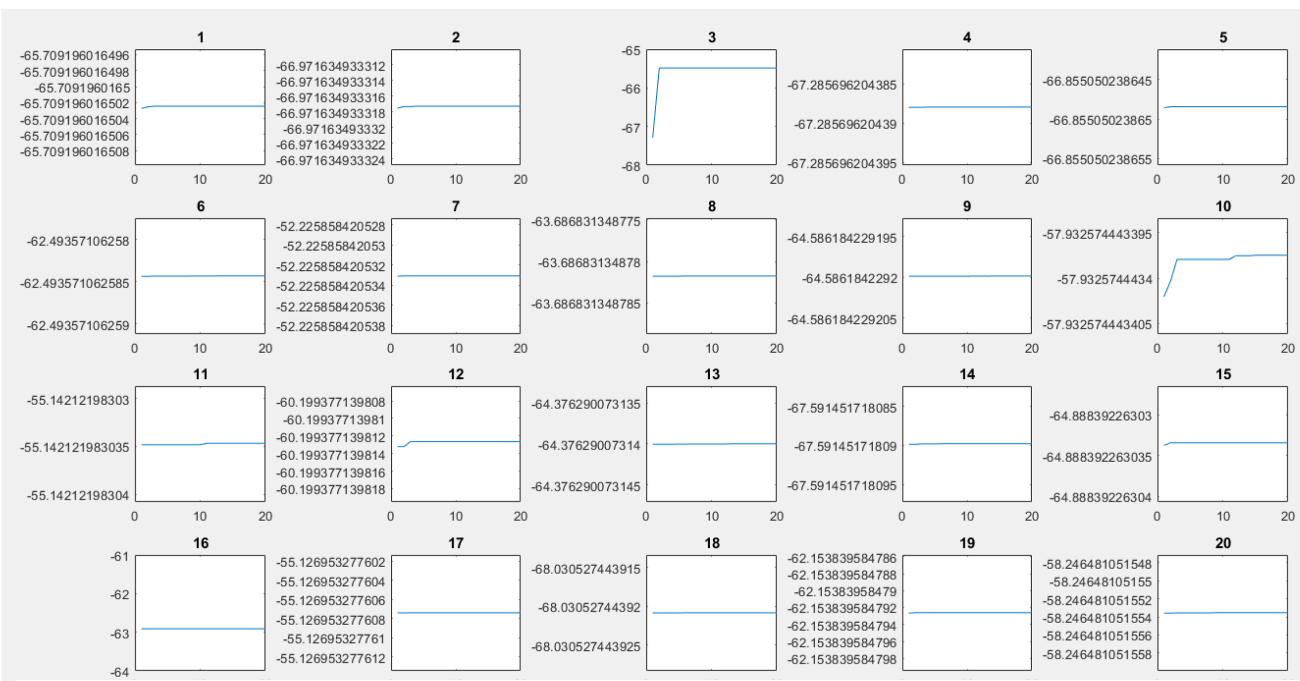

Figure 2- Rescorla Wagner Model Log-Likelihood

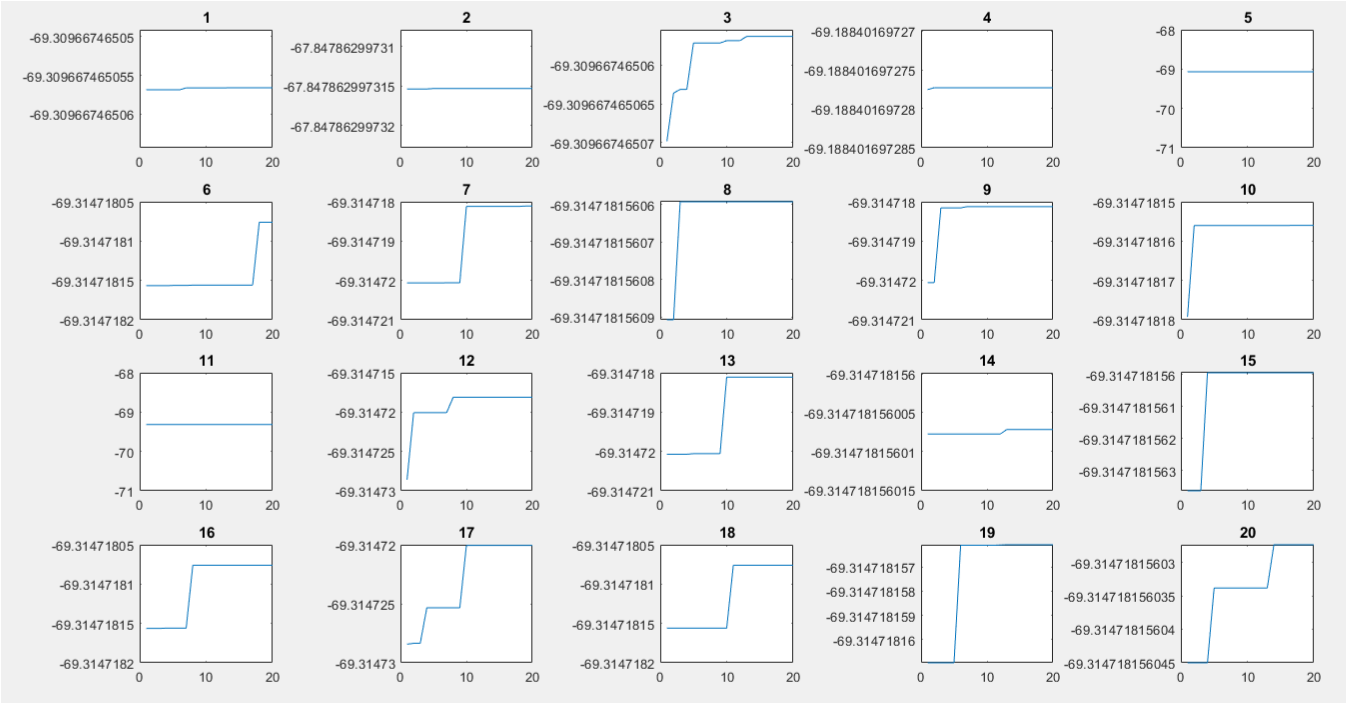

Figure 3- Noisy Win-Stay-Lose-Shift Model Log-Likelihood

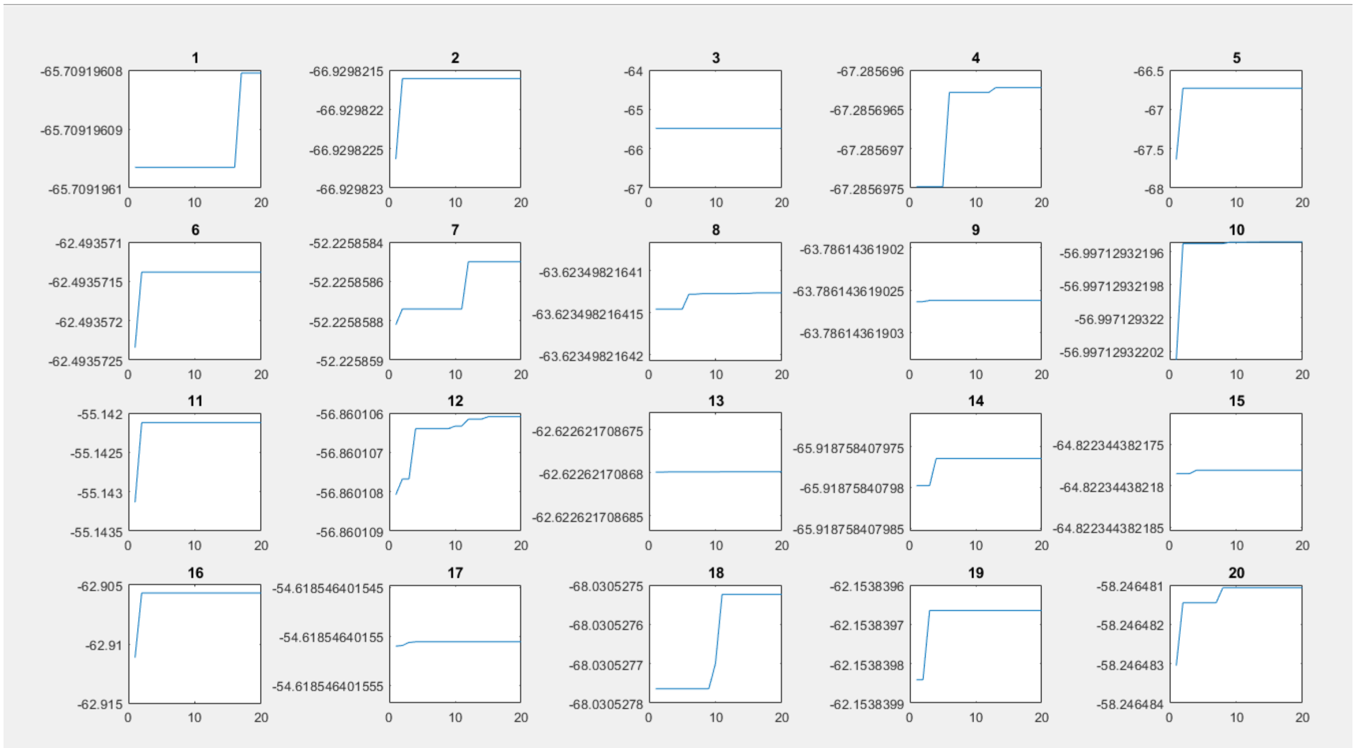

Figure 4- Rescorla Wagner+ choice kernel Model Log-Likelihood

