Pattern-oriented modeling of commons dilemma experiments

Marco A Janssen[1,2,3], Nicholas P. Radtke[2,3], and Allen Lee[1,3]

[1] School of Human Evolution and Social Change, Arizona State University, PO Box 872402 , Tempe AZ 85287-2402, USA
[2] School of Computing and Informatics, Arizona State University
[3] Center for the Study of Institutional Diversity, Arizona State University
{Marco.Janssen, Nicholas.Radtke, Allen.Lee}@asu.edu

Abstract. Experimental data from a spatially explicit dynamic commons dilemma experiment is used to empirically ground an agent-based model. Four distinct patterns are identified in the data. Two naïve models, random walk and greedy agents, do not match the patterns. A more comprehensive model is presented that explains how agents make movement and harvest decisions. Using pattern oriented modeling the parameter space is explored to identify the parameter combinations that meet the four identified patterns. Less than 0.1% of the parameter space meets all the patterns. The resulting model can be used to design future experiments.

Keywords: empirically grounded agent-based modeling, commons dilemma, individual decision making

1 Introduction

One of the main challenges for the use of agent-based modeling as a scientific method is to create empirically grounded models. Due to the stochastic and non-linear nature of agent-based models, calibration of models is a non-trivial task. The challenge of improving our methods for empirical grounding of agent-based models has been addressed in a number of recent papers [1,2,3]. Which approaches are appropriate depend on the specific research questions, available data and the dynamics of the candidate agent-based models.

In this paper we will use experimental data, as we perform group experiments on commons dilemmas where participants make many decisions in a real-time dynamic environment representing a virtual common resource. We have a large amount of highly detailed data that captures each and every in-game decision made by our participants, presenting an interesting case study for exploring the challenge of developing empirically grounded agent-based models. Unlike traditional approaches in behavioral game theory we are not able to make use of maximum likelihood approaches since we do not know the underlying probability distributions of the models tested. Since the experiments illuminate patterns at different scales, we are testing the challenges of using pattern-oriented modeling for agent-based modeling of human behavior [1].

In the next section we will discuss the experiment and the patterns from the experimental data. We then compare the observed patterns with some naïve models. In the following section we discuss the agent-based model developed to mimic the observed patterns. We then calculate for a large set of parameter combinations whether the model fits observed patterns and define subsets of acceptable models.

2 Spatially explicit commons experiments

Many natural resource problems can be classified as commons dilemmas, a dilemma between the interest of the individual and the interest of the group as a whole [4]. A common-pool resource (CPR) is such a dilemma, where a resource is shared by multiple users. CPRs are characterized by the fact that it is difficult to exclude users, and units appropriated by one user are not available anymore for other users. Examples of CPRs include forests, pastures, irrigation systems, and fishing grounds.

When individual interests dominate, a conflict on common resources may lead to the tragedy of the commons in open access situations [5]. However, many empirical studies have shown that people are able to govern common resources effectively [4]. A typical way to study the fundamental processes on how individuals are able to self-govern themselves in such commons dilemmas are controlled laboratory experiments. In those experiments participants interact with an abstract resource during a number of rounds and derive monetary incentives.

The experimental data used is part of a series of experiments performed to study how groups develop new institutional arrangements if they share common dynamic resources. For more information on the experimental design we refer to [6].

Participants were recruited from a large database of undergraduate students at Arizona State University during the Spring semester of 2007. The average age was 21.4 years and 67% of the participants were male. Data used for this paper is a subset of the actual data, using only round two of the experiment. In the experiment, the participants first have a practice round on an individual plot, and the first round of the actual experiment is a one-person experiment. After round 2 participants are allowed to use text chat to coordinate their actions. We will use in the paper only data from round 2, before the participants start communicating. Data from 16 groups and 64 individuals is used.

In the experiment, groups of four randomly assigned participants share a renewable resource that grows on a 28 X 28 spatial grid of cells. Participants implicitly harvest a green token by moving their virtual avatar's location on top of the token. They move their avatar by pressing the arrow keys (left, right, up, and down). There are two modes, implicit and explicit, that can be toggled by pressing the 'M' key. In explicit mode one can move around without automatically harvesting tokens. When one wishes to harvest a token, they must press the spacebar when their avatar is on a cell with a token.

