TRANSFERABLE STRATEGIC META-REASONING MODELS

BY MICHAEL WUNDER

Written under the direction of

Matthew Stone

New Brunswick, New Jersey

October, 2013

ABSTRACT OF THE DISSERTATION

TRANSFERABLE STRATEGIC META-REASONING MODELS

by MICHAEL WUNDER

Dissertation Director: Matthew Stone

How do strategic agents make decisions? For the first time, a confluence of advances in agent design, formation of massive online data sets of social behavior, and computational techniques have allowed for researchers to construct and learn much richer models than before. My central thesis is that, when agents engaged in repeated strategic interaction undertake a reasoning or learning process, the behavior resulting from this process can be characterized by two factors: depth of reasoning over base rules and time-horizon of planning. Values for these factors can be learned effectively from interaction and are transferable to new games, producing highly effective strategic responses. The dissertation formally presents a framework for addressing the problem of predicting a population's behavior using a meta-reasoning model containing these strategic components. To evaluate this model, I explore several experimental case studies that show how to use the framework to predict

and respond to behavior using observed data, covering settings ranging from a small number of computer agents to a larger number of human participants.

Preface

Captain Amazing: I knew you couldn't change.

Casanova Frankenstein: I knew you'd know that.

Captain Amazing: Oh, I know that. AND I knew you'd know I'd know you knew.

Casanova Frankenstein: But I didn't. I only knew that you'd know that I knew. Did you know THAT?

Captain Amazing: Of course.

-Mystery Men

Portions of this dissertation are based on work previously published by the author [Wunder et al., 2010, 2011, 2012, 2013].

Acknowledgements

I would like to extend my gratitude to all those who have made contributions in support of my completion of a Ph.D.

First and foremost, my advisors Michael Littman and Matthew Stone enabled me to satisfy my curiosity in a number of directions until something stuck. I profoundly appreciate their complimentary roles in helping me develop as a researcher and a computer scientist. Matthew constantly encouraged me to focus my thinking and clarify my writing with his sharp lines of questioning, and helped me to understand the broader context of this research project. Through my interactions with Michael, I benefited immensely from his experience and good nature. His feedback and advice have shaped me in too many ways to count and have been invaluable throughout this process.

I could not have hoped for a stronger committee, which includes Kobi Gal, Haym Hirsh, Michael Littman, and Matthew Stone. Their diverse views and inputs during the defense process made it a very rewarding experience.

Over the years, I have been supported by several grants from the National Science Foundation that made my research experience both possible and fruitful. The support via HSD-0624191, in particular, was crucial during the early years of my study. Later, NSF IIS-1018152 and IIS-1017811 were of great assistance.

This dissertation depends heavily on prior work, and without my co-authors it never would have come to fruition. Collaborators who share much of the

credit for this earlier work include Monica Babes-Vroman, Michael Kaisers, Michael Littman, Siddharth Suri, Duncan Watts, and Bobby Yaros. I have learned an incredible amount through my discussions with everyone. A special thanks goes to Bobby who generously helped me proofread portions of this document, and whose door was always open for me to stop by and kick around my latest ideas.

Another important contributor to this research was Martin Zinkevich, who invented the Lemonade Stand Game and his tireless efforts made the competition possible. The four tournaments taking place during these past four years provided a crucial outlet to apply my research and test my ideas in the field. I am glad I was able to be a part of them, and it was a lot of fun working with the Rutgers dream team of Michael Kaisers and Bobby Yaros to design the perfect agent.

To all the members of the RL^3 lab, including John Asmuth, Monica Babes-Vroman, Carlos Diuk, Sergiu Goschin, Lihong Li, Chris Mansley, Vukosi Marivate, Ali Nouri, Tom Walsh, and Ari Weinstein, thanks for the memories and punctual lunches. You all made the lab a friendly and stimulating place.

A warm thanks also goes to everyone else I met at Rutgers who helped me develop as a scientist and a person, as well as all of my other friends who have accompanied me on this journey.

Finally, I would like to thank my loving and supportive family. My mom, Patricia, and sister, Beth, were always there for me as I made my way over the years. I am eternally grateful for your non-stop encouragement and faith in me.

Dedication

To Mom and Beth.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

The incredible growth of the Internet continues to change our lives, whether we look at our careers, social networks, or the way we read news. A recent survey by the Interactive Advertising Bureau points out that between 2007 and 2011 the number of jobs relying on web advertising doubled to over 5 million, accounting for over $500 billion in the economy, at a time when job growth was anemic at best. [1] Part of this growth results from the emergence of online social networking and web personalization, which has created an avalanche of new data about the diverse behaviors of individual users. This same period witnessed the tremendous destructive effects of the financial crisis, which was for the most part unpredicted by mainstream economists, creating a simultaneous intellectual crisis within that discipline [Krugman, 2009]. A major connecting thread between these phenomena is that they deal with the interaction of strategic individuals who implement the ability to reason about others as well as some capacity to learn and respond to their world. A deeper understanding of the dynamics at work in complex social systems could lead to fantastic advancements.

This dissertation confronts the challenge of building predictive models of self-interested agents who interact in strategic environments over time. On the surface, this type of problem falls under the domain of economics. I will

[1] http://www.iab.net/insights_research/industry_data_and_landscape/economicvalue

argue that, based on new evidence from experiments with people, traditional models are not up to the task of either describing behavior or prescribing effective behavior. Fortunately, the ever-increasing access to both large amounts of computational power and relevant social data has opened the door to a new empirical program for looking at these questions. The richness and diversity of the data at hand has also brought to light the need for better models, which my research has attempted to address.

1.1 The Multiagent Cycle

The advances described above have led to a novel scientific methodology that I will henceforth term the *Multiagent Cycle*. It is meant to describe the decision-making process of participants in strategic environments who employ some combination of reasoning, learning, and planning. The players, or *agents*, in these environments constitute a population, and can be either human or computer. The population of human players or software agents behave according to some internal and emergent dynamics and are also affected by the rules and payoffs of their game environment.

The Multiagent Cycle has four main stages:

- Hypothesis generation and theory
- Experimentation and observation
- Data mining and modeling
- Computational simulation and parameterized space search.

Some of these paradigms, defined as the set of practices that define a scientific discipline [Kuhn, 1970], have existed separately for a long time. However, in order to study complex social systems, scientists need to incorporate all of them together. Game theorists have long proposed models of strategic behavior, and have worked out a vast mathematical theory around the idea that social systems tend toward equilibrium. Behavioral game theorists put these theories to the test in lab experiments with real people who play these games for rewards [Ho et al., 1998; Costa-Gomes et al., 2001]. While this line of research is still relatively new, these experimenters have discovered that people play in ways that do not entirely conform to the theoretical predictions. In turn, they have proposed new models that incorporate heuristics and cognitive shortcuts, suggesting that thinking hard to "solve" a game entails an extra cost that is not accounted for in the original equations.

Since the earliest days of computers, data analysts have constructed exquisite statistical approaches to make sense of the large and growing mass of information that results when lots of people interact with each other or the entity collecting that information. Much of the time, however, the patterns and features used in data mining is stripped of its strategic or cognitive meaning, and reduces to the same basic approach as could be used with any natural or physical phenomena.

The fourth thread, computer simulation, is growing in importance for modeling social behavior through the fields of *agent-based modeling* and *social simulation* [Axelrod, 1997; Macy and Willer, 2002]. Although this new type of modeling allows for greater complexity and richness that is not present in a set of linear equations, the agent models are often built from preexisting assumptions or in a way that satisfies the model-builder's goals [Lazer and Friedman,

2007; Fang et al., 2010; Mason and Watts, 2012]. As a result, these models are sometimes detached from reality. What is required, therefore, is a coherent meta-theory for explaining and predicting social behavior that also can be supported and refined by empirical evidence.

Now, a more seamless linkage between these schools of thought is possible due to recent advances in computational power and online data collection. For the first time, it is possible for social scientists to gather data for their theories on previously unattainable scales, learn complex models of behavior, and compute a dynamic system of social interaction informed by the observations. The explosion of computational power makes it feasible to search a parameter space for anomalous properties of these heterogenous agent models, which researchers can then use to build new theories and hypotheses to test. One might say that people perform a miniature version of this cycle every time they enter a game of poker, but our goal is a formal and systematic analysis. The aim of this work is to add an additional layer to social data analysis which incorporates some degree of self-reflection or theory of mind. Without this layer, a model can lack structure that results from awareness of other decision-makers, which can result in inaccurate predictions.

A major application of this work is to model actors in financial markets. Leading theories claim that market prices incorporate all available information, so that they reflect value perfectly, in an intellectual framework known as the Efficient Markets Hypothesis [Fama, 1970]. In practice, this theory cannot explain the periodic cycle of bubbles and busts that have occurred frequently throughout history. The presence of "irrationality" seems to be universal when people deal strategically with one another, even as the degree of sophistication

among individuals varies widely. These observations have been made experimentally [Ho et al., 1998] as well as in real markets, and therefore need to be addressed. Modeling the amount of irrationality in a population, as well as simulating its effects, is a precursor for regulators to implement effective policies to prevent or manage financial turmoil. Otherwise, they will be forever left to react to these crises in an *ad hoc* and ultimately self-defeating manner. Some economists have begun moving in this direction to establish a foundation for agent-based economics [Fagiolo et al., 2007b], but their lack of rigorous training in programming or machine learning algorithms in most economics programs is an obstacle in this endeavor.