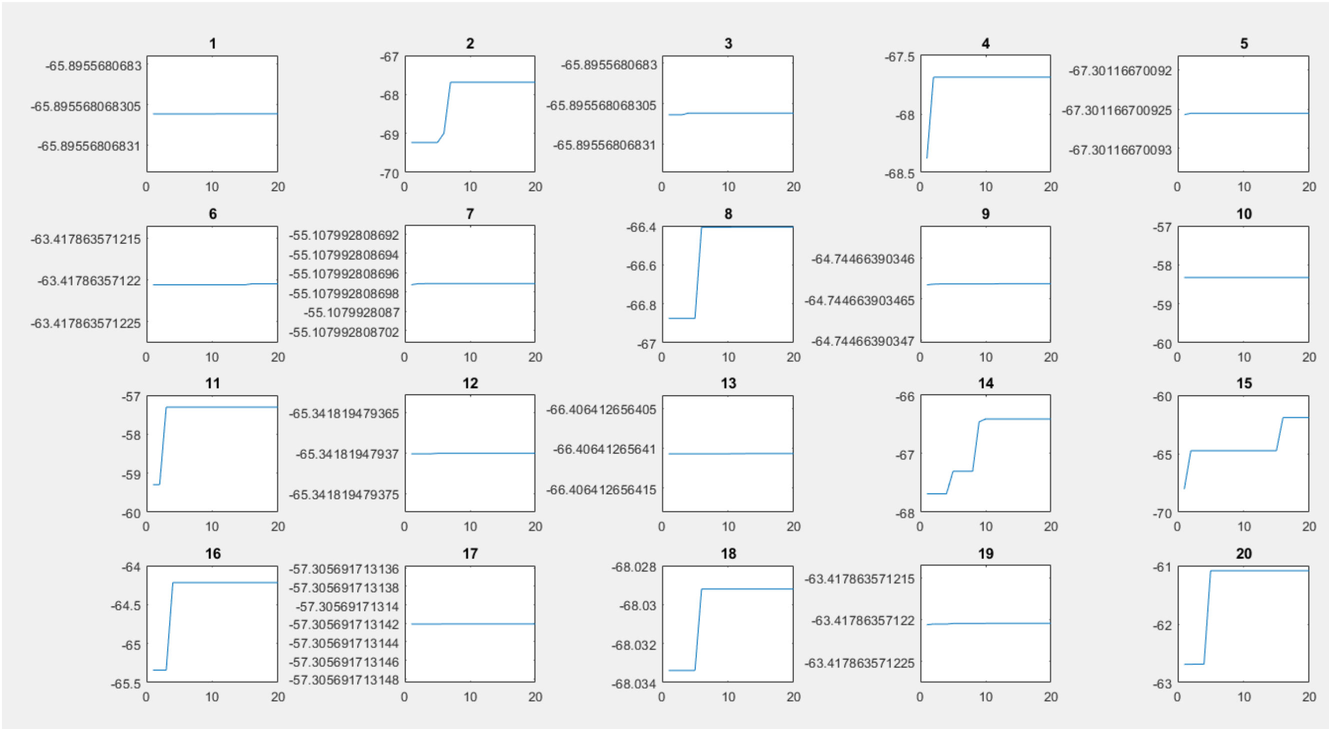

Figure 5- Epsilon Greedy Model Log-Likelihood

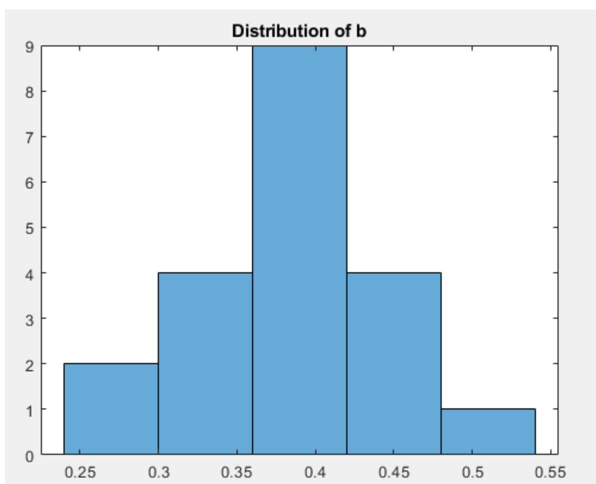

Figure 6- Random Model Parameter's Distribution

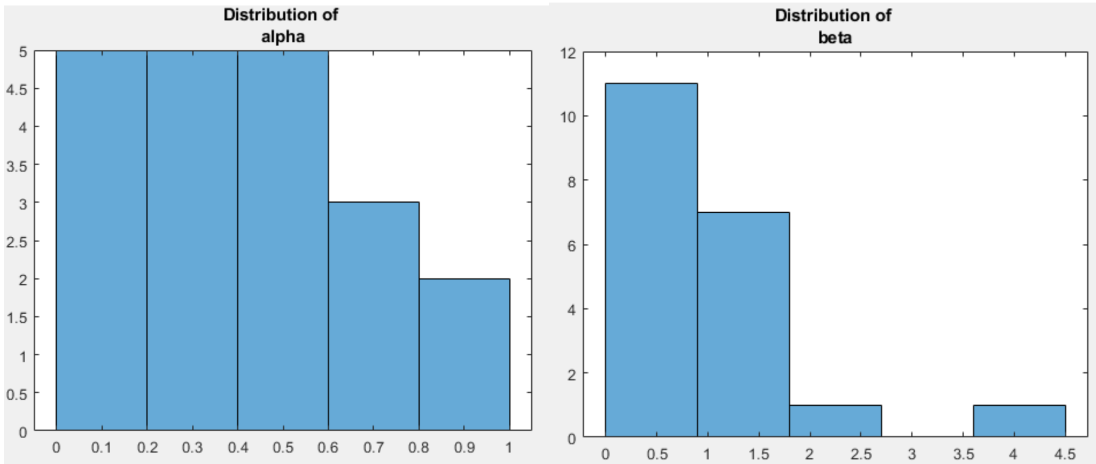

Figure 7- Rescorla Wagner Model Parameters' Distribution

