PAT data analysis

Magda Dubois

11/04/2022

Task

4 different FBs = -5, -1, +1, +5
Participants can either click after seeing the cue: 'Hit', or not do anything 'Miss'. In both cases they then see the reward but only recieve it if Hit. Frequencies of reward are cue-specific.
Cues = High Punishment (Cue_HP), Low Punishment (Cue_LP), Low Reward (Cue_LR), High Reward (Cue_HR)
N trials = 112 (per cue = 28)
N runs = 2 (56 trials per run, participants had a short break in between)

Reward frequencies:

```
##        Cue  R_-5  R_-1   R_1   R_5
## 0  Cue_HP  14.0   6.0   4.0   4.0
## 1  Cue_HR   4.0   4.0   6.0  14.0
## 2  Cue_LP  10.0  10.0   4.0   4.0
## 3  Cue_LR   4.0   4.0  10.0  10.0
```

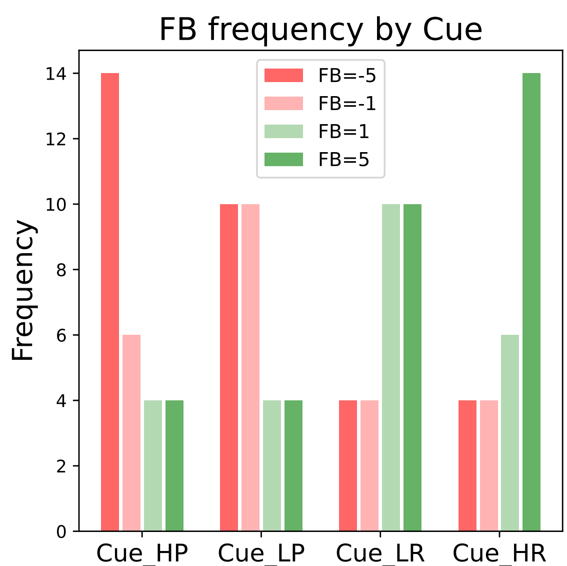

Figure 1: Task information

Behaviour

Overall N participants = 170.

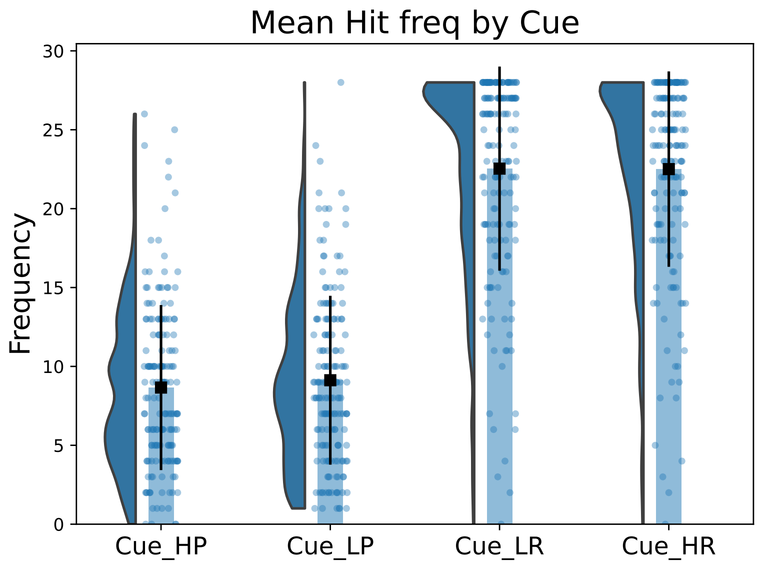

Figure 2: Summary of Behaviour

Accross time
Participant average:
Bump at the start of the 2nd run (each run is made of 56 trials)
t3 = $3 * 16 = 48$ trials
t4 = $4 * 16 = 64$ trials

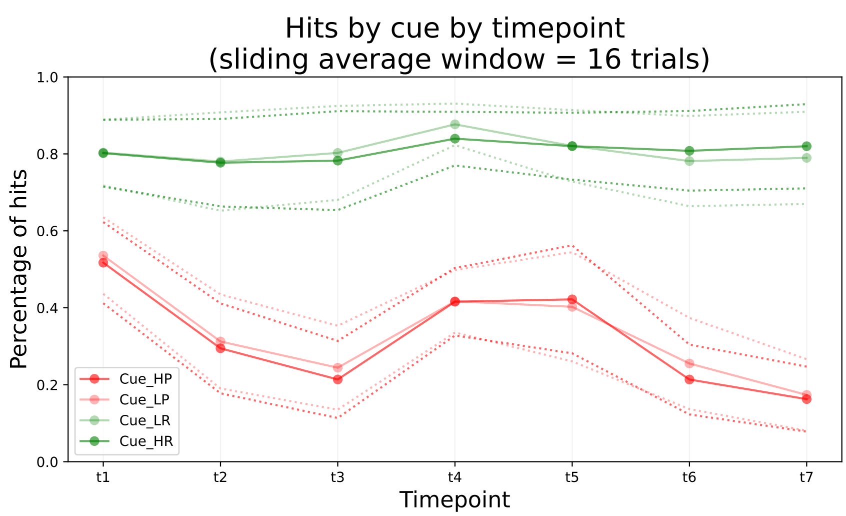

Figure 3: Behaviour accross time

Per Run
Doesn't seem to be an overall difference between runs

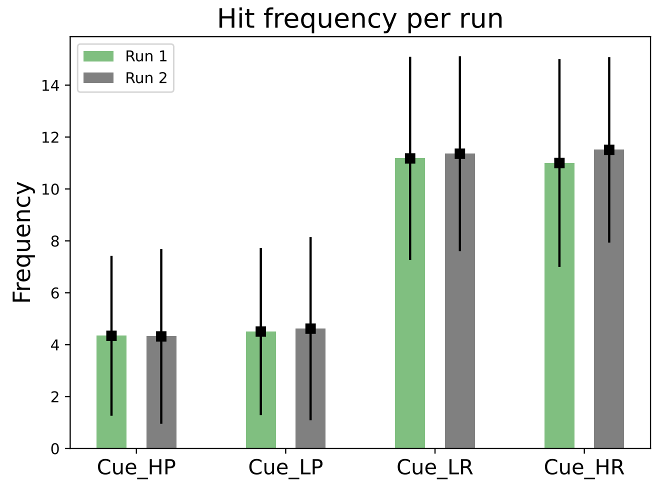

Figure 4: Behaviour by run

Per Block
Pressing bias at the beginning of each run

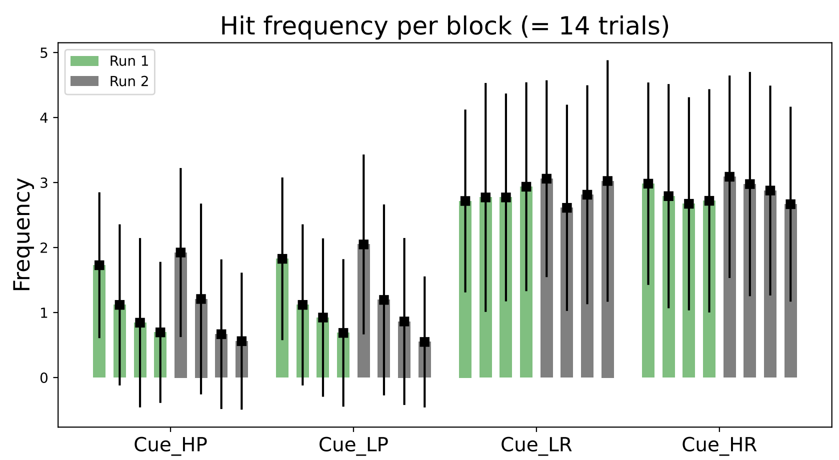

Figure 5: Behaviour by run

Simulations

Plot softmax for different betas in a likely value range [-10,10] to know how to constrict the beta parameter in the model fitting.

