Title: Studying reciprocity based on welfare tradeoff ratios using the 2D λ slider
Authors: Wenhao Qi, Lindsey J. Powell
Affiliation: Department of Psychology, UC San Diego

Research questions

1. Can participants infer the opponent's WTR from their decisions on the 2D λ slider?
2. Do participants adjust their own WTR toward the opponent based on their inference of the opponent's WTR?

Hypotheses

1. The participant's mean prediction of the opponent's decision on the 2D λ slider, which reflects the opponent's WTR, is positively correlated with the opponent's actual WTR.
2. The participant's mean WTR toward the opponent, as reflected by their decisions on the 2D λ slider, is positively correlated with the opponent's actual WTR.
3. The slope of predicting the participant's WTR from the round number is higher when the opponent's actual WTR is higher, and vice versa.

Methods

Experiment design

The participants play a game with a computer opponent who pretends to be a human through a web interface. In each round, either the participant or the computer (called the allocator) makes an allocation decision on a 2D λ slider, represented as a parabolic curve with a downward opening on a square board at the center of the interface. The participant is represented by a red icon on the left of the board, and the computer is represented by a blue icon on the right of the board. All elements in the interface are color-coded: red for the participant and blue for the computer. An allocation decision is made by choosing a point on the curve (by dragging a slider handle), whose horizontal and vertical locations correspond to a payoff for the participant and a payoff for the computer, each represented by the length of a horizontal or vertical bar. When the participant is the allocator, the origin is the lower-left corner of the board, the horizontal bar at the bottom of the board is the computer's payoff, and the vertical bar on the left of the board is the participant's payoff; when the computer is the allocator, the origin is the lower-right corner of the board, the horizontal bar at the bottom of the board is the participant's payoff, and the vertical bar on the right of the board is the computer's payoff. The payoffs are bounded at 0 and 100 in an arbitrary unit. There is a one-to-one correspondence between points on the 2D λ slider and the allocator's WTR toward the other player. The exact shape and location of the curve are randomly changed across rounds, but the range of WTR on the slider is always [-0.5,1.5].

The participant and the computer take turns making allocation decisions; i.e., the participant allocates in the first round (called a “participant’s round”), the computer allocates in the second round (called a “computer’s round”), etc. In each round, when the allocator (either the participant or the computer) is “thinking” about which point to choose, the other player (called the predictor) needs to predict the allocator’s decision by choosing a point on the slider. So in each round, either player takes one of two actions: allocate or predict, and two players act simultaneously without seeing the other player’s action. The computer pretends to spend some time (randomly generated) thinking about each action. When it is “thinking”, a loading icon is displayed next to its icon; when it “has acted”, the loading icon changes to a checkmark.

When both players have acted, the allocator’s decision is revealed to the predictor, but the predictor’s prediction is hidden from the allocator (this only matters when the target of revelation is the participant since the computer can see the participant’s actions anyway). Then the two payoffs resulting from the allocation decision are added to the two players’ total payoffs, respectively, via an animation. The total payoffs start at 0. The computer’s total payoff is hidden from the participant. Then, if it is a computer’s round, the participant receives a bonus payoff based on the accuracy of her prediction. The closer the predicted point on the slider is to the actual point the computer chose, the larger the bonus. Specifically, the bonus payoff is a linear interpolation between 0 (when the absolute deviation between the prediction and actual decision is 1 in terms of WTR) and 10 (when the absolute deviation is 0). When the absolute deviation is greater than 1 in terms of WTR, there is no bonus payoff. At this point, either the participant’s allocation decision or both the computer’s allocation decision and the participant’s prediction are shown on the board. Before going to the next round, the participant can drag the slider to change the location of the allocation handle and the corresponding payoffs, in order to explore what other options were available to the allocator. Once she releases her mouse, the handle will pop back to the actual decision.

There are 20 rounds in total, 10 of which are participant’s rounds and the other 10 are computer’s rounds. The participant always allocates first.

There are three types of attention checks in the experiment. In some rounds (called “discard rounds”), the payoff bar for the non-allocating player is colored gray instead of the usual red or blue. This indicates that this payoff will be discarded instead of being given to the non-allocating player, so the rational decision for the allocator is to maximize her own payoff and choose the WTR=0 point on the slider, regardless of her WTR toward the other player. In some other rounds (called “self rounds”), the payoff bar for the non-allocating player is in the same color as the allocator’s payoff bar. This indicates that this payoff will be given to the allocator instead of the non-allocating player, so the rational decision for the allocator is to maximize the sum of the two payoffs and choose the WTR=1 point on the slider, regardless of her WTR toward the other player. The third type of attention check is a memory check after the participant clicks “Next round”. The participant is asked to reproduce the decision of the allocator in that round on the same slider. There are two discard rounds, two self rounds, and two memory checks in the experiment.

Before the real game, the participant signs a consent form, walks through an interactive tutorial, plays 6 practice rounds of the game, and waits for a (simulated) pairing with “another participant”. The tutorial consists of 34 steps that explain all the details of the game. During the practice rounds and the real game, the participant can revisit the tutorial at any point. In the practice rounds, the computer acts very quickly because the participant is told that the other player is a computer agent. All three types of attention checks are featured in the practice rounds. The wait duration for the pairing to succeed is randomly sampled from a uniform distribution between 5 and 30 seconds.