The resource renewal rate is density dependent. The probability that a green token will appear on an empty cell increases as the number of green tokens neighboring that empty cell increases. The probability p_t is linearly related to the number of neighbors: $p_t = p*n_t/N$ where n_t is the number of neighboring cells containing a green token, and

N the number of neighboring cells (N = 8 because we use a Moore neighborhood). The parameter p is defined in such a way that the renewal of the resource is quick enough to be observed by the participants, but sufficiently slow that the participants experience a dilemma between immediate, individual benefits and longer-term, group benefits. If participants collect tokens as quickly as they can, there will soon be no tokens remaining on the screen. Once every token has been harvested, no further opportunity exists for any new tokens to be created.

The participants have four minutes to collect tokens and each token is worth $0.02. Two treatments are considered: A low growth case with p equal to 0.01 and 25% of the cells initially populated with tokens, and a high growth case with p equal to 0.02 and 50% of the cells initially populated with tokens. We have 6 groups for the low growth treatment and 10 groups for the high growth treatment.

3 Patterns

We discuss four different patterns discovered in our data analysis. These are the patterns we wish to replicate with our agent-based model. The first pattern is the number of tokens on the screen over time. Figure 1 shows the mean as well as the range (+/- standard deviation) for both the low and high growth conditions.

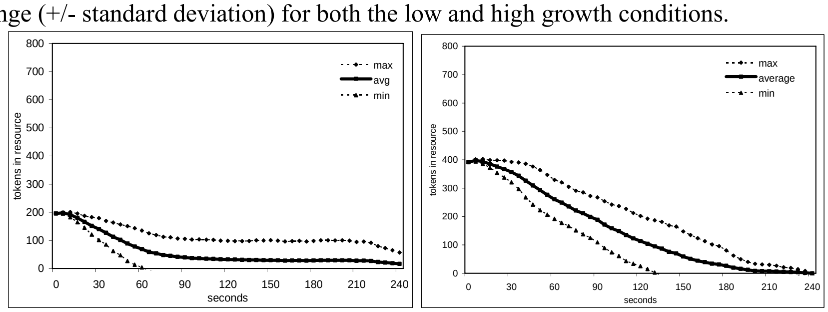

Fig. 1. Average number of tokens on the screen every five seconds for round 2. Left is the low growth case and right is the high growth case. The max and min refer to the average +/- one standard deviation.

The second pattern is the relative inequality of tokens collected within a round. We calculated the share of tokens collected by participant i compared to the group total. Then we calculated the gini coefficient for each group. If all participants collect equal numbers within their group, the gini coefficient would be 0. If one participant takes everything, the gini coefficient would be 1.0. We find that the gini coefficient for the low growth rate experiments is higher on average compared to the high growth rate experiments: 0.23 vs 0.11. This means that there is more inequality when resource regrowth is low.

Table 1: Average gini coefficients and standard deviations of groups, as well as the lower and upper boundaries used for the pattern oriented modeling.

	Mean	Stdev	Lower value	Upper value
Low growth	0.2323	0.1294	0.1029	0.3617
High growth	0.1124	0.0563	0.0561	0.1787

The average number of tokens collected by groups during the four minutes of the experiment is obviously lower for the low growth rate treatment compared to the high growth rate treatment. This information is used as the third pattern.

Table 2: Average number of tokens collected per person, as well as the lower and upper boundaries used for the pattern oriented modeling.

	Mean	Stdev	Lower value	Upper value
Low growth	274.0	63.75	210.25	337.75
High growth	743.1	114.1	629.0	857.2

The fourth pattern originates from our observation that participants have a tendency to continue in the direction they are moving instead of changing direction. There is a slight cost of changing direction (hitting a different key on the keyboard). Since we record every in-game action of every participant we can analyze how often a participant moves in the same direction until (s)he changes direction. In both treatments we observe that participants move in straight lines more often than they would if they were simply making random directional choices.In our pattern oriented analysis we clamp the number of moves in the same direction to ten straight moves, so any additional moves past the first ten is disregarded..