Because of the nature of social interaction, agent-designing researchers face great algorithmic challenges arising from the complexity of many independent actors interacting in a dynamic domain. As such, the work presented here is not primarily concerned with social questions, but with the algorithms required to bridge from the raw data to the other phases of the cycle, especially between the data gathering, modeling, and simulation steps. The complexity of social interaction and human behavior in general requires that sufficient sophistication be encoded in the models used to explain that behavior. In fact, it is possible that a major reason that social science has not yet proceeded down this path is that model-fitting has not advanced much beyond basic regression. Therefore, it is up to computer scientists and others working at the intersection of social science and machine learning (also known as computational social science) to bring these tools to fruition by showing how data mining can guide the development of empirically based reasoning agents. In effect, my proposed framework presents a structured way for constructing behavioral features which can then be input into the supervised learning method of

choice. This technique is just a form of feature selection, analogous to those commonplace in supervised learning, that uses a reasoning process rooted in the relevant context to identify and define these helpful features. A successfully abstracted model will then create the space for novel theories incorporating heterogeneous agent populations. This document, therefore, aims to address the following central question: how can we predict how strategic agents make decisions?

1.2 Thesis Statement and Dissertation Overview

Thesis: Assuming that agents engaged in repeated strategic interaction undertake a reasoning or learning process, the behavior resulting from this process can be characterized by two primary factors: time-horizon of planning and depth of reasoning over base rules, in particular the primitive strategies of repeating, imitating, and randomizing. Values for these two factors, along with the relative prominence of the base components, can be learned effectively from interaction or observation and the resulting predictive model is transferable to new payoff functions, producing highly effective strategic responses.

The problem considered below is how to predict the future behavior of individuals and populations given historical observations of their behavior. At first glance, this task appears monumental. After all, there are an infinite number of potential strategies a player can use. In a repeated game, this fact takes on even greater import because it is true even if limited to equilibrium strategies [Aumann and Maschler, 1995]. To address this issue, I will abstract the decision-making problem into a reasoning structure with a small number of parameters, which I will argue can adequately represent agents in these limited domains and therefore generalize to a variety of related but different game

scenarios. Some qualities of the strategic decisions faced vary from game to game, such as available actions, number of opponents, payoff functions, and even whether the payoff is known to a player. However, there are enough significant common features that we can propose a starting point for designing solutions. For instance, one of the most powerful ways to achieve an abstracted model is to focus on reasoning capacity. The case studies explored throughout this dissertation show how to apply this mindset to reach specific solutions to prediction-focused problems, formalized through a unified approach. The other important feature, time horizon, captures the tension between short-run and long-run performance, and behavior can significantly depend on how an agent balances these considerations.

To illustrate what is meant by an abstracted model, consider an agent attempting to select a decision in a single turn of poker: either to call a bet, raise the bet, or fold the hand. Regardless of the inner workings of the agent, we can characterize its behavior over many hands as a distribution over these three actions, ignoring the actual cards held by the player. The implicit assumption of this very simple model is that two agents who happen to behave according to an identical distribution are functionally equivalent.

One obvious weakness in such a naive learning mechanism is that it ignores the case-specific factors that went into each decision. Notice that if we take the particular situation into account, it is possible to analyze an agent's behavior from a social-reasoning standpoint—a computer scientist's way of saying you should play the player, not the hand. A player who understands the game but has no reasoning model of an opponent might simply wait for the best hand possible to maximize chances of winning, a tight risk-averse strategy known as the rock or nit. Obviously, this type of strategy is easy to predict and exploit,

because the bet itself reveals the cards in hand. To profit from a nit (or constant bluffer, or other simple type), one requires the ability to identify these types and respond appropriately. That is, each bet or fold can be projected on some scale of aggressiveness or caution, as well as degree of bluffing.

Over many hands, an observer would witness some distribution over these axes, leading to an observation inconsistent with the action-distribution model: agents who reason the same amount will behave identically, all else being equal. However, in the first case, a learned distribution (over actions) cannot be reasonably transferred to new hands or games, whereas the second model (distribution over reasoning) can. More details about this model will be formally introduced in Chapter 2. It should be noted that while there have been some major advances in building poker bots who can do very well in one-on-one games, progress has been slow in games with multiple opponents [Littman and Zinkevich, 2007]. A meta-reasoning model could be required to make substantial improvements in this space moving forward.

Another important factor for making decisions is their time scale for reward collection. Certainly, there are many situations where having a high value for future rewards would cause an agent to take one course, whereas a low value of the future would lead to a different choice, focused on short-term effects. In psychology, a series of experimental studies have noted that children who can delay eating one marshmallow in order to receive two after waiting half an hour are more successful later in life [Mischel et al., 1972]. Looking at this question through our poker example, we could say that a strong bluff that would work for the current weak hand causes harm in the long run, as it may increase the likelihood of bluffs being recognized and called in the future. In a game setting, we will find that the time horizon or discount factor is a key

component in strategic behavior, especially in standoffs or social dilemmas.

A concept related to behavioral modeling is multiagent learning, when an opponent has the ability to adapt its strategy with experience. Agents who adapt their decisions are sometimes not predictable solely by previous behavior, and so the model has to be extended to dynamically incorporate a relationship between reward and behavior. At a more fundamental level, the optimal strategy against a learning opponent can include an element of teaching. Teaching is itself a form of reasoning, as it requires the teacher to respond to an agent that is also responding, albeit in a dynamic way over time. A teaching strategy can be designed to manipulate a learning opponent into outcomes better for the teaching agent, and therefore this concept is an important building block for multiagent modeling.

Before discussing the concrete experiments, I will first address the methodology for supporting my thesis. The meta-learning/meta-reasoning population model I introduce in Chapter 2 can be thought of as machine learning over best response features for the purpose of predicting behavior. Setting up the problem this way distinguishes this model from an agent that responds from statistical knowledge assuming a fixed world without these behavioral features. That is, we can make a distinction between a population with some internal reasoning process, and a natural or mechanistic system that simply outputs signals. Starting from base heuristics (random, repeat, imitate, etc.), the reasoning agent can be expected to follow a procedure for doing well in response to others, but varying the amount of reasoning and time scale will produce different actions.

Once these features have been extracted, a modeler can then use a supervised learning algorithm of choice to fit the relevant parameters. In order to

show how including new features allows for considerable transfer between similar but non-identical games, we need to compare the predictions of a meta-learner with those of a naive learner in these types of situations. A helpful analogy is that evaluating the transferability of an agent model to a new game is like testing a trained statistical model on an out-of-sample data set. In both cases, the new data has not been used to build the model, so we are talking about mitigating the risks of over-fitting.

This question of what constitutes an opportunity for knowledge transfer is not a simple one. For example, it is clear that our confidence in the applicability of a model trained on a certain agent will be diminished when the rules of the game change substantially, even other game properties remain the same, like the number of actions. What about simply changing the payoffs? The transferability of models over games depends on more subtle concepts like structure and complexity. That is, if there is only one action that gives a positive reward and the rest are always zero, an agent faces an easier decision than when there exists a dilemma or conflict between long-run and short-run rewards. Because I am making claims in this thesis about the decision-making capacity of agents, in terms of their depth of reasoning or time horizon, it is important to remember that issues that alter the computational costs of the decision will affect a model of an agent. This observation would call for some way to measure the relationship between complexity and the resulting model, but the field has not yet reached that point.

While the purposes of prediction may differ in the experimental studies shown in this work, the value and goal of accurate prediction remains. Performance in many games strongly depends on how well a player can predict an opponent's behavior. Indeed, the perspective of an individual agent in one

of these games and an analyzer of a social data set is the same with respect to prediction.

This document will explore a diverse range of domains where individuals benefit from acting strategically, and can make decisions by building implicit or explicit models of their neighbors or opponents. I will focus on the repeated setting, where players have the opportunity to adapt to others' behavior over time. Games with a small action space can lead to greater confidence that the strategic reasoning space has been sufficiently covered and the reasoning process can be shown in detail. Even so, these games exist at the cusp of a complexity explosion, which demonstrates that computational problems arise even in small games whenever multiple agents are present.

The following case studies include games with two players, three players, and a population of N players, but common features, such as bounded reasoning, are present in all of these multiagent settings. A diversity of evaluation methods strengthens the central claim by showing how such limits to reasoning and adaptation are universal in an empirical sense. The first two experimental domains focus on predicting the behavior of artificially intelligent computer agents and algorithms, while the final one investigates human behavior. In the software agent setting, we can ask how agents based on a model informed by meta-reasoning perform when placed in the given setting. The agent-construction process may be less of a concern in the human cases because we are not as interested in outperforming as much as explaining through prediction. However, we would like to show how population models can be applied in each of these diverse settings (human versus computer, two players versus many) to demonstrate their generality. The experiments are designed to

apply the meta-learning and meta-reasoning processes generally across a variety of situations, but in some cases the settings require some custom changes resulting from structural differences.