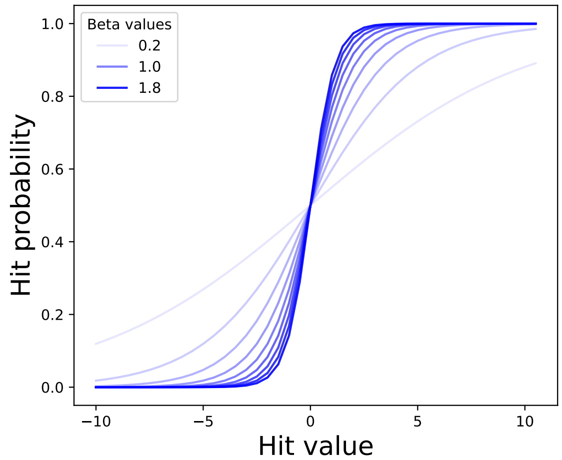

Figure 6: Softmax visualisation

Functions

Model glossary: PE = prediction error, FB = observed feedback (irrespective of hit), $V^{miss} = 0$

Value functions

Rescorla Wagner no V0

function name = rescorla_wagner_noV0
rescorla_wagner with fixed parameter: $V_0 = 0$

Rescorla Wagner

function name = rescorla_wagner
For each cue:

$$PE = FB_t - V_t^{hit}$$
$$V_{t+1}^{hit} = V_t^{hit} + \alpha \cdot PE$$

Rescorla Wagner 2 learning rates

function name = rescorla_wagner_2LR_FB
For each cue:

$$PE = FB_t - V_t^{hit}$$

FB could take the following values: -5, -1, +1, +5
Different learning rates for reward and punishment:
if $FB_t > 0$:

$$V_{t+1}^{hit} = V_t^{hit} + \alpha_{rew} \cdot PE$$

if $FB_t < 0$:

$$V_{t+1}^{hit} = V_t^{hit} + \alpha_{pun} \cdot PE$$

Rescorla Wagner weighted FB

function name = rescorla_wagner_weightRew
For each cue:
Scaling of feedback:
if abs(FB_t) = 5:

$$FB_t = w \cdot FB_t$$

Prediction error:

$$PE = FB_t - V_t^{hit}$$

$$V_{t+1}^{hit} = V_t^{hit} + \alpha \cdot PE$$

Rescorla Wagner shrinking learning rate

function name = rescorla_wagner_shrinking_alpha
For each cue:

$$PE = FB_t - V_t^{hit}$$

Shrinking factor:

$$shrink = \frac{N_{trials} - t}{N_{trials}}$$

With $N_{trials} = 112$, and $t \in [1, 112]$

$$V_{t+1}^{hit} = V_t^{hit} + \alpha_t \cdot shrink \cdot PE$$

Decision functions

Softmax

function name = my_softmax
For each cue:

$$p_t(hit) = \frac{e^{\beta \cdot V_t^{hit}}}{e^{\beta \cdot V_t^{hit}} + e^{\beta \cdot V^{miss}}} = \frac{e^{\beta \cdot V_t^{hit}}}{e^{\beta \cdot V_t^{hit}} + 1}$$

Softmax press bias

function name = my_softmax_press_bias
For each cue:

$$p_t(hit) = \frac{e^{\beta \cdot (V_t^{hit} + \pi)}}{e^{\beta \cdot (V_t^{hit} + \pi)} + e^{\beta \cdot V^{miss}}} = \frac{e^{\beta \cdot (V_t^{hit} + \pi)}}{e^{\beta \cdot (V_t^{hit} + \pi)} + 1}$$

Softmax shrinking press bias

function name = my_softmax_shrinking_press_bias
Shrinking factor:

$$shrink = \frac{N_{runtrials} - t_{run}}{N_{runtrials}}$$

With $N_{runtrials} = 56$, and $t_{run} \in [1, 56]$

For each cue:

$$p_t(hit) = \frac{e^{\beta \cdot (V_t^{hit} + \pi_t \cdot shrink)}}{e^{\beta \cdot (V_t^{hit} + \pi_t \cdot shrink)} + e^{\beta \cdot V^{miss}}} = \frac{e^{\beta \cdot (V_t^{hit} + \pi_t \cdot shrink)}}{e^{\beta \cdot (V_t^{hit} + \pi_t \cdot shrink)} + 1}$$

Models

Model 0: alpha, beta

```
mod = 0
mod_info = print_model_info(mod)
```

```
## Model = model0
## Value function = rescorla_wagner_noV0
## Decision function = my_softmax
## Parameters = alpha, beta
```

Free parameters: α = learning rate, β = inverse temperature
Fixed parameter: $V_0 = 0$

Parameter fits

```
model_folder, data_mod, data_mod_num = print_model_stats(mod)
```

```
##               nLL  Ntrials  Nparams         alpha          beta
## count  170.000000    170.0    170.0  1.700000e+02  1.700000e+02
## mean    58.107500    112.0      2.0  6.582904e-02  6.519460e+00
## std     15.140554      0.0      0.0  9.337897e-02  6.652688e+00
## min      5.050840    112.0      2.0  3.737686e-09  5.820973e-09
## 25%     48.714325    112.0      2.0  4.740678e-03  1.290706e+00
## 50%     61.159156    112.0      2.0  1.462594e-02  3.345980e+00
## 75%     70.869630    112.0      2.0  1.136493e-01  9.997270e+00
## max     77.632484    112.0      2.0  5.801689e-01  2.000000e+01
```

Plots

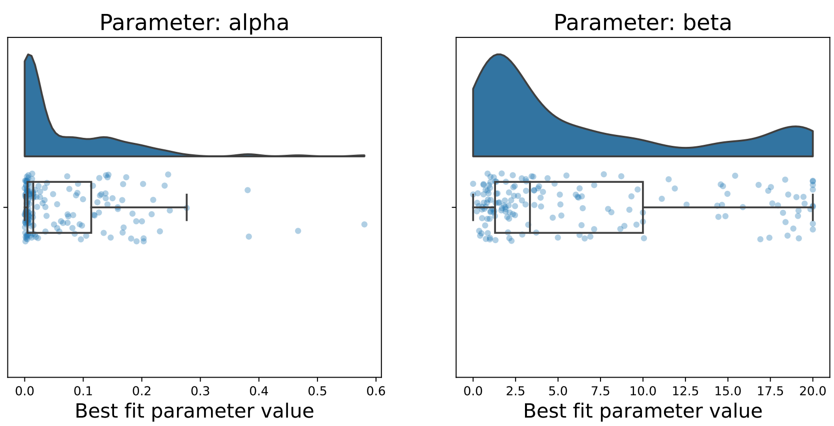

Figure 7: Model parameters

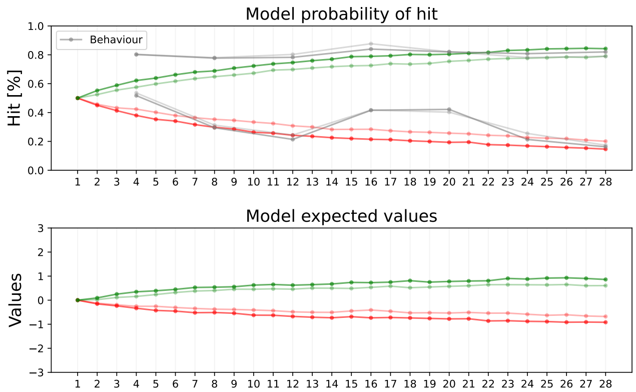

Figure 8: Model predictions

Model 1: alpha, beta, v0

```
mod = 1
mod_info = print_model_info(mod)
```

```
## Model = model1
## Value function = rescorla_wagner
## Decision function = my_softmax
## Parameters = v0, alpha, beta
```