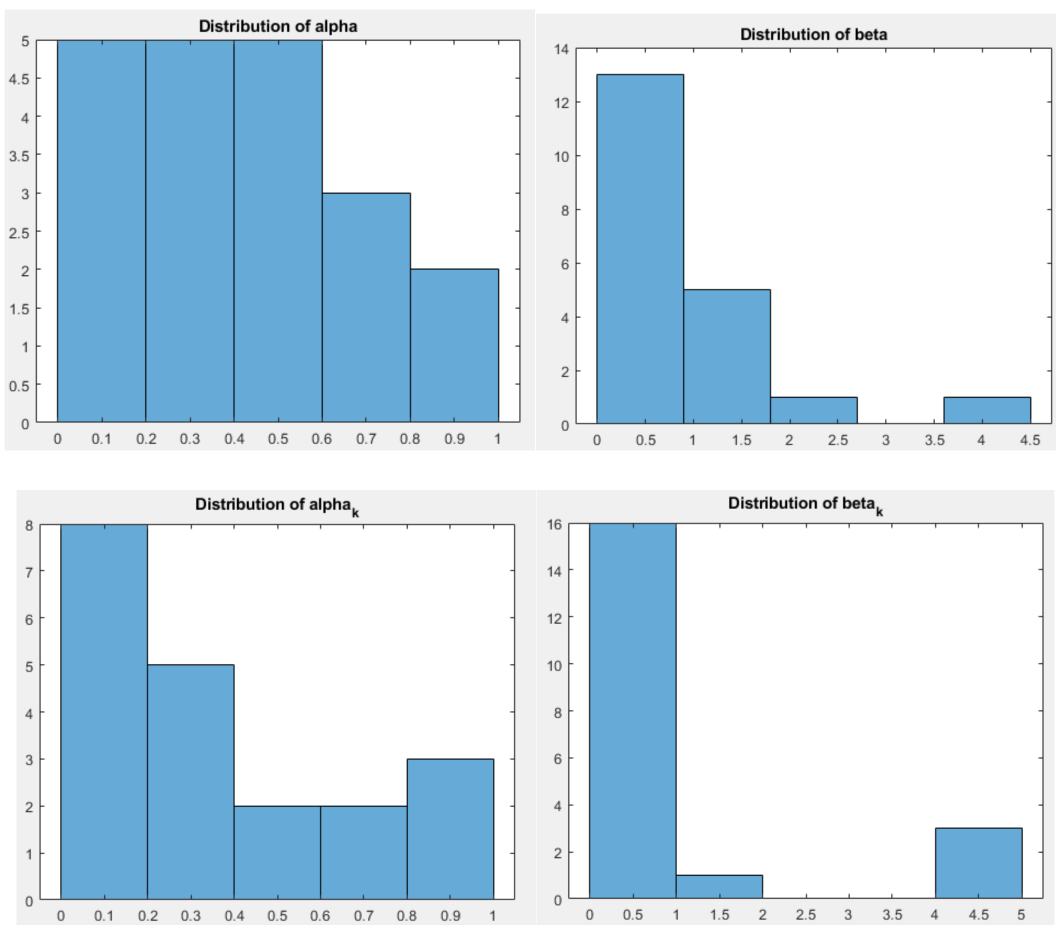

Figure 8- Rescorla Wagner + choice kernel Parameters' Distribution

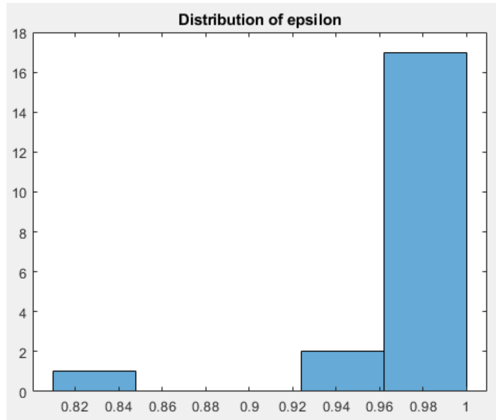

Figure 9- Noisy Win-Stay-Lose-Shift Model Parameter's Distribution

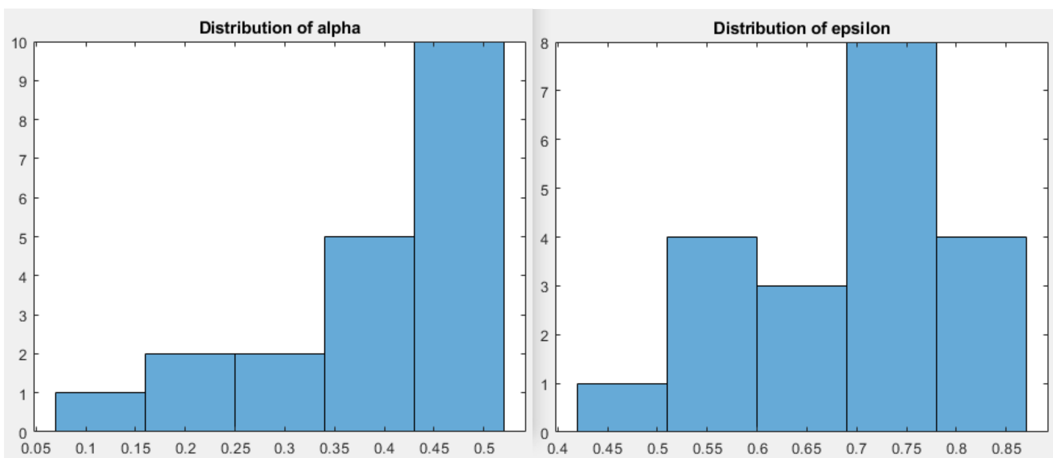

Figure 10- Epsilon Greedy Model Parameters' Distribution

Conclusion:

Epsilon Greedy Model has low variance for its parameters and because of that, it seems to be a better model in comparison to Rescorla Wagner and its extended model.

Reference:

[1] R. C. Wilson, A. GE Collins, “Ten simple rules for the computational modeling of behavioral data”, eLife, 2019.

[2] D. A. Worthy, W. T. Maddox, “A comparison model of reinforcement-learning and win stay-lose-shift decision-making processes: A tribute to W.K. Estes”, Journal of mathematical psychology”, 2014.