1.3 Experimental Case Studies

Throughout the work, I seek to answer the following questions:

- How do the abstract parameters of learning algorithms employed by agents alter their behavioral signatures?
- Can we apply a meta-reasoning model built from knowledge about opponents to predict behavior in new, unseen games?
- Does a meta-reasoning model replicate population behavior?

These questions are meant to investigate how to build behavioral models that can transfer to new situations. We can evaluate these models by how well they predict individuals, or if that information is unfeasible or inaccessible, how well they predict populations. To answer these questions, I will explore three experimental setups.

The first considers how to build models of learning algorithms in simple games, which can be classified as a machine vs. machine competition. The goal for this experiment, detailed in Chapters 3 and 4, is to construct an agent that outcompetes the other participants in a tournament by building predictive models. The second case study involves the Lemonade Stand Game tournament, where computer agents are designed by researchers to play against each other in a competitive scenario. One significant twist in this game, as we will see in Chapter 5, is that the game payoffs are shifted from match to match, so

that model generalizability yields a major advantage. In Chapter 6, the third set of experiments focuses on human experiments in the public goods game, where models are trained on people vs. people interactions with the aim of reproducing the behavior of the entire population. This type of goal is a relatively unexplored one in these settings, but is important if we would like our model to capture group dynamics.

1.3.1 Learning Algorithms in Two-player Normal-form Games

Question 1. *How do the abstract parameters of learning algorithms employed by agents alter their behavioral signatures?*

A single opponent can be modeled given a historical sequence and the right hypothesis space of algorithms. Chapter 2 introduces definitions, background, and related work in game theory, multiagent learning, and prior opponent modeling algorithms as well as formally presenting the proposed framework.

The various dynamical systems created by learning/reasoning algorithms in a game have a small number of parameters in a handful of possible adaptive rules. Chapter 3 presents several multiagent learning algorithms and investigates in depth the unique properties of one called ϵ-greedy Q-learning. Chapter 4 explores how a meta-reasoner can best model the algorithm used by an agent in a simple two-action game. Along the way, we discover how differently structured learners show radically divergent behavior in certain famous games like prisoner's dilemma.

To be consistent with the overall thesis, I frame the problem as one of how to use observed data to predict future behaviors, given that the player is learning new strategies. This component can be a critical one in repeated games because intelligent agents are usually not stationary. Since non-stationarity breaks the

assumptions of traditional learning algorithms, the challenge is therefore to construct a model of the learning itself and then to fit that model to data, in a process defined as meta-learning. Although this document focuses on the simplest games, the techniques used can inform an approach for richer domains.

Features for Modeling Learning Algorithms

Consistent with the central thesis, Chapter 4 will show how a meta-reasoner can correctly model the actions output by a learning algorithm with a combination of a set of base strategies and the optimal discounted responses to its opponent. The process is based on the idea that learners will converge to an optimal strategy given enough state representation, in this case knowing the actions of each player in the previous round. The problem at hand is to identify opponents who are responsive to cooperation, and by extension the optimal amount of cooperation to offer. Answering this question requires a modeler to identify the amount and type of learning displayed by the other player, regardless of the particular algorithm employed.

In some cases where convergence is not assured but a payoff higher than minimax is possible, the learner exhibits ongoing dynamic behavior and so the model requires a way to capture the relationship between the opponent reward and strategy. This extra step can be accomplished by extending the features of the default meta-reasoning model to be weighted by the average reward received. In each of these cases, the modeler's optimal strategy can be best described as having the aim of entraining or teaching its adaptive opponent. There exist special cases where the behavior output by a learner will substantially differ given a new set of input observations. Fortunately, the meta-reasoning model can be extended to adapt itself in the face of different levels

of rewards, so that the change in model is a linear function of the deviation of reward.

1.3.2 Reasoning Models in the Lemonade Stand Game Tournaments

Question 2. *Can we apply a meta-reasoning model built from knowledge about opponents to predict behavior in new, unseen games?*

In Chapter 5, I focus on evaluating the proposed meta-reasoning model in an agent-tournament setting where the payoff rules are never exactly the same and is based on original work that has been published [Wunder et al., 2011, 2012]. The central claim remains: that a strategic hierarchy exists, can be learned automatically, and applied effectively to predict behavior in a general way. Furthermore, to abstract the decision-making process, we can use the same parameters identified by researchers in simple human experiments. To validate this approach, I undertake a dual-pronged attack.

The first method of evaluation depends on successful classification of individual agents, and measuring how well they do against a specific population of strategies. Essentially, this line of inquiry is concerned with proving the soundness of the resulting strategic hierarchy. If an agent identified as using a certain depth of reasoning does well against others that it should be optimized against (because they were identified as such), then this result would provide evidence that the reasoning model is well-formed. If it does not, then either some of those agents were mis-classified or something is wrong with the hypothesized reasoning model. An even stronger piece of evidence would show that more sophisticated agents (again, as defined by the model learner)

perform better on average, although there are exceptions to this rule in some circumstances. The second, and more important, mode of evaluation involves strategy design, in which a computer agent is built after or during the learning of the model. Evidence for the claim therefore rests on performance of the resulting strategy in a competition of players, because if it does poorly (or worse than a competing algorithm) we cannot say the model adds any value.

Many researchers have focused on the problem of building models of single opponents. Chapter 5 explores four separate tournaments of a novel multiagent environment called the Lemonade Stand Game (LSG), in which computer agents competed to maximize payoffs in a three-player game. When a game has three players, the challenge deepens as the number of pairwise interactions grows quadratically with N, the number of agents. In addition, when game payoffs shift from game to game, high-level knowledge encoded in the reasoning strategy must be transferred to new situations. The tournaments in the succeeding years gradually grew more complex, moving from a completely symmetric setting to one that changed every new match, where players could remember history over time. The formal model presented in Chapter 2 handles these types of challenges very well and an agent applying the lessons of the model won the last two tournaments.

In Chapter 5 of the dissertation I characterize unknown, real-life agents according to the two factors and three base heuristics. The LSG provides a simple and yet rich environment to test the predictions made by a meta-reasoning population model, as well as the sophisticated strategies created by the model. Prior approaches tend to ignore or downplay the strategic decision-making capacities of rival agents, and yet the field simply lacks a better algorithm to compare against. In games with different payoffs, they are not appropriate because

the old knowledge is obsolete unless projected into a reasoning space like the one described earlier. We show that a meta-reasoning model has a significant advantage when predicting the behavior of groups of reasoners.

Features for Learning a Level-based Discounted Model

This section explores how to use meta-reasoning as described formally in the next chapter to solve this concrete learning problem. In a game where players aim to stake out good actions in early rounds, successful strategies need to both send a teaching signal and tailor this signal to have maximum effect by inferring who will be receiving it. Reasoning plays a critical role when there is a limited observation window, and a complex action space forces agents to do some limited computation, but agent sophistication is necessarily unequal. A simulated environment allows for estimation of the distribution over the depth of reasoning that individual agents, or the whole population of agents, perform. The relationship between discount rates and regret are also a central focus as location games create exactly the type of standoffs where the players' relative future action values become an important consideration. Meta-reasoning therefore prescribes doing long-term planning with the parameters learned from prior interactions. This planning amounts to a simulation of the coming game before it takes place, so that the agent can compute long-term values of the first action in the game.

Experimental Results

Keeping with the overall evaluation procedure, the experiments emphasize tournament performance. While inter-game learning was only allowed in one of the competitions, this limitation gives a unique opportunity to see how well

the meta-reasoning framework performs with regard to reasoning without the threat of agents changing between games. It turns out that even in the competition where learning was allowed, only our Rutgers agent was successful in building a complete model and using it to plan. This outcome further simplifies our task because it means that the strategies are mostly stationary over many games. To evaluate the proposed method, an agent generated from the meta-reasoning algorithm will be tested against all the agents from each of the tournaments. A meta-reasoner should do well against each constituent population.

1.3.3 Populations of Heterogeneous Strategies in Public Goods Games

Question 3. *Does a meta-reasoning model replicate population behavior?*

Prior sections deal with small populations of two or three opponents, which can be modeled individually with a small number of relations between participants. Chapter 6 poses the question of how to build models when the population has larger number of agents, where an exploding number of pairwise interactions creates large computational challenges as well as data sparsity. As such, we will represent the information each player uses to build a strategy in a compressed form, where only average or total actions are available, thereby making the assumption that the joint opponents act as one big opponent. We note that this assumption is not always valid but depends on the nature of the payoff function and whether it incorporates group actions as a whole.

Specifically, for this chapter I provide an analysis of a series of experiments

conducted on Amazon Mechanical Turk where a medium-sized group of participants interact via a social network topology to play the public goods game, a ten-round repeated game that has been documented in a recent paper [Wunder et al., 2013]. Each player has five neighbors. The overarching goal of behavior prediction is still present, although here the behavior of populations in the aggregate becomes more salient in contrast to small numbers of opponents. As such, the emphasis will be on predicting population-wide distributions of behavior, in terms of the target output. In some cases, heterogeneous models will help achieve this goal. The available data contains records from some individuals over many matches and in several different experimental treatments. This data allows for analysis of short-term reasoning (within the ten rounds) and long-run learning as a result of experience.