Free parameters: α = learning rate, β = inverse temperature, V_0 = prior mean

Parameter fits

```
model_folder, data_mod, data_mod_num = print_model_stats(mod)
```

```
##               nLL  Ntrials  Nparams          v0         alpha        beta
## count  170.000000    170.0    170.0  170.000000  1.700000e+02  170.000000
## mean    52.360935    112.0      3.0    0.939961  9.182191e-02    5.850742
## std     15.228512      0.0      0.0    2.489884  1.355430e-01    6.377281
## min      4.867525    112.0      3.0  -10.000000  4.399281e-17    0.089751
## 25%     42.300949    112.0      3.0    0.025935  8.848244e-03    0.980210
## 50%     54.590413    112.0      3.0    0.198867  3.341520e-02    2.523140
## 75%     64.096722    112.0      3.0    1.003701  1.231332e-01    8.274388
## max     76.499244    112.0      3.0   10.000000  7.821228e-01   20.000000
```

Plots

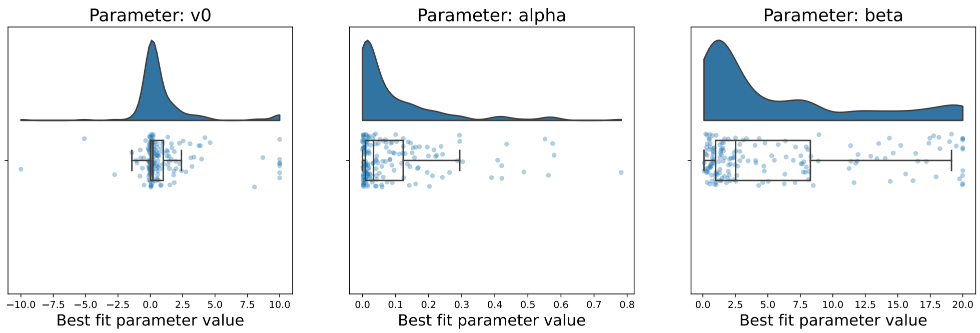

Figure 9: Model parameters

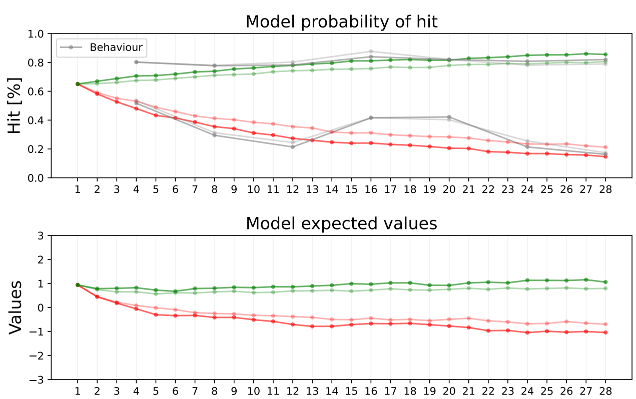

Figure 10: Model predictions

Model 2: alpha, beta, v0, pi

```
mod = 2
mod_info = print_model_info(mod)
```

```
## Model = model2
## Value function = rescorla_wagner
## Decision function = my_softmax_press_bias
## Parameters = v0, alpha, beta, pi
```

Free parameters: α = learning rate, β = inverse temperature, V_0 = prior mean, π = press bias

Parameter fits

```
model_folder, data_mod, data_mod_num = print_model_stats(mod)
```

```
##               nLL  Ntrials  Nparams
## count  170.000000    170.0    170.0
## mean    50.262414    112.0      4.0
## std     15.314180      0.0      0.0
## min      3.240895    112.0      4.0
## 25%     39.246651    112.0      4.0
## 50%     52.002387    112.0      4.0
## 75%     61.634458    112.0      4.0
## max     76.216917    112.0      4.0
##                v0         alpha        beta          pi
## count  170.000000  1.700000e+02  170.000000  170.000000
## mean     0.990495  1.153701e-01    4.880064    0.051348
## std      2.795047  1.520297e-01    5.862168    1.212308
## min    -10.000000  2.849998e-16    0.100666   -3.717110
## 25%     -0.298361  1.232139e-02    0.910771   -0.641916
## 50%      0.340262  6.827151e-02    2.096763    0.123916
## 75%      1.436749  1.472323e-01    6.459079    0.747604
## max     10.000000  1.000000e+00   20.000000    4.374439
```

Plots

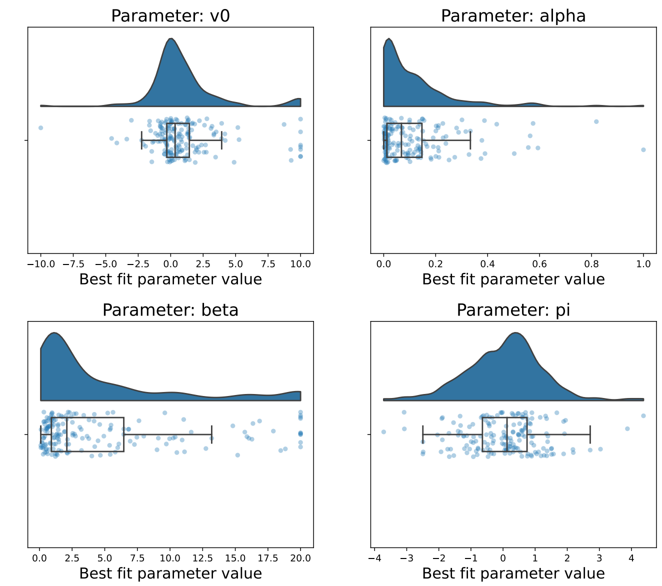

Figure 11: Model parameters

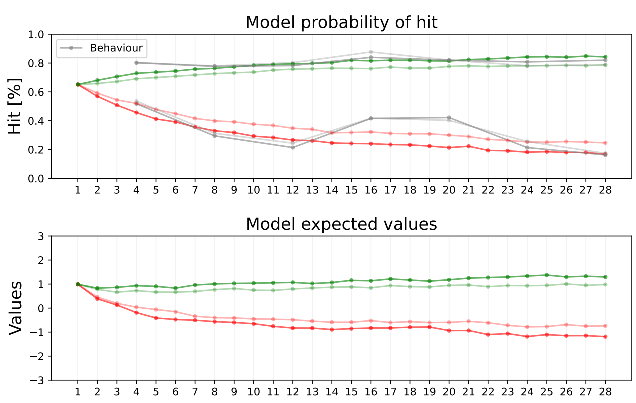

Figure 12: Model predictions

Model 3: alpha_rew, alpha_pun, beta, v0

```
mod = 3
mod_info = print_model_info(mod)
```

```
## Model = model3
## Value function = rescorla_wagner_2LR_FB
## Decision function = my_softmax
## Parameters = v0, alpha_rew, alpha_pun, beta
```

Free parameters: $\alpha_{rew,pun}$ = learning rate for reward/punishment, β = inverse temperature, V_0 = prior mean

Parameter fits

```
model_folder, data_mod, data_mod_num = print_model_stats(mod)
```

```
##               nLL  Ntrials  Nparams
## count  170.000000    170.0    170.0
## mean    50.796260    112.0      4.0
## std     15.403051      0.0      0.0
## min      2.906027    112.0      4.0
## 25%     39.812044    112.0      4.0
## 50%     52.587297    112.0      4.0
## 75%     63.267533    112.0      4.0
## max     76.348831    112.0      4.0
##                v0     alpha_rew     alpha_pun        beta
## count  170.000000  1.700000e+02  1.700000e+02  170.000000
## mean     1.191445  1.392805e-01  1.336050e-01    2.581129
## std      2.746771  1.560690e-01  1.659156e-01    3.604441
## min    -10.000000  8.300815e-21  5.344292e-17    0.093321
## 25%      0.072541  4.538379e-02  3.665528e-02    0.648320
## 50%      0.383772  8.588038e-02  7.920729e-02    1.442147
## 75%      1.284494  1.851164e-01  1.452102e-01    2.820631
## max     10.000000  1.000000e+00  1.000000e+00   20.000000
```