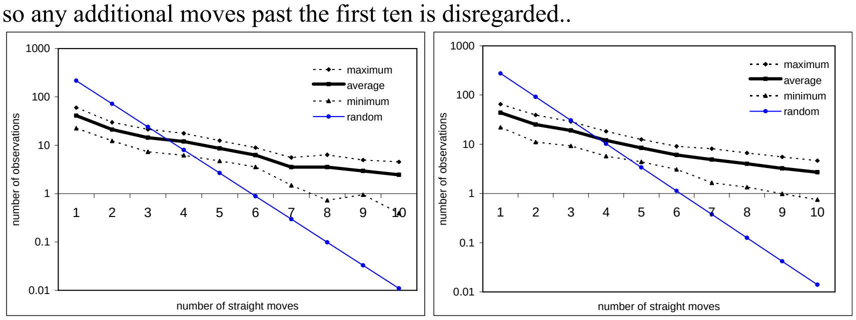

Fig 2. Round 2 data for low growth rate (left) and high growth rate (right) for number of straight moves. As a reference we include the relationship when participants make random moves.

Besides these four patterns we would like to explain, we extracted two patterns from the data that we will use as inputs: speed of movement and the use of explicit mode.

First, we determine the percentage of participants that use the explicit mode. With the higher growth rate fewer participants use the explicit mode. In the model we assume an exogenous fraction of agents using the explicit mode during the round. More in depth analysis shows that participants do not frequently change their mode during the round. As a result we will use 30% explicit mode for the low growth treatment, and 15% for the high growth treatment.

To determine the base speed of the participants, we calculated the average speed per second between 10 and 70 seconds. We ignored the first 10 seconds, since participants needed to get up to speed when the round started, and after 70 seconds some resources started to be depleted. Figure 3 shows that the distribution of the average number of movements per second can be approximated by a normal distribution with a mean of 3 and a standard deviation of 0.8.

Given the mean number of key presses per second, we assume a normal distribution of speed variation per second. The data shows a higher standard deviation when the number of key presses per second is higher: stdev = 0.384 * Mean

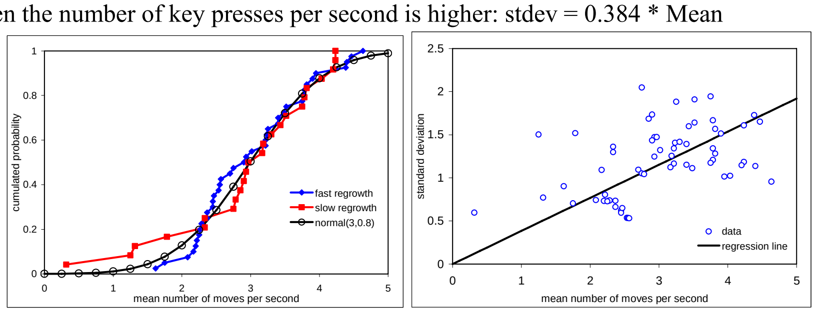

Fig. 3. Distribution of average clicks per second (left). Relation between standard deviation and the number of clicks per second (right).

4 Some naïve models

We will compare the comprehensive model with some simple, naïve models. If naïve models can explain the observed patterns well, there is no need for a more complicated model. The naïve models we use are:

- Random walk: agents are in implicit mode and randomly move around on the resource.
- Greedy agents: agents always move towards one of the nearest tokens.

We see in the Figure 4 that the random walk agents lead to a much lower number of tokens collected and therefore a much higher number of tokens in the resource. On the other hand, greedy agents rapidly reduce the resource size, much faster than within the range of observed patterns.