The strategies employed by people in medium-sized groups (5-30) can resemble reactive rules when an individual's action does not have a major impact on the utility of others (and vice versa). Reasoning about others can be difficult in medium-sized groups due to the large number of interactions, which can cause a reliance on heuristics to decide behavior. People often stay tied to their own prior behavior as a predictive variable, but also respond weakly to group behavior, although the effects strengthen if others are acting in an exploitative way. In other words, we might ask how players are characterized among the base strategy set and how strategy selection changes over many trials. There is some evidence that players do some forward planning under the expectation that their behavior has some effect on neighbors. In addition, the social nature of the public goods payoffs leads to reciprocation reducing to reciprocation, so that higher-level strategies closely resemble the base strategy class. Therefore,

I claim that within the reasoning structure I am exploring, players exhibit behavior consistent with the base strategy set with some forward-thinking. The support for this claim is found by using the model to replicate the behavior of groups. Forward-looking reasoning, where it exists, is expressed by checking how sensitive players are to future payoffs, that is, how much does behavior correspond to maximization of a future discounted investment function.

A common observation is that people adopt strategies across a range of types, showing a great deal of diversity. Given an experimental data set, we can use the techniques discussed here to learn the distribution over the various types and reconstruct an artificial population that behaves very closely to the original. This chapter will investigate how the multiagent cycle informs a predictive model of group behavior in a large set of public goods games in online experiments. For training data, I use either all players in all games (minus one for testing) to build a homogeneous model or a fraction of a single player's games in the heterogeneous model. In many cases, the simplest models are superior to more complex ones, and are therefore good candidates for a population model for an attempt at replication.

Features for Predictive Models

There are two main approaches of using models to predict behavior. One is to evaluate predictors of individual behavior, and another is to test these predictors in aggregate to evaluate predictions of group behavior. Within these goals we can subdivide the models further into a single homogeneous model that attempts to explain all individuals, and a heterogeneous model that fits each individual specifically. I propose and test a diverse variety of predictors and report how they perform. These predictors represent strategies across the

reasoning hierarchy, previous models proposed by researchers in the field, and other non-behavioral methods from the supervised learning space that offer a comparison without much explanatory value. Other researchers have proposed that players behave according to a set of types, such as free-rider, altruist, or conditional cooperator [Fischbacher et al., 2001]. My investigation extends this concept to a spectrum of strategies that assign some amount of each type to agent behavior. In particular, I used a regression model with various simple features as the base strategy, such as previous round action and previous average neighbor action. For a forward-looking reasoner, I add in the discount factor for estimating future returns on current contribution. The shape of this decision function turns out to be player-specific.

Building an Empirically-based Computational Social Model

Once we have a distribution over the parameters of a simple predictive model, we can truly test it by assembling artificial agents who execute the model. Evaluation consists of measuring the difference between the population's simulated behaviors compared to the actual ones seen in reality. Because the simulation process outputs a distribution over actions over many games, a method for comparing marginal distributions is necessary, and in this case Kullback-Leibler divergence suffices.

Empirically Informed Simulations in Networks

With a substantiated model, we can turn to questions that cannot be answered with human data due to time, cost, and feasibility constraints. For example, we can test how our human-agent models behave in various network topologies, with arbitrary numbers of nodes, through simulating a network populated by

the agent models fitted from the data. This type of experimentation, known in the literature as agent-based modeling, has gradually become more appealing to researchers in a variety of fields, especially in social sciences but also natural ones like ecology. However, in many cases these modeling approaches lack empirical support, and the results amount to little more than thought experiments that depend heavily on implementation details. On the other hand, when the model is built from actual human data, it assumes a more meaningful and powerful form. Because the model has been shown to replicate these behaviors, the conclusions drawn from further simulations are more likely to have real-world validity.

1.4 Why Multiagent Learning?

Before examining these domains and their corresponding algorithms, it makes sense to ask why we should study them and where this work is situated in the literature. After all, the game theorists would have us believe that the equilibrium framework is the end of the story, with no need for learning because all participants will either have "solved" the game, or will lose to players who have.

Shoham et al. 2007 addressed this issue by noting that there are many real-life examples where an equilibrium analysis is invalid. The authors go on to identify five main agendas in the field of multiagent learning, which are often conflated or left implicit. They are:

- Computational
- Normative
- Prescriptive, cooperative

- Prescriptive, non-cooperative
- Descriptive (Predictive)

These goals are listed in descending order of theoretical value, and ascending order of practical application. The computational agenda views learning algorithms as a way to compute some property of a game, such as an equilibrium or its corresponding strategy. They are used as a substitute for a search algorithm to arrive at the same goal through a series of interactions.

A normative analysis would seek to find properties of learning algorithms, such as which types of learners are in equilibrium with each other. Perhaps a contrasting question would be which algorithms are not in equilibrium with themselves, when this outcome might otherwise be expected. Chapter 3 on learning algorithms will provide some analysis from this viewpoint.

The two prescriptive approaches come from the standpoint of designing agents to interact in a multiagent environment. In the first one, cooperation or coordination of a population of decentralized agents is required to achieve some high-level goal, like a team of robotic soccer players in the Robocup tournament. As computerized agents have proliferated in virtual and physical space, adequate mechanisms for designing local protocols have become a tantalizing prospect. The likely advent of self-driving cars in the near future is one large-scale application of this agenda. A good or best equilibrium is certainly desired by traffic engineers, but finding it is not the central problem.

In non-cooperative settings, the agent-design goal acquires a different focus, namely high performance when interests of competing agents are non-aligned or in direct opposition. Equilibria are not a primary consideration for problems of this nature. This context is perhaps the closest to traditional AI questions of how to build an effective strategy for a given environment. The

main difference, of course, is that additional agents add instability, complexity, and an external source of intentionality. It is perhaps this reason that serious progress in the field has remained elusive. That is, building agents with social components is inherently difficult because the number and complexity of the required models are potentially unbounded.

Finally, the descriptive/predictive agenda aims to construct learning models that represent natural agents, such as people, as they interact in social environments. As a slight change to the original list, I have added prediction as a supplementary goal for two reasons. First, we may want to describe a set of interacting agents because we wish to predict their behavior, say as the result of some change to the game rules or state. Second, a full descriptor of an intelligent agent should be able to replicate its behavior under a set of circumstances, and should therefore be able to predict the actions of that agent under those same circumstances. If a model of a learning agent is not successful in predicting the actual behavior better than a random or other baseline predictor, we cannot conclude that it has captured that behavior.

This dissertation focuses primarily on the fourth and fifth agendas of this list, with some attention paid to the second. Another way of explaining this dissertation's contribution is that the framework works through prediction/description (Agenda 5) of software or human agents, where the goal is to create a competitive agent by using the model (Agenda 4). Therefore, the proposed framework acts as a bridge between several of these agendas, as well as between radically different domains.

The next chapter lays out the formal framework and definitions used throughout the rest of the dissertation.

Chapter 2

A Framework for Transferable Strategic Meta-reasoning

Here, I describe the major components of a model of strategic intelligence. First, however, let us review game-theory background as well as the foundational assumptions generally accepted in the fields of economics and traditional game theory where decision makers are concerned. The study of multiple self-interested agents was formally introduced by Von Neumann and Morganstern 1944 and is the basis for the models investigated in this thesis. In any interaction between two or more individuals, we define the *population P* as the set consisting of all participants playing the game. The nature of this population will be of particular interest later on. Many of the following definitions and concepts are common knowledge in the field of game theory, but the forms used here are from the book *Multiagent Systems* [Leyton-Brown and Shoham, 2009] unless otherwise stated.

Definition 1. *Define a game of strategy* Γ *as* $\Gamma = \langle P, A, S, U \rangle$ *consisting of:*

- *P: Population of N players indexed by i*
- A_i*: Set of M actions available to the player i*
- *S: Set of states of the world*
- $U_i(S, A)$*: Utility function for i,* $\mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$*.*

For shorthand, we may speak of joint actions to capture the population behavior: $a = (a_1, a_2, ...a_N)$ where agent j's action is $a_j \forall j \in P$ is an instance of a joint action while $A = A_1 \times A_2 \times ... \times A_N$ is the joint action space. Faced with joint action a, player i sees opponent joint action a_{-i}. For the time being, let's assume *complete information*: each player knows this space as well as all utility functions of the population. Players also choose actions at the same time. Another useful concept is that of a strategy, which essentially means a distribution over actions, where $\pi_i(s, a_i)$ is the probability that agent i chooses action a_i in state s. If there exists an action a_i for which $0 < \pi_i(s, a_i) < 1$, then we define π_i as a *mixed strategy*, and otherwise it is *pure*. Analogously, there is an opponent strategy, π_{-i}, which may also be mixed.

2.1 Equilibrium

The bedrock concept of game theory is that of equilibrium [Nash, 1951]. In the world of individual game players, this concept, called the Nash equilibrium after its inventor, John Nash, has been inseparable from game theory ever since it was first developed. A population in a game will be in equilibrium if each player is playing a *stable strategy*, where there is no incentive to unilaterally change to a new strategy. To help expand the notion of a stable strategy, it is necessary to introduce the idea of a best response B, which is a strategy that produces the highest utility given an opponent strategy.