Stats on parameters

```
# Paired samples t-test
stats.ttest_rel(data_mod_num['alpha_rew'], data_mod_num['alpha_pun'])
```

```
## Ttest_relResult(statistic=0.4463957230934856, pvalue=0.6558828708733997)
```

Plots

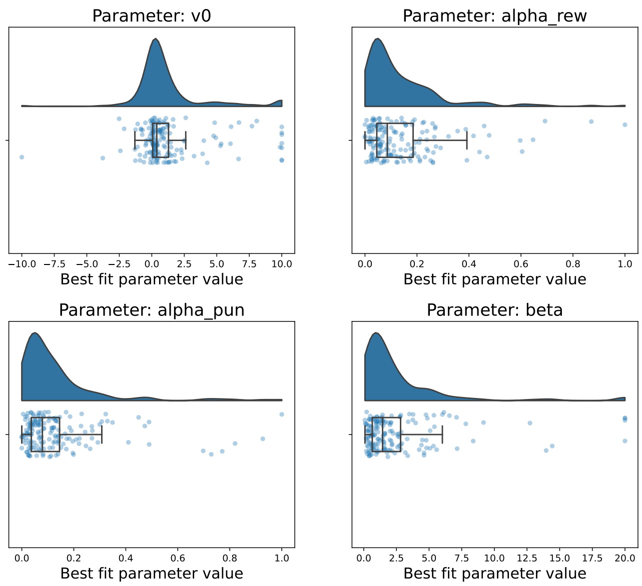

Figure 13: Model parameters

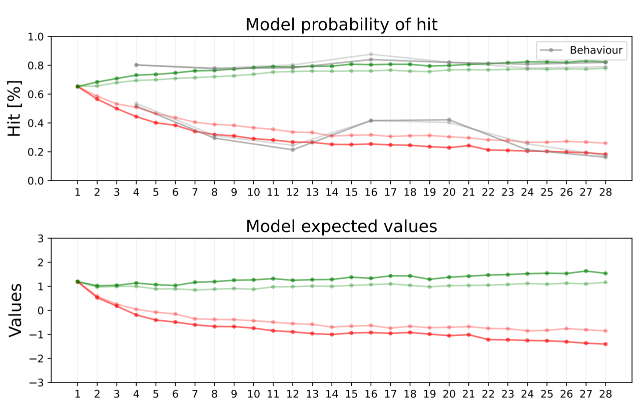

Figure 14: Model predictions

Model 4: alpha, beta, v0, w

```
mod = 4
mod_info = print_model_info(mod)
```

```
## Model = model4
## Value function = rescorla_wagner_weightRew
## Decision function = my_softmax
## Parameters = v0, alpha, beta, w
```

Free parameters: α = learning rate, β = inverse temperature, V_0 = prior mean, π = press bias

Parameter fits

```
model_folder, data_mod, data_mod_num = print_model_stats(mod)
```

```
##               nLL  Ntrials  Nparams
## count  170.000000    170.0    170.0
## mean    52.822755    112.0      4.0
## std     14.883087      0.0      0.0
## min      5.879380    112.0      4.0
## 25%     42.537941    112.0      4.0
## 50%     54.503453    112.0      4.0
## 75%     64.818023    112.0      4.0
## max     76.690227    112.0      4.0
##                v0         alpha        beta           w
## count  170.000000  1.700000e+02  170.000000  170.000000
## mean     0.978360  9.240670e-02    5.932748    1.999998
## std      2.627040  1.413147e-01    6.605705    0.000029
## min    -10.000000  3.254914e-16    0.060053    1.999627
## 25%      0.025435  1.157620e-02    0.884595    2.000000
## 50%      0.190074  3.639225e-02    2.804845    2.000000
## 75%      0.896349  1.154552e-01    8.564268    2.000000
## max     10.000000  9.698856e-01   20.000000    2.000000
```

Plots

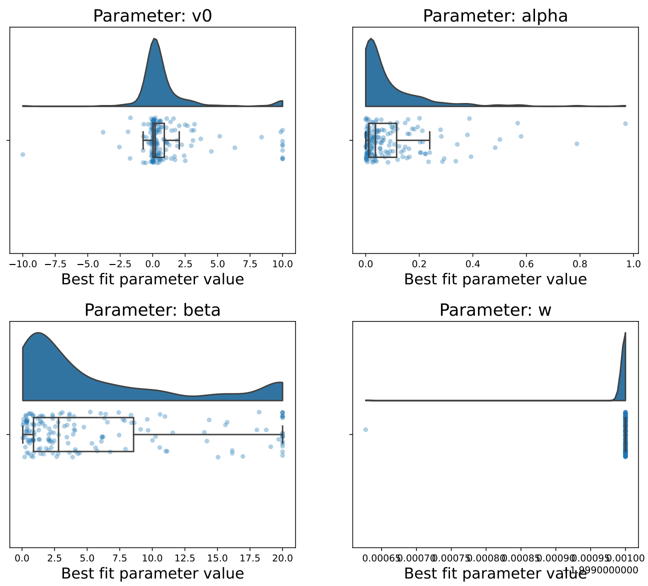

Figure 15: Model parameters

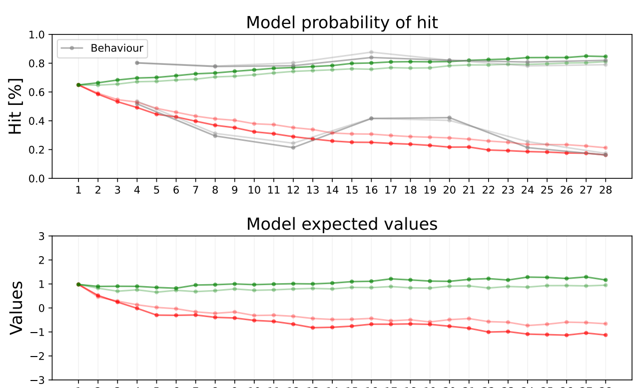

Figure 16: Model predictions

Model 5: alpha_t, beta, v0

```
mod = 5
mod_info = print_model_info(mod)
```

```
## Model = model5
## Value function = rescorla_wagner_shrinking_alpha
## Decision function = my_softmax
## Parameters = v0, alpha_t, beta
```

Free parameters: α_t = shrinking learning rate, β = inverse temperature, V_0 = prior mean

Parameter fits

```
model_folder, data_mod, data_mod_num = print_model_stats(mod)
```

```
##                 nLL  Ntrials  Nparams          v0       alpha_t        beta
## count  170.000000    170.0    170.0  170.000000  1.700000e+02  170.000000
## mean    51.755054    112.0      3.0    1.041759  1.366831e-01    5.308782
## std     15.374300      0.0      0.0    2.711148  1.721451e-01    5.995968
## min      4.647453    112.0      3.0  -10.000000  2.583236e-16    0.093505
## 25%     41.106491    112.0      3.0    0.025068  1.185567e-02    0.951547
## 50%     54.004206    112.0      3.0    0.244133  6.695768e-02    1.921968
## 75%     63.455126    112.0      3.0    1.305117  1.951048e-01    8.745747
## max     76.580572    112.0      3.0   10.000000  9.189379e-01   20.000000
```