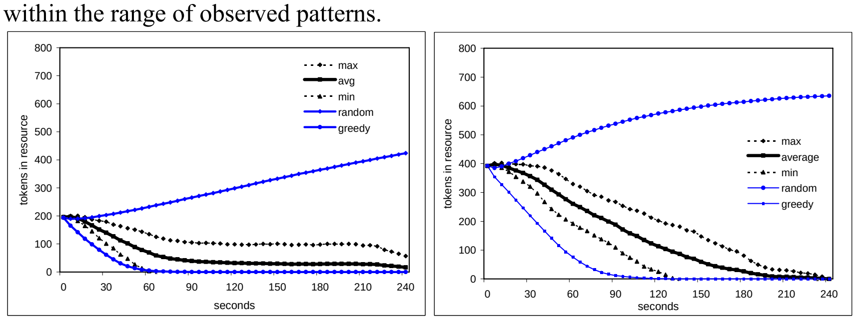

Fig. 4. Results of naïve model in relation with the empirical patterns. On the left side the low growth case and on the right the high growth case.

The naïve models cannot explain the other patterns either. The random agents collect too few tokens (73 for low growth case and 173 for high growth case). The greedy agent is within the observed pattern of the low growth case (223), but too low for the high growth case (513). The agents change direction more frequently for both the random and the greedy agents compared to observed distributions.

5 Model description

The model simulates the token regeneration as has been implemented in the experimental software using time steps of 1 second. For high growth rate cases p is equal to 0.02 and for low growth rate cases p is equal to 0.01. Agents cannot pass through the borders. In the initialization, we allocate 392 (50%) or 196 (25%) tokens randomly on the unoccupied cells, for high and low growth respectively.

Our experimental data tells us that some agents make, on average, more moves than others, and that an agent makes more moves in some seconds than other seconds as defined in the previous section.

For every move the agent must decide where to go to. This module is based on [7], who developed an empirical model for a similar experimental environment. The basic idea is that each agent defines the value of each token at the screen as a target to move to. This value is built up by four components.

- the closer a token is to the agent, the more valuable
- the closer a token is to the current target, the more valuable
- the more competing agents close to a token, the less valuable
- tokens that are straight ahead in the current directional path of the agent are more valuable.

As such we formulate the value of a token at location (i,j) as follows:

$$V_{i,j} = P_1 \cdot \left(\frac{1}{\text{tokendistance}} \right) + P_2 \cdot \left(\frac{1}{1 + \text{currentargetdistance}} \right) - P_3 \cdot agentdensity + [P_4]_A$$

where agent density is the number of agents in radius R_{P3} around the potential target.

One of the tokens will be drawn based on the relative value among all the tokens. When the agent reaches this position, it will select a new target. An agent in explicit mode does not consider a token in its same cell to be part of the eligible set of tokens.

Using probabilistic choice, the probability of having a token at location (i,j) as the target is defined as

$$P[T(i,j)] = \frac{e^{\beta \cdot V_{i,j}}}{\sum_{i,j} e^{\beta \cdot V_{i,j}}}$$

where β is the parameter that defines how sensitive agents are to differences in the value of the tokens.

Based on the chosen target, the agent defines the direction of the target and decides to go up, down, left, or right. When a move is made, and the agent is on a cell with a token, the agent automatically collects the token when in implicit mode. However, when in the explicit mode, the agent needs to decide whether or not to collect the token. We assume that agents are more tempted to take the token when more tokens are around the cell, and the probability to collect the token is defined as

$$P[collect] = \frac{x^b}{x^b + a^b}$$

where x is the number of tokens in the eight cells of the Moore neighborhood divided by 8. The parameters a and b define at what density the probability is 50% (x=a) and how steeply the probability increases with higher values of x.

Next to a few parameters we directly relate to the observations (speed and mode) we have eight parameters that we use for the calibration: P_1, P_2, P_3, P_4, R_{P3}, β, a, b

6 Models that meet patterns

Pattern oriented modeling argues that acceptable models are those that are able to match all observed patterns. Given the stochastic nature of the models, matching observed patterns is defined as generating average statistics that are within defined ranges for the patterns. This is a somewhat subjective exercise, but since we performed a number of experiments, averages and standard deviations of the patterns can be determined. We simply assume that the acceptable level of generated averages of patterns lies between the average +/- standard deviation.