Definition 2. *(Best response) A* best response *of player i to joint strategy π_{-i} is a strategy $B(\pi_{-i}) = argmax_{\hat{\pi}_i} u_i(\hat{\pi}_i, \pi_{-i})$.*

Note that a best response may consist of a mixed strategy. If that occurs, it means that the actions with positive probability have equal payoffs under the

current joint strategy.

The difference in utility between any strategy and the best response is known as its *regret*.

Definition 3. *(Regret) The* regret *of player i for playing strategy π_i against joint strategy π_{-i} is the value $\rho_i = \max_{\hat{\pi}_i} u(\hat{\pi}_i, \pi_{-i}) - u(\pi_i, \pi_{-i})$.*

Finally, combining the definitions from above, we reach the definition of Nash equilibrium, discovered by Nash as a solution for a given game.

Definition 4. *(Nash equilibrium) A* Nash equilibrium *is a joint strategy π^* at which no player has the incentive to change its strategy (unilaterally), so that $\forall i \ \max_{\hat{a}_i} u_i(\hat{a}_i, \pi^*_{-i}) \leq u(\pi^*_i, \pi^*_{-i})$.*

In other words, the regret for all players in a Nash equilibium is zero.

One more definition regards an action that always (or never) produces a payoff that is greater than some other action, keeping opponents fixed.

Definition 5. *(Dominant action) Action a^** dominates *action a' if $\forall a_{-i}, u_i(a^*, a_{-i}) \geq u_i(a', a_{-i})$, and $\exists a_{-i}$ where $u_i(a^*, a_{-i}) > u_i(a', a_{-i})$. We call action a^** dominant *and a'* dominated.

In a Nash equilibrium, no player plays a dominated action. Conversely, a rational player should always play a dominant action if it is the only one among all other actions. Connecting these two concepts more deeply, a bedrock assumption of game theory is that perfectly rational players should only play a Nash strategy, resulting in a Nash equilibrium. If players do not reach an equilibrium, than at least one player is not rational.

For a population of two or more players to be guaranteed to exist in equilibrium, two assumptions must hold. One is that all players are fully *rational* in

the sense that they want to maximize their profit and have the knowledge and capacity to do so. The second is that they know all other players have the same degree and powers of rationality, the so-called *common knowledge* assumption. Implied in this arrangement is that if at any point in time the players had less than perfect knowledge, they will learn completely and instantaneously everything they did not know, due to the underlying profit motive. The result of these assumptions is that players essentially perform an infinite amount of reasoning in constant or zero time, and are able to solve any problem or completely optimize their strategies. Furthermore, the players do not incur any additional cost for this optimization. The result of this process is that the profit maximization objective is attained for all, by which point the population achieves a Nash equilibrium.

As a theoretical tool, the advances of this framework certainly have the benefit of focusing researchers on relevant properties and possible stable outcomes of strategies and games. However, as a tool for explaining and investigating large social systems, the assumptions as stated fail to uphold many long-held practices of the scientific traditions and practices economists have sought to emulate. Specifically, the theoretical models do not match empirical reality. Indeed, social groups tend to exhibit as much chaos as stability. In addition, from a computational standpoint, the problem of finding a Nash equilibrium has been proven to be intractable for large action spaces or as N grows [Daskalakis et al., 2006]. This problem is multiplied if we are interested in repeated situations, which in practice are much more common than one-off contests. As detailed by many economists [Aumann and Maschler, 1995], there can be infinitely many equilibria between two players in an infinitely repeated general-sum game, even when the one-shot version has only one. This situation raises

the issue of how to craft a strategy so that the final equilibrium point is beneficial in terms of *average reward*, creating a meta-game in the learning/teaching space.

Definition 6. *(Average reward) Given an infinite sequence of payoffs $u_{i,1}, u_{i,2}, \ldots$ for player i, the* average reward *of i is*

$$\bar{u}_i = \lim_{T \to \infty} \frac{\sum_{t=1}^{T} u_{i,t}}{T}.$$

Another important concept is that of *future discounted reward*, which incorporates the idea that players place a higher priority on rewards received in the near term than in the long term.

Definition 7. *(Future discounted reward) Given an infinite sequence of payoffs $u_{i,1}, u_{i,2}, \ldots$ for player i and a discount factor γ where $0 \leq \gamma \leq 1$, the* future discounted reward *of i is*

$$U_i = \sum_{t=1}^{\infty} \gamma^t u_{i,t}.$$

In repeated games, players must take into account the future effects of their actions. Because of this central fact, some strategies can be enforced even if they lead to short term payoffs that are suboptimal.

Definition 8. *(Enforceable reward) An average reward $\bar{u}_i$ is* enforceable *if $\bar{u}_i \geq v_i$, where v_i is player i's minimax value, the least amount the player can guarantee if other players adopt minimax strategies against i so that $v_i = \min_{a_{-i} \in A_{-i}} \max_{a_i \in A_i} u_i(a_{-i}, a_i)$.*

Definition 9. *(Feasible reward) An average reward $\bar{u}_i$ is* feasible *if there exist rational, nonnegative values β_a such that we can express $\bar{u}_i$ as $\sum_a \beta_a u_i(a)$, with $\sum_a \beta_a = 1$.*

These definitions allow us to state a powerful theorem called the *Folk Theorem*, which captures the range of possible values that can be achieved in the presence of rational opponents.

Definition 10. *(Folk Theorem) Given a normal-form game G and any average reward $\bar{u}$:*

1. If $\bar{u}$ is the expected reward for any Nash equilibrium of G, then $\bar{u}$ is enforceable.

2. If $\bar{u}$ is both feasible and enforceable, then $\bar{u}$ is the average reward for some Nash equilibrium of the infinitely repeated G.

The Folk Theorem allows for a wide variety of repeated strategies that can result in average rewards that are higher than any single-shot Nash equilibrium of a particular game. A simple enforcement mechanism is as follows. The players of the game take actions consistent with the enforceable payoff. If at any point some player deviates from the prescribed action, then all players of the game resort to their minimax strategies against that player. For sufficiently high values of γ in the future discounting case, it will be optimal to accept the enforced strategy rather than attempting a different one.

The famous game of Chicken (see Table 2.1, explored in detail later) illustrates the conflict between a rational desire to minimize our own regret on one hand, and expecting the opponent to be rational in response to our teaching stance on the other. This direction of inquiry has led to work in several fields [Littman and Stone, 2003; Press and Dyson, 2012], but it remains a challenging question to decide whether to hold fast or give in. Often it is noted that a "theory of mind" can be a necessary component to constructing a rational response to learning opponents, and I will be addressing this issue shortly. As such, I will use the term meta-learner or meta-reasoner to refer to agents that do some opponent modeling of the reasoning process and also use observation data to inform their model.

Definition 11. *A* meta-reasoner *or* meta-learner *is an agent implementing a learning process in which it explicitly models another agent or agents according to their*

reasoning or learning mechanism, provided experience of their behavior.

A more nuanced way of understanding rationality is to define it as a dimension of agent behavior that can be high or low depending on the agent taking the action, instead of viewing it as an absolute feature fixed at an extreme point. To incorporate bounded rational strategies, empirically speaking, researchers in the field should ask how rational a population of players is, which decision rules they use, and so on, instead of taking these qualities as given. This agenda is more in line with the goals of artificial intelligence, going back to the writings of Nobel-prize winning economist Herb Simon 1982.

A supporting concept in economic theory is the widespread use of a representative agent, which goes back to the earliest days of mathematical economic modeling [Marshall, 1890]. Essentially, to simplify analysis, the behavior of all agents in the world is assumed to be represented by a single agent with a standardized set of beliefs, preferences, payoffs, etc. Even many laboratory experimenters in this space fall back to the temptation of defining the population according to the representative agent, thereby ignoring the heterogeneity of human behavior. To address this shortcoming, theorists and behavioral researchers alike need richer models of reasoning to make significant progress in long-term, sequential environments, and advanced algorithms are required to implement them.

Ironically, economists of all stripes have persisted in a strong, narrow definition of rationality without developing such methods, which advanced reasoners would presumably have access to in reality. In addition, models of phenomena should have an empirical basis, even as they attempt to generalize strategies across a population. Recent years have witnessed the rise of the fields of algorithmic and behavioral game theory, and computer scientists

have been at the forefront of efforts to build predictive models of people. On the software side, there has been an emergence of practical approaches like *empirical game models* [Jordan et al., 2010], which explicitly include simulations as a necessary step in the problem-solving process. Empirical games typically represent population behavior as an aggregated *strategy profile*, and the model-construction process may even use game-reduction methods for greater speedup. These simulations aim to discover equilibria among the provided strategy set by repeatedly selecting the best agent strategies from games among subsets.

This document aims to take such approaches one step further, towards the aim of building profiles at the level of strategic sophistication. At a high-level, this approach is focused on the problem of constructing and learning such models of collective reasoning, and evaluating them for the purpose of simulating and predicting group behaviors. The empirical view of multiagent learning parallels the discipline of machine learning: projecting a set of samples into a low-dimensional space for the purpose of predicting new samples. In our case, however, the space consists of the result of applying reasoning or learning algorithms.

