Multiplayer fisherman game - Pilot results

Natalia Vélez, Kelsey Allen

August 27, 2016

Methods

n = 9 triads (26 unique subjects) participated in a coordination game. Each player represented a "fisherman," who had to silently coordinate with the other players to (1) catch fish, and (2) clear trees from the road in order to sell the fish in the market. Each player had a different "strength," corresponding to the number of fish they could catch *and* the number of trees they could clear from the road. (For example, a player with strength 2 could either clear 2 trees or catch 2 sacks of fish.)

Participants played through 24 scenarios of the fisherman game. Each scenario was presented on a screen through an online game written in jsPsych. In each scenario, participants would have a different strength (between 1-3) and there would be a different number of trees blocking the road (between 1-3). Each participant received two action cards (one with a fish, and the other with a tree) that they could use to indicate their action on each particular day. Participants responded on each trial by placing their action card face-down in the center of the table; once all participants indicated the response, the cards were revealed, and the experimenter logged their actions through the online interface. If participants did not converge on a solution in the first try, they repeated the same scenario.

Scenarios were generated by orthogonally varying each player's strength and the number of trees to create 80 unique scenarios. (One scenario, in which all players have a strength of 1 and there are 3 trees blocking the road, was not included.) Of these, 8 had 3 possible solutions—for example, if there are 3 trees blocking the road ($T = 3$) and each fisherman has a strength of 3 ($S(A) = S(B) = S(C) = 3$), any one of the players could have cleared the tree. All 8 of these scenarios were included in the trial order. Of the remaining scenarios, 33 had 1 possible solution (e.g., $S(A) = 1$, $S(B) = 2$, $S(C) = 3$, $T = 2$) and 39 had 2 possible solutions (e.g., $S(A) = 1$, $S(B) = 2$, $S(C) = 3$, $T = 3$). For each new game, 8 scenarios were randomly sampled from each category. Scenarios were presented in a pseudo-randomized order; the trials were arranged into scrambled blocks consisting of one scenariow with 1 solution, one with 2, and one with 3.

Loading and cleaning up data

Because I forgot to code the number of solutions in each trial, the code chunk below is adding that information to the subject data in a roundabout way. :(

```r
scenario_info = fromJSON(file = 'scenarios.json')
data_files = list.files('pilot-results', pattern = '*.json')
fishing_data = NULL

# Helper function to find items in a list
lookup <- function(list, fun) {
  is_match = sapply(list, fun)
  return(which(is_match))
}

# Uses match_function to pull up information for each trial
match_scenario <- function(trees, strengths) {
  match_index = lookup(scenario_info,
                       function (x) x$trees == trees && all(x$strengths == strengths))
  return(scenario_info[[match_index]])
}

for (f in 1:length(data_files)) {
  session_data = fromJSON(file = file.path('pilot-results', data_files[f]))

  for (trial in 1:length(session_data)) {
    trial_data = session_data[[trial]]
    num_trees = trial_data$trees
    strengths = trial_data$strengths
    n_solutions = match_scenario(num_trees, strengths)$n_solutions

    fishing_data = rbind(fishing_data,
                         data.frame(
                           group = f,
                           trial_no = trial_data$num,
                           num_trees = num_trees,
                           p1_strength = strengths[1],
                           p2_strength = strengths[2],
                           p3_strength = strengths[3],
                           n_solutions = n_solutions,
                           p1_choice = trial_data$player_choices[1],
                           p2_choice = trial_data$player_choices[2],
                           p3_choice = trial_data$player_choices[3],
                           payoff = trial_data$payoff,
                           max_payoff = trial_data$max_payoff,
                           is_max = trial_data$is_max
                         ))
  }
}
```