Plots

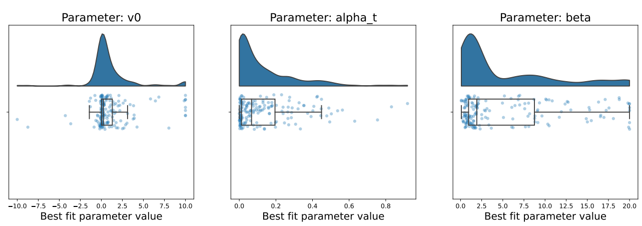

Figure 17: Model parameters

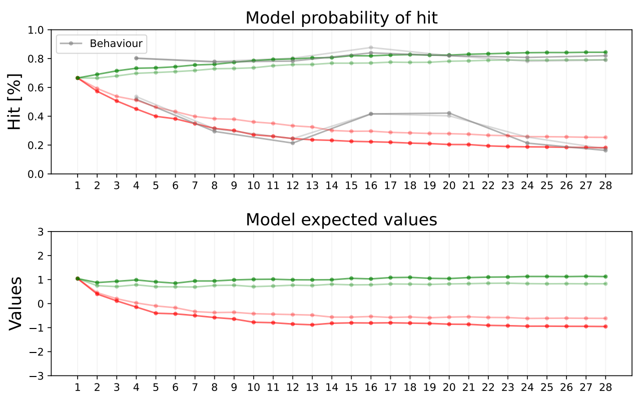

Figure 18: Model predictions

Model 6: alpha_rew, alpha_pun, beta, v0, pi_t

```
mod = 6
mod_info = print_model_info(mod)
```

```
## Model = model6
## Value function = rescorla_wagner_2LR_FB
## Decision function = my_softmax_shrinking_press_bias
## Parameters = v0, alpha_rew, alpha_pun, beta, pi_t
```

Free parameters: $\alpha_{rew,pun}$ = learning rate for reward/punishment, β = inverse temperature, V_0 = prior mean, π_t = shrinking press bias

Parameter fits

```
model_folder, data_mod, data_mod_num = print_model_stats(mod)
```

```
##              nLL  Ntrials  Nparams
## count  170.000000    170.0    170.0
## mean    46.913267    112.0      5.0
## std     16.729136      0.0      0.0
## min      3.155677    112.0      5.0
## 25%     35.609496    112.0      5.0
## 50%     48.677384    112.0      5.0
## 75%     60.739674    112.0      5.0
## max     76.499753    112.0      5.0
##               v0     alpha_rew     alpha_pun        beta        pi_t
## count  170.000000  1.700000e+02  1.700000e+02  170.000000  170.000000
## mean    -0.107151  1.098816e-01  1.311821e-01    3.791271    1.112207
## std      1.964040  1.420514e-01  1.940812e-01    4.530921    1.677841
## min     -5.000000  1.744307e-22  3.737089e-09    0.120519   -7.095337
## 25%     -1.046358  2.654594e-02  2.467444e-02    0.958264    0.233941
## 50%     -0.318710  5.443208e-02  5.757258e-02    1.912919    0.833557
## 75%      0.405973  1.398089e-01  1.516415e-01    4.395514    1.759054
## max      5.000000  1.000000e+00  1.000000e+00   15.000000    7.500008
```

Stats on parameters

```
# Paired samples t-test
stats.ttest_rel(data_mod_num['alpha_rew'], data_mod_num['alpha_pun'])
```

```
## Ttest_relResult(statistic=-1.7422330810432594, pvalue=0.08328687759338284)
```

Plots

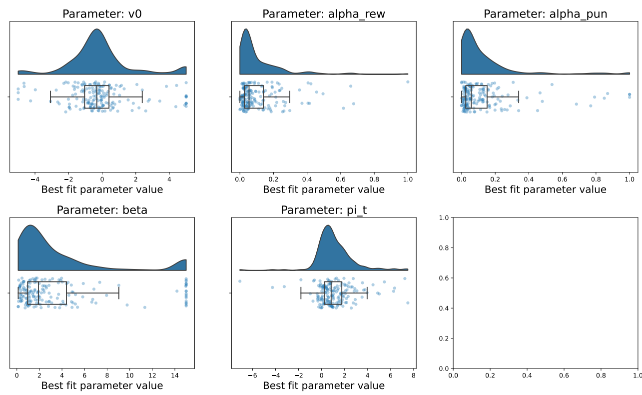

Figure 19: Model parameters

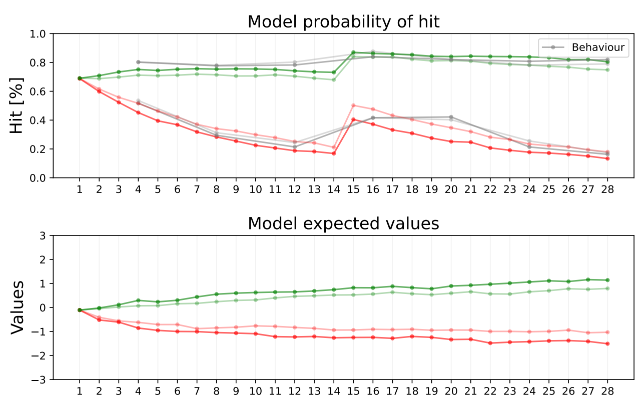

Figure 20: Model predictions

Model 7: alpha, beta, v0, w, pi__t

```
mod = 7
mod_info = print_model_info(mod)
```

```
## Model = model7
## Value function = rescorla_wagner_weightRew
## Decision function = my_softmax_shrinking_press_bias
## Parameters = v0, alpha, beta, w, pi_t
```

Free parameters: α = learning rate, β = inverse temperature, V_0 = prior mean, w = large FB weight, π_t = shrinking press bias

Parameter fits

```
model_folder, data_mod, data_mod_num = print_model_stats(mod)
```

```
##               nLL  Ntrials  Nparams
## count  170.000000    170.0    170.0
## mean    49.086785    112.0      5.0
## std     16.379621      0.0      0.0
## min      5.694755    112.0      5.0
## 25%     37.089098    112.0      5.0
## 50%     50.825920    112.0      5.0
## 75%     62.037113    112.0      5.0
## max     76.516800    112.0      5.0
##                v0         alpha        beta             w        pi_t
## count  170.000000  1.700000e+02  170.000000  1.700000e+02  170.000000
## mean    -0.076820  1.185178e-01    3.839539  2.000000e+00    1.108973
## std      2.130397  1.423396e-01    4.262047  4.173539e-10    2.071525
## min     -5.000000  4.686312e-16    0.123766  2.000000e+00   -5.600134
## 25%     -0.870512  2.929666e-02    0.788073  2.000000e+00    0.127697
## 50%     -0.262699  7.865536e-02    2.308973  2.000000e+00    0.657731
## 75%      0.379249  1.546136e-01    4.779899  2.000000e+00    1.782952
## max      5.000000  1.000000e+00   15.000000  2.000000e+00    8.659615
```

Plots

```
## <string>:11: RuntimeWarning: More than 20 figures have been opened. Figures created through the pypl
```