2.2 Level-k Reasoning and Non-reasoning Strategies in Repeated Domains

The main alternative to the unlimited rationality approach is a finite reasoner, and the most common version is one that iteratively applies steps of reasoning, in the form of best responses, to some belief over strategies, starting at a base set of strategies. The idea of iterated reasoning rests upon two tenets: that a

step of reasoning is well-defined and that a non-reasoning strategy exists. For our purposes, *strategic reasoning* is the act of making decisions in the context of the actions that another agent is believed to take. A *step of reasoning* is the direct payoff-maximizing response to some set of these beliefs—formally, a single application of optimization rule $B(\pi_{-i})$. Recursive application of best response, also known as iterative best response (IBR), will output a *cognitive hierarchy* of strategies. The first step, or *level*, up the hierarchy would in essence be a direct response to a belief that the population is composed of non-reasoning agents, known as the *base strategies*.

Definition 12. *(Base strategies) A set of* base strategies Σ *consists of the probabilistic strategies that can be derived without knowledge of utility function.*

The second step requires the formation of a new set of beliefs composed of the strategy derived from a step of reasoning over the first set. This choice of non-reasoners is therefore a crucial determinant of the ultimate hierarchy. Without these beliefs, there exists no ground for reasoning to stand upon, and it is impossible for optimization to take place.

A non-reasoning strategy is broadly defined as one that does not use payoffs to make a decision, potentially covering a wide range of behaviors. It is important to note that a non-reasoning strategy, usually consisting of a fixed mapping over actions, can depend on the type of game it exists within. Reasoning requires a belief about opponents, and the first step starts by considering simple strategies that represent basic reactions and do not optimize performance. It should be mentioned that most of the literature in this space considers single-round games, where it is not necessary for base strategies to include multi-round play. However, it is impossible to rule out defining non-reasoning play that somehow depends on prior action, since we could always define the

strategy as one that selects a single action for the whole game. We will primarily consider three action primitives to form a base strategy, which can be combined together in various ways to form a mixed type: *uniform*, *previous self*, and *previous other*.

A typical choice in one-round games is uniform randomness: $\forall a_j \ \pi(a_j) = \frac{1}{|A_j|}$. In one-shot games, where a single action is played and then the game ends, uniform random action is a simple and unbiased way to cover the range of actions, guaranteeing that any strategy responding to it must outperform this unsophisticated base case. Uniform randomness neatly represents a completely uninformed player, who might not understand or care about the rules and payoffs, and is therefore equally likely to choose any action. Some other choice of base strategy, such as only choosing action a_1 so that $\pi(a_1) = 1$, could lead to flawed conclusions in some cases, like if all actions have equal utility against it. This situation might have the undesirable result that a dominated action is played in response by a reasoner, which could be the worst possible choice. A uniform opponent will at the very least rule out that outcome, because there will be another action that performs better than the dominated one against a uniform distribution over opponent actions (by definition of dominated action). Uniform randomness has another advantage in that a player who optimizes against it in a symmetric game will do at least as well as a random player would, which is important because otherwise she should choose an action at random.

However, if a game is repeated multiple times, past action can be a determinant of future action, even without allowing for payoff maximization or agent reasoning. The well-known psychological effect of anchoring or priming creates a tendency to make decisions in the context of what came before [Tversky

and Kahneman, 1974]. In practice, this phenomenon leads to repeated actions, so that a fixed, unchanged action is an important part of the base strategy set. From the perspective of our hypothetical reasoner, a fixed opponent mimics an environment with a stable state, allowing for a less strategic but still potent response. In fact, these two basic strategic components, fixed and uniform, combine with reasoning elements to cover a wide range of conceivable base strategy sets seen in populations because together they represent the forces of stability and change in a strategic environment.

A final reactive type is to vary play in direct relation to opponent action. This strategy can still be considered non-reasoning because it ignores outcomes, and it represents a mirroring function. Imitation is the most basic of adaptive strategies, as it carries a recognition that a player defers to another's judgment. Conveniently, copying another player's action fits with the idea of reciprocity, which is a core concept in dilemma games. A similar kind of reaction would be to play the opposite of another player or players, which taps into the tendency of people to differentiate from others.

Upon closer examination, we can propose the following theorem:

Theorem 1. *If an agent has the memory of the current round, the only choices for non-reasoning strategies are combinations of randomness, repetition, and imitation, and their complements.*

Proof: The non-reasoning strategy set has no knowledge of utilities (or more precisely is blind to them), so we cannot include a dominant action in the mix. Other possibilities, like arbitrary distributions, are equally likely and therefore amount to uniform randomness. That leaves actions that depend on the known current actions: the non-reasoning agent, or its opponents. The first

case leads to either repetition or anti-repetition, and the second leads to imitation or anti-imitation. We have thus exhausted the possible candidates for non-reasoning strategies given one round of memory. □

We will explore various multiagent settings where the three pillars of constancy, mirroring, and randomness arise again and again, if for no other reason than that there is no simpler strategy that is also symmetric or unbiased over actions. Going further, the above theorem states that if payoffs are not taken into account, these three choices (or their complements) are the *only* options available to a non-reasoning player. Fortunately, they provide a robust and computationally efficient basis for reasoning to begin in many of these environments. An additional justification for this base reasoner definition is that it contains several behavioral qualities that people use in their everyday lives: keeping things steady, imitation, and exploration. In general, we refer to the probability that a non-reasoning strategy *repeats* its previous action as ϕ, the probability of *copying* another's strategy μ, and the probability of *random* action as ϵ. A *probabilistic base strategy* therefore consists of a stochastic mixture of these baseline actions, which we just proved to be complete for next-step blinded response. Representing a base type as a stochastic mix of other types lends a great degree of flexibility when we would like to classify observed behaviors and it is impossible to fit a player into a single box. It also strictly adds generality in the sense that we could represent a pure base strategy as only one of the action primitives simply by setting the relevant probabilities to 0 or 1.

A set of base strategies combined with a reasoning rule leads to a strategic hierarchy.

Definition 13. *(Strategic Hierarchy) A strategic hierarchy H for utility function U is a set of base strategies* Σ*, a sequence of strategies indexed by k corresponding to ordered*

levels, and a rule of assigning beliefs to strategies in level $k-1$ down to the base level 0. A rule is response based but the response can be output with some noise.

We call an agent that performs reasoning within this hierarchy a strategic reasoner.

Definition 14. *(Strategic Reasoner) A strategic reasoner is a high-level function over a strategic hierarchy H, utility function U, and distribution over levels σ, which chooses a level-based strategy $\pi_k \in H$ with probability σ_k.*

There are several design decisions for learning or reasoning agents besides the selection of the base set. One variant of this iterated best response model is the level-k reasoner, which believes that all members of the population do exactly $k-1$ steps of reasoning/optimization. Another option, the cognitive hierarchy model, rests on the assumption that the population consists of some distribution over levels $1...k-1$ and as such is somewhat more robust, for the same informal reason given earlier in regards to the uniform base strategy. Using a distribution over levels, as with actions, helps to optimize higher-order strategies against each of the possible strategies present at lower levels as well as to eliminate blind spots. There is also a great deal of laboratory evidence that these models are a better representation of human decision-making. Ultimately, the success or failure of a given hypothesis class is determined by its ability to generalize empirically to the true population. Whichever form of level computation process is used, the resulting *level-types* are universal, in the informal sense that any strategy can be classified as one of them, with some residual error. The only question is how accurate the resulting classification will be in terms of behavior prediction, an outcome that depends partly on the hypothesis class itself.

When we speak of individuals, it is possible that agents may draw actions from several of these levels, so that we may say they have a certain strategic "center of gravity" but are not limited to a specific level. In this view, player i's depth of reasoning is best represented as a distribution $\vec{\sigma}_i$ where $\sigma_{i,k}$ is the probability that agent i demonstrates behavior derived from k steps of reasoning. Finally, in repeated games, we may need to distinguish between the reasoning done initially, in the so-called *stateless* game, and the decisions made once the game begins and there is some historical state. Even within a single agent, there may be differences of behavior in the two situations, owing to the fact that some of the base strategies require a previous action for initialization. Therefore, let us propose two separate distributions $\vec{\sigma}^I$ for initial actions and $\vec{\sigma}^S$ for the remaining actions. The distinction between these two kinds of decision is analogous to a prior belief versus a belief informed by evidence in the Bayesian sense. A decision made solely based on priors may diverge qualitatively from one made using an inference process because there is no reason why they would be innately linked in heterogenous individuals. In a simpler sense, $\vec{\sigma}^S$ represents a probability conditioned on the game state or joint action.

The mechanism behind constructing these reasoning models is intertwined with fitting the model to data, which can present new problems on the computational front as there is often a tradeoff between precision and complexity. In this context, an estimated value v_a is simply the weighted sum of utilities from that action, where the weights are the probabilities of the strategies of the opponent j: $v_a = \sum_{a_j} \pi_j(a_j) u_i(a, a_j)$. In both the cognitive hierarchy and level-k, some action value v_a estimate is attained for all actions, which then must be converted into a strategy, meaning that some decision

rule is necessary. One possibility is the maximum action value or best response, $\pi_i(a_i) = \arg\max_{a_i} v_i(a_i)$, or some mixture of the max action and uniform noise (the ϵ-greedy rule). Another popular choice is the quantal response function [McKelvey and Palfrey, 1995], also known as softmax or Boltzmann, which exponentially weights the action probabilities by their estimated values.