The model is implemented in Mason [8] and we ran the model one hundred times for each parameter combination. For each of the eight parameters we vary the model for four different values, leading to 4^8 = 65536 parameter combinations (Table 3):

Table 3: Parameter values used in exploring the parameter space for matching the four patterns.

Parameter	Description	Parameter values
P_1	Weight of tokens nearby	0.1; 0.4; 0.7; 1.0
P_2	Weight of tokens nearby current target	0.1; 0.4; 0.7; 1.0
P_3	Weight of other agents nearby token	0.1; 0.4; 0.7; 1.0
P_4	Weight of clicking straight direction	0.1; 0.4; 0.7; 1.0
Β	Agent sensitivity to differences in values for tokens	1; 4; 8; 13
A	[explicit] parameter affecting threshold	0.1; 0.4; 0.7; 1.0
B	[explicit] Steepness of curve	1; 4; 8; 13
Radius	Radius to determine the number of agents nearby token	1; 4; 7; 10

Patterns 2 and 3 are not discriminating. However, pattern 4, the straight line movements of the avatars is met by only about 10% of the parameter combinations (Table 4). For the low growth case 2457 parameter combinations meet all four patterns, while only 90 parameter combinations meet all four patterns for the high growth case. 37 parameters combinations meet all patterns in both data sets. In those cases P_1 is 0.7, P_2 is 0.1, P_3 varies between 0.4 and 1, P_4 is 0.4, the radius is 10, β is 8, a and b vary between 0.1 and 1, and between 1 and 10, respectively. This indicates that agents tend to select targets based on the number of tokens around the target cell and the number of agents in the neighborhood. Since explicit mode is used by only a small fraction of the agents, the parameters a and b are not found to be discriminating. When 2 or more agents are in the radius of 10 around a token, this token is not appealing as a target, except when the token is one or two cells from the avatar.

Table 4: The number of parameter combinations that meet the patterns for the low and high growth conditions.

	Pattern 1 Tokens left	Pattern 2 inequality	Pattern 3 Tokens collected	Pattern 4 Direction	All patterns
Low growth	29098	65530	47902	4264	2457
High growth	7133	33029	39459	7296	90

7 Conclusions

In this paper we apply pattern-oriented modeling [1] to identify parameter combinations that meet empirical patterns from a number of human subject experiments. We find that less than 0.1% of the parameter space meets all of the patterns, and as such it helped to successfully narrow down the parameter space of acceptable parameter combinations.

Although pattern oriented modeling has mainly been used in ecology, we think that this approach is also suitable for the empirical analysis of agent-based models in the social sciences.

Acknowledgments. We acknowledge support from the National Science Foundation

References

1. Grimm,V. Revilla, E. Berger, U. Jeltsch, F., Mooij, W.M., Railsback, S.F., Thulke, H.-H., Weiner, J., Wiegand, T., DeAngelis, D.L.: Pattern-Oriented Modeling of Agent-Based Complex Systems: Lessons from Ecology. *Science* 310: 987-991 (2005)
2. Janssen, M.A. and E. Ostrom, Empirically based agent-based modelling, *Ecology and Society 11(2): 37* (2006)
3. Robinson, D.T., D.G. Brown, D.C. Parker, P. Schreinemachers, M.A. Janssen, M. Huigen, H. Wittmer, N. Gotts, P. Promburom, E. Irwin, T. Berger, F. Gatzweiler, and C. Barnaud, Comparison of empirical methods for building agents-based models in land use change, *Journal of Land Use Science* 1(1): 31-55 (2007)
4. Ostrom, E. R. Gartner and J. Walker Rules, Games and Common Pool Resources, University of Michigan Press (1994)
5. Hardin, G. The Tragedy of the Commons, *Science* 162, 1243-1248 (1968)
6. Janssen, M.A., R.L. Goldstone, F. Menczer and E. Ostrom, Effect of rule choice in dynamic interactive spatial commons, *International Journal of the Commons*, in press
7. Roberts, M.E., Goldstone, R.L.:. EPICURE: Spatial and Knowledge Limitations in Group Foraging. *Adaptive Behavior* 14: 291-313 (2006).
8. http://www.cs.gmu.edu/~eclab/projects/mason/