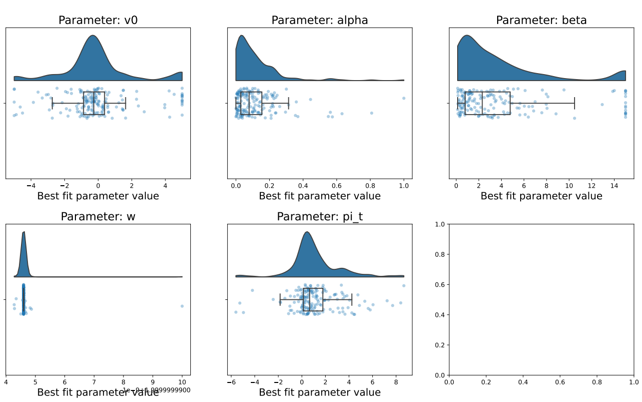

Figure 21: Model parameters

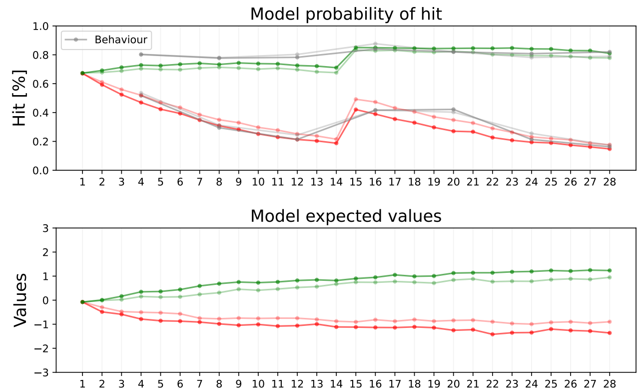

Figure 22: Model predictions

Model 8: alpha_t, beta, v0, pi_t

```
mod = 8
mod_info = print_model_info(mod)
```

```
## Model = model8
## Value function = rescorla_wagner_shrinking_alpha
## Decision function = my_softmax_shrinking_press_bias
## Parameters = v0, alpha_t, beta, pi_t
```

Free parameters: α_t = shrinking learning rate, β = inverse temperature, V_0 = prior mean, π_t = shrinking press bias

Parameter fits

```
model_folder, data_mod, data_mod_num = print_model_stats(mod)
```

```
##               nLL  Ntrials  Nparams
## count  170.000000    170.0    170.0
## mean    47.798468    112.0      4.0
## std     16.920126      0.0      0.0
## min      3.325319    112.0      4.0
## 25%     37.156746    112.0      4.0
## 50%     50.614676    112.0      4.0
## 75%     61.386058    112.0      4.0
## max     76.705921    112.0      4.0
##                v0       alpha_t        beta        pi_t
## count  170.000000  1.700000e+02  170.000000  170.000000
## mean    -0.116659  1.811305e-01    3.293070    1.062610
## std      2.272556  2.089355e-01    3.500643    1.764987
## min     -5.000000  1.264101e-16    0.122773   -6.415641
## 25%     -1.234541  3.692270e-02    0.877272    0.192999
## 50%     -0.338046  1.023679e-01    1.757615    0.948407
## 75%      0.635814  2.453132e-01    4.711434    1.927004
## max      5.000000  1.000000e+00   15.000000    6.668164
```

Plots

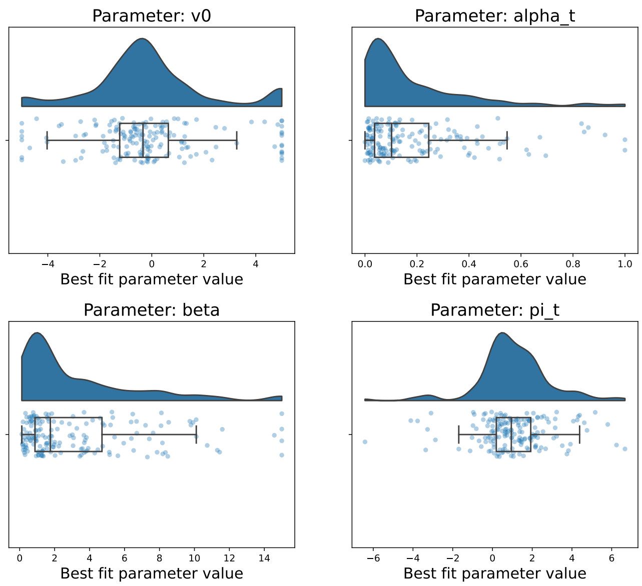

Figure 23: Model parameters

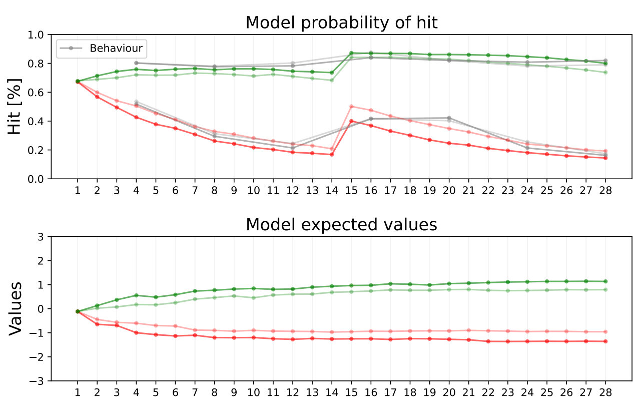

Figure 24: Model predictions

Model 9: alpha, beta, v0, pi_t

```
mod = 9
mod_info = print_model_info(mod)
```

```
## Model = model9
## Value function = rescorla_wagner
## Decision function = my_softmax_shrinking_press_bias
## Parameters = v0, alpha, beta, pi_t
```

Free parameters: α = learning rate, β = inverse temperature, V_0 = prior mean, π_t = shrinking press bias

Parameter fits

```
model_folder, data_mod, data_mod_num = print_model_stats(mod)
```

```
##               nLL  Ntrials  Nparams
## count  170.000000    170.0    170.0
## mean    48.435748    112.0      4.0
## std     16.706529      0.0      0.0
## min      3.563870    112.0      4.0
## 25%     36.917690    112.0      4.0
## 50%     50.649393    112.0      4.0
## 75%     61.727837    112.0      4.0
## max     77.041330    112.0      4.0
##                v0         alpha        beta         pi_t
## count  170.000000  1.700000e+02  170.000000  170.000000
## mean    -0.092623  1.139156e-01    3.445958    1.030806
## std      2.170045  1.427369e-01    3.950705    2.004097
## min     -5.000000  3.737686e-09    0.038199   -7.720410
## 25%     -1.136027  2.639210e-02    0.818287    0.100299
## 50%     -0.309904  6.618056e-02    1.717946    0.984759
## 75%      0.472413  1.488237e-01    4.586511    1.948065
## max      5.000000  1.000000e+00   15.000000    8.429271
```

Plots

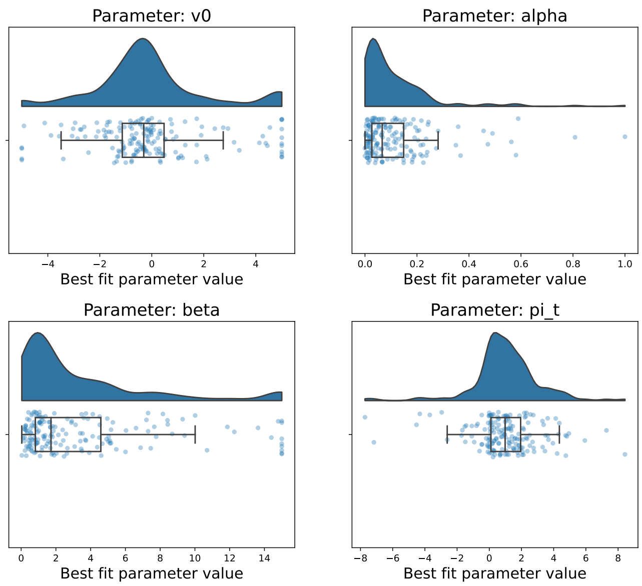

Figure 25: Model parameters

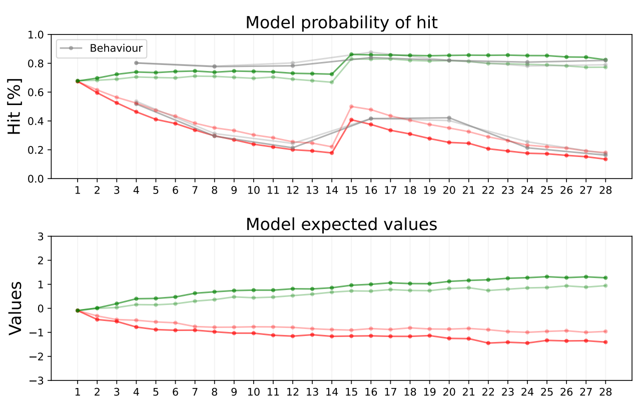

Figure 26: Model predictions

Model comparison

Formulas

$$BIC = 2 \cdot nLL + Nparams \cdot ln(Ntrials)$$
$$AIC = 2 \cdot nLL + 2 \cdot Nparams$$