Definition 15. *(Quantal response) A quantal response of player i to joint strategy π_{-i} is an exponentially weighted strategy tuned by precision parameter λ:* $QRE(\pi_{-i}, a_i) = \frac{\exp[\lambda \cdot v_i(a_i)]}{\sum_{a_i'} \exp[\lambda \cdot v_i(a_i')]}$.

This function over precision parameter λ yields a range of cost-proportional strategies between playing only the best action when $\lambda = \infty$, and all actions uniformly equally when $\lambda = 0$.

2.3 Learning in Multiagent Environments

While reasoning can take place before an opponent is ever encountered, the mechanism of learning allows for adaptive behavior after an interaction starts. There are many learning algorithms that have been applied to game environments, with varying degrees of success. Some methods are designed to reach an equilibrium after a certain time period, while others find cooperative outcomes that are unstable. All of these algorithms share the property that they use experience in the game to generate new behavior, towards an objective of higher rewards.

Definition 16. *(Strategic learner) A* strategic learning *agent i implements an update rule $L(\pi_{i,T-1}, \pi_{-i,T-1})$ which outputs a new strategy $\pi_{i,T}$ so that if $\pi_{i,T}(a_{ij}) > \pi_{i,T-1}(a_{ij})$ and $\pi_{i,T}(a_{ik}) < \pi_{i,T-1}(a_{ik})$, then $u(a_{ij}, \pi_{-i,T-1}) > u(a_{ik}, \pi_{-i,T-1})$.*

We assume that T refers to trial T, which may consist of several rounds of a match, depending on whether each match has a single round or many.

Some learning algorithms operate using a prescribed learning rate α, which may change over time. Most algorithms also undertake some exploratory actions to evaluate the various courses of action open to it, if action is necessary to gain further experience. This exploration is sometimes described as noise or precision. We can broadly distinguish between two types of learners: *value-based* and *policy search*, which differ in their decision-making and exploration selections. Value-based learning tracks the values of actions taken, measured by the payoffs received. Policy search directly alters the probability that certain actions are taken along a positive payoff gradient.

Instead of attempting to fit the actual dynamical system that can take any of countless forms, this dissertation will focus on using the meta-reasoning framework as a general algorithm for modeling the externally-observed behaviors. In some cases, the basic model will have to be adapted to account for different learning outcomes resulting from changes in the strategy used by the modeling agent. The variation in outcomes has two sources. One is that the policy optimized over discounted future rewards (see next section) changes when the modeler's strategy changes. The second stems from the mix of strategies that occurs as a result of the learning process itself, which we might call a dynamic equilibrium. These topics will be covered in more detail in Chapter 4.

2.4 Planning over Discounted Time Horizons

While learning can be applied in situations where there is repeated feedback, there may be choices to be made based on the limited information available.

We will refer to a process of *planning* to describe action in cases where immediate experiential knowledge is scarce but a sequence decisions needs to be made nonetheless. The task of planning has been explored a great deal throughout the history of artificial intelligence [Sutton, 1990]. Here, we focus on one aspect of the planning process that is encapsulated in the time horizon used to make decisions.

The time dimension of decision makers can have a big impact on the decisions taken, especially when learning or reasoning is involved. It is important to know whether a set of agents is making decisions to maximize short-term or long-term payoffs, as sequential decisions are meant to lead to some goal. Behind the forward-thinking decision is the discount factor γ. This parameter can be viewed as a proxy for patience or valuing the future, as it is used to downweight future rewards and is typically used in descriptions of sequential decision making and in many economic settings. The valuation method known as the time value of money uses this method of exponentially declining future values, and so has a strong basis in financial operations [Williams, 1938], whether or not it is an accurate picture of human behavior. A discounted future reflects the universal conflict between smaller rewards now and delayed higher rewards some time later. It is especially required when the future states or rewards are uncertain, such that some evidence about the distribution of the current world must be acquired through learning. In sequential domains, these tradeoffs must be made all the time. We will explore settings where the amount of discount applied to future utility changes the strategies employed, with a new finding that ties regret to future expected gains.

Given its potential impact, it is surprising that this factor has not been incorporated in any opponent models in the literature. To address these issues

Table 2.1: The game of Chicken. The two Nash equilibria are defined by the (Hawk, Dove) and (Dove, Hawk) joint actions.

R/C	Dove	Hawk
Dove	$3, 3$	$1, 4$
Hawk	$4, 1$	$0, 0$

we now define a γ-model as one where discount rates play a prominent role in the decision-making of strategic agents. Because γ measures how much an agent values the future, a value u_t received at time t is worth $\gamma^t u_t$ at time 0. This form is mathematically convenient because of the so-called memoryless property of geometric discounting— we can always convert the value at step t to the value at step $t+1$ by simply multiplying by another factor of γ. Further, this additional parameter allows us to minimize the considerations of inductive depth by looking at the future as a whole and focus instead on strategic depth.

To see the difference between these two forms of reasoning, consider the game of Chicken (Table 2.1), which introduces the idea of a social standoff, where one can either Swerve or Dare. (Alternately, the daring player is a Hawk and the swerver a Dove.) A player does better if the other Swerves, but self-interest dictates they should play the opposite of the other. Since two Hawks are the worst possible outcome, the Nash equilibrium is for one player to be a Hawk and the other to play the Dove. In a repeated context, it is possible for players to enforce the social optimal outcome, although it may be unstable if this efficient payoff is much less than deviating to Hawk. It also may be stable for two players to alternate these roles, but there can be coordination difficulties. Imagine a finitely repeated Chicken scenario where two players start in the Hawk role, producing suboptimal utilities. As soon as one plays Dove,

the actions are fixed for the rest of the game. It is clear that one player should deviate from this arrangement, but when? If a player believes the opposing player is going to be the Hawk for all time, he will end up as the Dove in the last round. Due to inductive reasoning, this strategy unfolds to the present moment. A player operating under this set of conditions applies a reasoning rule to arrive at the belief that the other is a Hawk. From a strategic standpoint, it is therefore more advanced and profitable for an agent to convince, or teach, the other that she is in fact a Hawk.

At the same time, a player in a repeated game is faced with a learning, or perhaps meta-learning, problem, against an unknown opponent from a distribution of strategies. It may be the case that there is a distribution over possible strategies and a decision rule is required to react to whichever members of that population happen to be present in the current sequence of trials. For example, imagine there is an unknown probability that a player will permanently change from Hawk to Dove for the remainder of the interaction. Then, the longer that the learner sees the opponent playing Hawk, the higher it must revise the estimated probability that Hawk is played again. A Bayesian learner might now expect to see the same number of Hawk moves it has already seen, say for τ periods.

Here, we use this observation to further define the γ-model for use in repeated games. The discount factor γ_i is used to balance the desire to minimize regret with the expectation that the opponent will back down, changing the state of the game to a more rewarding outcome for the learner playing Hawk. Two learners playing Chicken comprises a meta-game over the learning parameter space in which the player with the higher γ_i becomes the Hawk and the lower γ player the Dove. As in the single-shot game, two opposing players

with high γ have worse long-term utilities than two players with low values of γ.

The problem of inferring a player's γ_i given a set of observations amounts to some simple algebra using the expected time for higher rewards along with the current utility values. That means that the player will experience regret, ρ, from the missed opportunity of playing Hawk instead of Dove, but anticipates a future gain, G, from the other player eventually becoming the Dove. Therefore, upon witnessing a player starting to play Dove in repeated Chicken, we solve the following inequality for a bound on γ where the expected regret exceeds the expected gains:

$$\begin{aligned}
\sum_{t=0}^{\tau-1} \gamma^t \rho &\geq \sum_{t=\tau}^{T} \gamma^t G \\
\frac{1-\gamma^{\tau-1}}{1-\gamma}\rho &\geq \frac{\gamma^\tau - \gamma^T}{1-\gamma} G \\
(1-\gamma^{\tau-1})\rho &\geq (\gamma^\tau - \gamma^T) G \\
(1-\gamma^\tau)\rho &\geq \gamma^\tau G \\
\gamma &\leq \left(\frac{\rho}{\rho+G}\right)^{1/\tau}.
\end{aligned}$$

Assuming $T >> \tau$, we therefore arrive at an upper bound for γ, given a switch to Dove at time $t = \tau$. Given γ_i, this equation can be used to yield an estimate for the wait time:

$$\tau \leq \log_{\gamma_i}\left(\frac{\rho}{\rho+G}\right).$$

This principle can be employed in any game situation with a standoff payoff structure, such as location games as explored in later chapters. In location settings, aggressive posturing often arises when two or more players find themselves fighting over the same territory. In this context, standing one's ground

is similar to being Hawkish, while moving to a less desirable location amounts to a Dovish choice.