After the real game, the participant fills out a debriefing survey with 6 questions:

1. “In general, how nice (or mean) were you towards the other participant?” The participant answers on a 17-point slider with 5 labels, ranging from “Extremely mean” to “Extremely nice”.
2. “In general, how nice (or mean) do you think was the other participant towards you?” The participant answers on the same slider as Question 1.
3. “Did you adjust your niceness (or meanness) towards the other participant according to how nice (or mean) they were towards you?” The participant can choose “Not at all” or “Yes, to some extent”. If they choose the former, they need to fill out a text box explaining “Why not”. If they choose the latter, they need to fill out a text box describing “How did you adjust”.
4. We disclose to the participant that the other player was actually a computer agent. Then we ask them how much they believed that the other player was a real human, either at the beginning or near the end of the real game. They answer on two 17-point sliders, each with 5 labels, ranging from “Not at all” (coded as -7) to “Absolutely” (coded as +7). If either of these two answers is ≤0, the participant needs to fill out a text box describing “What made you suspicious that the blue player wasn’t a real human”.
5. “Did you find any part of the experiment confusing?” If the participant chooses “Yes”, they need to describe it in a text box.
6. “Did you encounter any technical problems?” If the participant chooses “Yes”, they need to describe it in a text box.

Conditions

There are three between-subjects conditions, where the computer’s WTR is fixed to 0, 0.5, or 1. The computer’s actual allocation decisions are given by this fixed WTR plus a small random perturbation to make it seem more plausible to the participant. The perturbation is sampled from a uniform distribution between -0.1 and 0.1 in terms of WTR. The participants are unaware of the conditions they are assigned to

To control for any effect of raw payoffs, the three conditions are yoked in the sense that the raw payoffs from the computer’s allocation decisions are fixed across the conditions for a given computer’s round. For instance, the payoffs from the computer’s allocation decision in Round 2 are 41.1 for the participant and 78.8 for the computer, regardless of the condition. For a given

computer's round, the location of the slider is varied across the conditions to reflect the computer's WTR in a particular condition.

Planned sample

As of the submission of this research plan for preregistration, the data have not yet been collected.

Our main analyses for Hypotheses 1 and 2 involve a simple Pearson correlation between two variables, where each participant is a data point. We would like to be able to detect medium effect sizes of r=0.3 with α=0.05 and power=0.8, which corresponds to a sample size of 84. Therefore, we plan to recruit 90 participants, 30 in each condition, on Prolific. If some participants are excluded from data analysis based on the exclusion criteria below, we will keep recruiting new participants until the desired sample size is reached. The participants will be drawn from the "standard sample", be located in the USA, be fluent in English, have an approval rate of at least 95%, and have at least 10 previous submissions on the platform.

It takes about 15 minutes to complete the experiment. Each participant will be paid $3 based on a rate of $12/hour. They will also receive a bonus payment of at most $1 after the experiment. The bonus is determined by the participant's total payoff at the end of the experiment. We set a reasonable lower bound (RLB, which is not the actual lower bound) and an upper bound (UB) on the total payoff. RLB is the total payoff of a participant whose WTR is always 1, whose average prediction error is 0.5 in terms of WTR, and who acts rationally in the "discard rounds" and "self rounds". UB is the total payoff of a participant whose WTR is always 0, whose prediction error is always 0, and who acts rationally in the "discard rounds" and "self rounds". The bonus payment is a linear interpolation between $0 (when the total payoff is RLB) and $1 (when the total payoff is UB). If the total payoff is lower than RLB, there is no bonus payment.

Exclusion criteria

A participant is considered to fail the attention check of a "discard round" if their allocation decision or their prediction of the computer's allocation decision deviates from the rational decision (WTR=0) by more than 0.3 in terms of WTR. This threshold is increased to 0.5 for "self rounds" because it is harder to maximize the sum of the two payoffs than only one payoff. A participant is considered to fail a memory check if the slider location they reproduce deviates from the actual slider location by more than 0.3 in terms of WTR. A participant is excluded from data analysis if they fail two or more of the six attention checks.

Analysis plan

Hypothesis 1

For the following analyses, the independent variable is the computer's actual WTR (0, 0.5 or 1 depending on the condition), and the dependent variable is the participant's mean prediction of the computer's WTR across all 10 "computer's rounds".

The primary analysis is the Pearson correlation between the two variables. We will conduct a two-tailed t test and a permutation test on the correlation with $\alpha=0.05$ for statistical significance. We will use the Bayesian bootstrap and weighted Pearson correlation to determine the 95% confidence interval of the correlation.

The secondary analysis is the pairwise comparisons between conditions. We will compare the group mean of the dependent variable between each pair of conditions using a two-sample two-tailed t test and a permutation test with $\alpha=0.05$. We will use the Bayesian bootstrap and weighted mean to determine the 95% confidence interval of the difference.

Hypothesis 2

The analyses for Hypothesis 2 are identical to Hypothesis 1, except that the dependent variable is replaced with the participant's mean enacted WTR toward the computer across all 10 "participant's rounds".

Hypothesis 3

We will fit a linear mixed-effects model to the data. The dependent variable is the participant's enacted WTR toward the computer in each "participant's round". The fixed effects are the computer's actual WTR, the round number, and their interaction. The random effects are the participants' individual intercepts and slopes with respect to the round number. The R formula will look like `participant_wtr ~ computer_wtr * round_number + (round_number | participant_id)`. We will first use the `lmerTest` package to obtain a (restricted) maximum-likelihood fit of the model, and then use the `brms` package to obtain a Bayesian fit of the model. We will compare the results of these two fits.

We will first treat `computer_wtr` as a continuous variable. Then the interaction term between `computer_wtr` and `round_number` corresponds to the expected increase in the slope of `round_number` when `computer_wtr` increases by 1. A significantly positive interaction term would support our hypothesis. For the maximum-likelihood fit, we will look at the p-value and the 95% confidence interval. For the Bayesian fit, we will look at the probability of direction and the 95% highest posterior density interval (using the `bayestestR` package).

We will then treat `computer_wtr` as a discrete variable (factor) and compare the slopes of `round_number` between each pair of conditions using the `emmeans` package.