With: nLL = negative log likelihood

Results

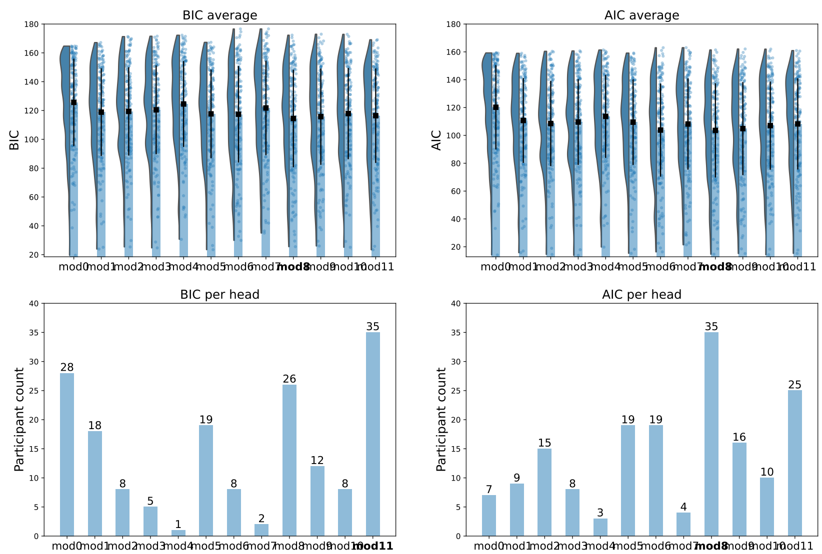

Figure 27: Model comparison

Compare mod8 and mod9

```
##          nLL_mod8     v0_mod8    alpha_mod8    beta_mod8    pi_t_mod8
## count  170.000000  170.000000  1.700000e+02  170.000000  170.000000
## mean    47.798468   -0.116659  1.811305e-01    3.293070    1.062610
## std     16.920126    2.272556  2.089355e-01    3.500643    1.764987
## min      3.325319   -5.000000  1.264101e-16    0.122773   -6.415641
## 25%     37.156746   -1.234541  3.692270e-02    0.877272    0.192999
## 50%     50.614676   -0.338046  1.023679e-01    1.757615    0.948407
## 75%     61.386058    0.635814  2.453132e-01    4.711434    1.927004
## max     76.705921    5.000000  1.000000e+00   15.000000    6.668164
```

```
##          nLL_mod9     v0_mod9    alpha_mod9   beta_mod9    pi_t_mod9
## count  170.000000  170.000000  1.700000e+02  170.000000  170.000000
## mean    48.435748   -0.092623  1.139156e-01    3.445958    1.030806
## std     16.706529    2.170045  1.427369e-01    3.950705    2.004097
## min      3.563870   -5.000000  3.737686e-09    0.038199   -7.720410
## 25%     36.917690   -1.136027  2.639210e-02    0.818287    0.100299
## 50%     50.649393   -0.309904  6.618056e-02    1.717946    0.984759
## 75%     61.727837    0.472413  1.488237e-01    4.586511    1.948065
## max     77.041330    5.000000  1.000000e+00   15.000000    8.429271
```

Plots

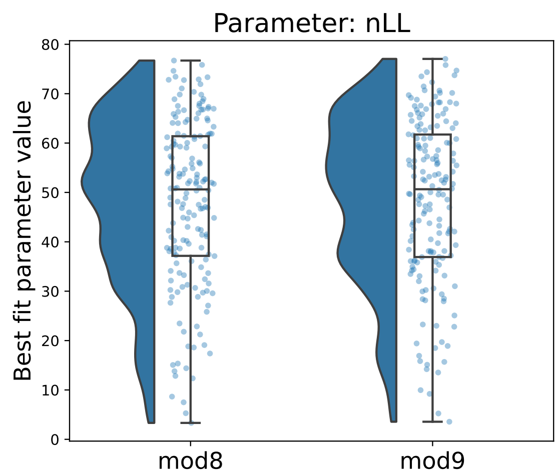

Figure 28: Likelihoods

Stats: t-tests

$$V_0 :$$

```
## mod8:  Ttest_1sampResult(statistic=-0.6693136632195669, pvalue=0.504208757753263)
```

```
## mod9:  Ttest_1sampResult(statistic=-0.5565136129805143, pvalue=0.5785959469927398)
```

Stats: paired t-tests

```
## nLL:
## means:
## mod8:  47.79846829164395
## mod9:  48.43574797474338
## normality asumption:
## mod8:  ShapiroResult(statistic=0.9671643376350403, pvalue=0.00047616203664802015)
## mod9:  ShapiroResult(statistic=0.965816855430603, pvalue=0.00034187137498520315)
## paired t-test:
## Ttest_relResult(statistic=-3.8856065639237953, pvalue=0.00014633609210649466)
## Wilcoxon:
## WilcoxonResult(statistic=4761.0, pvalue=9.61605197477806e-05)
##
##
## alpha:
## means:
## mod8:  0.18113047681349354
```

