Alice-Simulation

Daniel Gotthardt

2024-08-15

prototype theory (attraction mechanism)

```r
# Use a large number of observations to reduce simulation error
n_obs <- 1000
# Draw prototype valence distribution from truncated normal distribution
prototype_valence <- rnormt(n_obs, c(0,1), 0.5, 0.2)
# Draw censored prosociality starting distribution
# Values below 0 are measured as 0, values above 1 are measured as 1
prosociality_pre <- rnormc(n_obs, c(0,1), 0.25, 0.2)

# Define attractor based on prototype valence
prototype_attractor_pos <- 0.5 * prototype_valence + 0.25
prototype_attractor_int <- 1 * prototype_valence
# Generate prosociality combining starting values and attraction mechanism
prosociality <- prosociality_pre +
  prototype_attractor_int * (prototype_attractor_pos - prosociality_pre)
```

exemplar theory (sampling mechanism)

```
# Draw average exemplar valence distribution from truncated normal distribution
exemplar_mean <- rnormt(n_obs, c(0,1), 0.5, 0.2)
# Draw exemplar variability from uniform distribution
exemplar_var <- runif(n_obs, 0, 1)
# Generate 100 memory episodes conditional on individual average & variability
exemplar_episodes <- sapply(1:n_obs, function(i) {
  rnormc(100,
         c(0,1),
         exemplar_mean[i], sqrt(exp(log(0.005)+8*exemplar_var[i])))
}
)
# Sample 10 episodes and calculate mean for each person
exemplar_sampled_mean <- apply(exemplar_episodes, 2, function(events) {
  sampled_events <- sample(events, 10)
  mean(sampled_events)
}
)
# Calculate expected prosociality
prosociality_exp <- 0.25 + 0.5 * exemplar_sampled_mean
# Generate prosociality as censored normal distribution
prosociality <- rnormc(n_obs, c(0,1), prosociality_exp, 0.05)
```