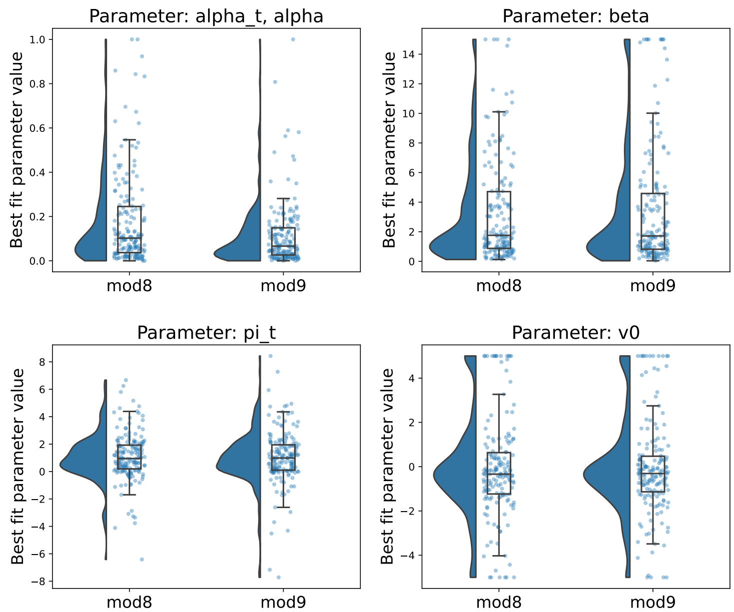

Figure 29: Parameters

```
## mod9:  0.11391557796897692
## normality asumption:
## mod8:  ShapiroResult(statistic=0.766213595867157, pvalue=3.5342551368139352e-15)
## mod9:  ShapiroResult(statistic=0.6862607002258301, pvalue=1.4018962722958882e-17)
## paired t-test:
## Ttest_relResult(statistic=6.986333592510897, pvalue=6.205745788256391e-11)
## Wilcoxon:
## WilcoxonResult(statistic=1553.0, pvalue=6.018819096847661e-19)
##
##
## beta:
## means:
## mod8:  3.2930702178297024
## mod9:  3.4459583450832176
## normality asumption:
## mod8:  ShapiroResult(statistic=0.7934234142303467, pvalue=3.1690155463974176e-14)
## mod9:  ShapiroResult(statistic=0.7512106895446777, pvalue=1.1391245299393021e-15)
## paired t-test:
## Ttest_relResult(statistic=-0.8245677093453079, pvalue=0.4107793372206133)
## Wilcoxon:
## WilcoxonResult(statistic=6077.0, pvalue=0.06396807304546269)
##
##
## pi_t:
## means:
## mod8:  1.0626103441929362
## mod9:  1.0308060195981652
```

```
## normality asumption:
## mod8:  ShapiroResult(statistic=0.9398530125617981, pvalue=1.3971595080874977e-06)
## mod9:  ShapiroResult(statistic=0.9111093282699585, pvalue=1.2255516601555883e-08)
## paired t-test:
## Ttest_relResult(statistic=0.45381132580971545, pvalue=0.6505468573621205)
## Wilcoxon:
## WilcoxonResult(statistic=6366.0, pvalue=0.16069971758461887)
##
##
## v0:
## means:
## mod8:  -0.11665941953886272
## mod9:  -0.09262325863891928
## normality asumption:
## mod8:  ShapiroResult(statistic=0.9229835867881775, pvalue=7.624829123642485e-08)
## mod9:  ShapiroResult(statistic=0.919902503490448, pvalue=4.672326170407359e-08)
## paired t-test:
## Ttest_relResult(statistic=-0.2631533724384663, pvalue=0.7927529234474232)
## Wilcoxon:
## WilcoxonResult(statistic=6258.0, pvalue=0.14670780158044705)
```

Parameter recovery

Simulation

Parameter values for simulation sampled from fitted parameter values (numpy.random.normal function):

$$param \sim \mathcal{N}(mean, std)$$

Dataset was randomly chosen from the 170 datasets (random.suffle function).

Model 8

```
## Model = model8
## Value function = rescorla_wagner_shrinking_alpha
## Decision function = my_softmax_shrinking_press_bias
## Parameters = v0, alpha_t, beta, pi_t
```

N sim = 1000 Simulation parameter values:

```
##             v0   alpha_t      beta      pi_t
## mean -0.116659  0.181130  3.293070  1.062610
## std   2.272556  0.208935  3.500643  1.764987
```

Plots

Model 9

```
## Model = model9
## Value function = rescorla_wagner
## Decision function = my_softmax_shrinking_press_bias
```

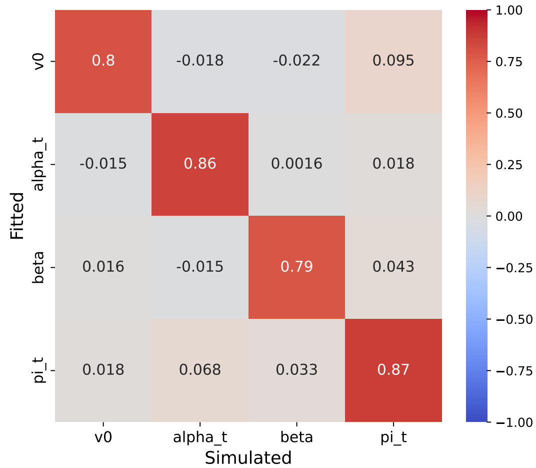

Figure 30: Confusion matrix mod8

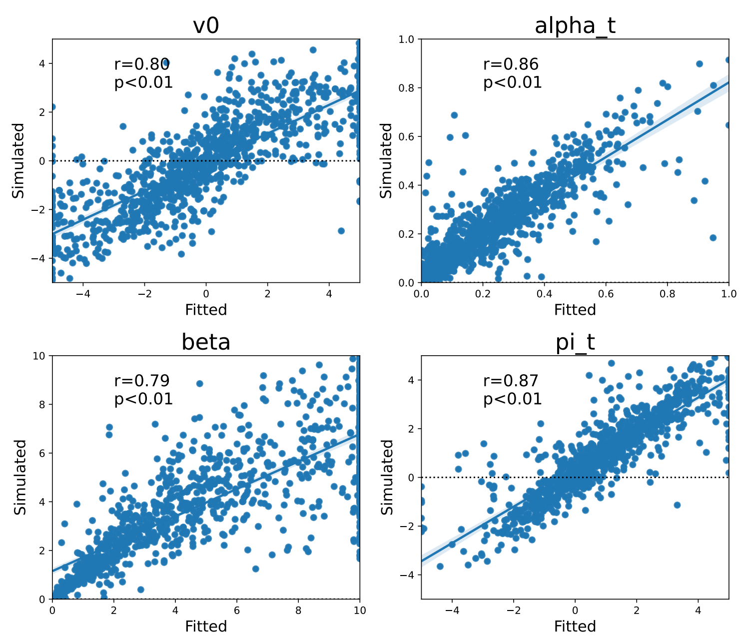

Figure 31: Conrrelations Model 8

```
## Parameters = v0, alpha, beta, pi_t
```

N sim = 1000 Simulation parameter values:

```
##              v0     alpha      beta      pi_t
## mean -0.092623  0.113916  3.445958  1.030806
## std   2.170045  0.142737  3.950705  2.004097
```

Plots

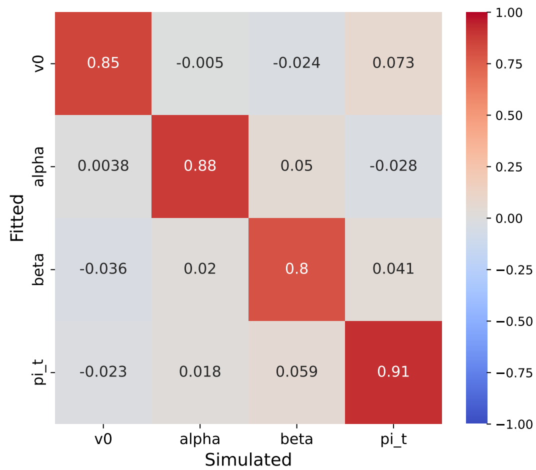

Figure 32: Confusion matrix Model 9

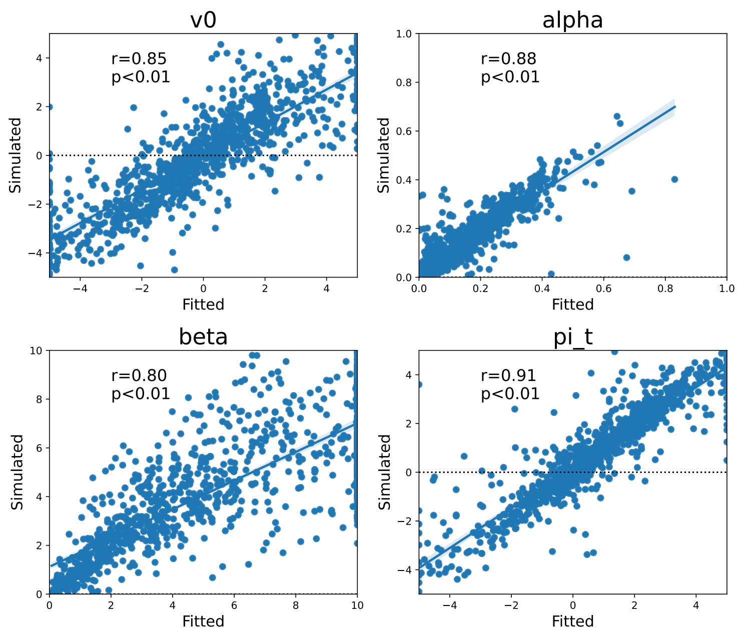

Figure 33: Conrrelations mod9