An alternative method for estimating γ, given an observed history, is to view an action as an investment in a future flow of payoffs. In this context, an action may be suboptimal in the current round, but it results in long-term benefits because of the way that others will respond. As before, the calculation rests on the ability to project an agent's expected future payoffs given its beliefs. In other words, the amount of projected future reward as the result of some action with a short-term cost will decrease over time, and at some point it will make sense to switch to the better short-term option. The selected time for this change will reveal the value of γ that would best fit the set of observations. For example, if a player must pay 1 to get 1.5 in all future rounds (up to time T), then she will stop paying at time τ when $1 \geq \sum_{t=1}^{T-\tau} 0.5\gamma^t$.

2.5 A Formal Model of a Population of Strategies

Definition 17. *(Population Model) A population model $\mathcal{M}$ of strategic agents is defined as consisting of the following components:*

- *P: Population of N agents indexed by i*
- *S: Set of states*
- *A: Set of actions*
- *$U(S, A)$: Utility functions $S \times A \rightarrow \mathcal{R}$*
- *M: Hypothesis set of learners or reasoners*
- *H: Set of hierarchies resulting from a reasoning rule*

- $M_i(\gamma, \vec{\sigma^I}, \vec{\sigma^S}, (\epsilon, \phi, \mu), U, s_{i,t})$: *Strategic learner-reasoner (corresponding to a player i) over payoff function U, game state $s_{i,t}$, and model parameters:*
  - γ: *the discount factor or time horizon*
  - $\vec{\sigma^I}$: *distribution over initial levels indexed by k, the depth of reasoning parameter*
  - $\vec{\sigma^S}$: *distributions over state-based levels indexed by k, the depth of reasoning parameter*
  - ϵ: *randomness probability*
  - ϕ: *repeat probability*
  - μ: *imitation probability*

  returns π_i, the strategy function, a distribution over actions, at time t.

The meta-learning/meta-reasoning problem is to use training data D consisting of input set X and output set Y. We shall assume that each input x represents a sequence of states, composing a *trajectory*. Each output y refers to a single action or joint action made by our sample population in the trajectory denoted by x. A single trajectory $x_i = \{s_0, s_1, ..., s_T\}$ consists of a finite sequence of games, within a larger loop of repeated learning trials. Model hypothesis M_i is a reasoning process that takes as input x_i and returns a strategy function π_i, which in turn outputs an action y_i. The strategy returned depends on how hierarchy H_i constructs a π_i from its parameters as well as the values of these parameters for a particular instance of an agent. With enough observations of a real-world agent, we can then test this hypothesis to see if the behavior matches sufficiently. Given the above model definition, the problem analyzed in this document can be posed in two parts. First, how do we *predict*

the actions of actual players in a particular game? Second, how can we build agent strategies to *respond* well against an unknown population?

The goals of prediction and performance are often related, in the sense that successful opponent prediction can lead to a stronger performance, and we might evaluate the prediction in terms of eventual performance. However, there may be situations where a researcher is not interested in building agents but merely needs to makes sense of data, as in social network analysis. Agent designers might not necessarily require accurate predictions as long as the resulting strategy does well. Therefore, we split the problem in this way. We should further note that game theorists rarely aim to predict behavior, even in experimental contexts [Camerer, 2003; Costa-Gomes et al., 2001]. Instead, the preference is for a post-hoc analysis as is the standard in econometric models, where the aim is typically to explain existing data, not predict future data. Often the entire data set is employed when building models, opening these analytic methods to criticisms of over-fitting.

Often, the resulting trajectories are themselves drawn from distributions of behaviors, and so cannot be predicted exactly; high-performing results will exist only in expectation. In effect, we are ultimately looking to infer the distribution of strategic level-types across the entire population and may not be too concerned about particular individuals unless we are able to identify them on an ongoing basis for the purpose of prediction. This model definition provides a hypothesis class C of learning/reasoning/planning algorithms that are candidates for explaining agent behavior even as the above parameters (or perhaps utility functions or preferences) are unknown. Each hypothesized reasoner M_i operates on a trajectory and outputs a strategy function π^t taking

inputs (s_t, u_i) and returning new action a_{t+1}. It is important to note that in order to limit complexity of the meta-learner representation, we will assume that the computation M_i carries out over trajectory x to output π^t is constrained to some simple mapping from current state and utility to action. For example, a meta-reasoner might update a small number of variables to update this strategy, but there will not be an exponential number of strategy function mappings from entire trajectories. Some examples of this type of behavior would be particular iterated best response methods or learning algorithms. These have been explored extensively in the literature under the description of Q-learning or gradient adaptation and will be detailed in later chapters.

We can evaluate a hypothesis or the resulting meta-learned model instance in a number of ways, depending on the desired goal. In one case, we might want to test the performance of an agent designed to respond to the predictions, while in others it might suffice to use the action–prediction error itself, and ask whether we can predict within ϵ of the actual target behavior.

As a naive first attempt, we might consider fictitious play [Brown, 1951], which assumes that a player will continue to demonstrate the same fixed strategy that has been observed previously: $\hat{\pi}_j$. This basic model is equivalent to assuming at each decision point that the other player is stationary and not looking to change strategy over time or adapt it to a new situation to achieve a higher score. Assume for the moment all players are fixed in their strategic ability and play the same strategy in a given game G_0. However, consider the problem that occurs if a new game, G_1, is played. The historical distribution over actions $\hat{\pi}_j$ is meaningless if we want to predict the next action. It is in this scenario where a reasoning model can assist us by transferring the knowledge gained from experience in G_0.

The model construction process investigated here proceeds over three steps, corresponding to the strategy-function calculation, inference of the model hypothesis, and application of the meta-learned model to new inputs. The representation of the strategy function depends largely on the hypothesis choice, but also on how we would prefer to measure an agent's action precision. While purely rational agents achieve perfect precision, we might wish to allow for some noise, whether we take this noise to be exploration, a trembling hand, or some other source of error. One example of a strategy function was discussed above, and it depends on action-value estimates as well as the precision to convert the values into a strategy for reasoning level K:

$$v_{i,K}(a_i) = \sum_{k}^{K-1} \sum_{a'_{-i}} \hat{\sigma}_{i,-i,k} \pi_{-i}(a'_{-i}) u_i(a_i, a'_{-i})$$

$$\pi_{i,K}(a_i) = \frac{\exp[\lambda \cdot v_{i,K}(a_i)]}{\sum_{a'_i} \exp[\lambda \cdot v_{i,K}(a_i)]}.$$

This estimate is made with the probability that each level is expected to play an action assuming a strategy profile $\pi_{-i}(a_{-i})$ of the other players, and population belief vector over level-types $\hat{\sigma}_{i,-i}$. To identify an instance of a hypothesis, a meta-reasoning algorithm needs to compare the output of a strategy function with the data. Using Bayes' Rule, we find the probability that an agent is operating at that level-type, given its behavior:

$$P(L_i = K|a_i) = \hat{\sigma}_{i,i,k} = \frac{\pi_{i,K}(a_i)}{\sum_{k=0}^{K} \pi_{i,k}(a_i)}.$$

Using simple inference we can either apply max-likelihood to focus on a single level-type, or keep the belief vector over the set of all level-types. It may also be necessary to perform this task for both initial-action levels and state-based action levels. That is, a player may act differently from the starting point than he or she does once other players have made their moves and the game is afoot.

Another option, if the level-based strategies can be represented as a binary decision or only a single best response candidate is necessary (per discount rate), is to simply use regression on features extracted from the behavioral history. That is, the meta-reasoner can identify which actions would have been taken if the agent was acting according to each of the model components, and then use these features to compute the relative frequencies of each strategy. This method will be especially useful when there are several base strategies that appear to explain much of the behavior, as will be the case in later chapters.

To estimate the long-run values of actions, we can simulate the likely course of the game for each possible action choice (see Algorithm 1). Using the estimated frequencies, the planning algorithm will then project how the population is likely to initiate and react. Assume for the moment that the course of a game can be divided into two time periods: initial and state reaction. Following the initial choices of the players, the ensuing standoff keeps everyone in the original configuration. At some point, someone shifts from this state to a new action, and the game enters its next phase. To apply the model using the long-run value-estimation algorithm, first consider the possible outcomes allowed by the model. The model contains probabilities that allow us to weight each outcome by the observed likelihood in the population, thus leading to an estimate of the values of each initial action. This process may continue to repeat for a number of iterations, but in general zero-sum games do not create these social standoffs and so the initial action does not matter as much. Consider Rock-Paper-Scissors.

Example 1. *Rock-Paper-Scissors.*

Imagine we would like to construct an agent to perform well in a Rock-Paper-Scissors (RPS) tournament in the following form. An agent observes a history of play of agents in a population. One of these agents is drawn from the population to be our opponent and we are not told which one. Both players are given the previous action by their opponent and remember their own last action. A meta-reasoning algorithm is given the historical sequence of the population in this form ($x_i = [a_B^{t-1}, a_C^{t-1}], y_t = [a_B^t, a_C^t]$) and asked to play an action. For the moment, let us assume that all or most members of the population employ fixed strategies, which do not significantly change over time. In this game, prediction and performance go hand in hand, so we would like to compute an expected distribution over actions, requiring an accurate opponent model.

A naive learner might do something like fictitious play [Brown, 1951], and respond to the overall population frequency of the actions we have seen. This type of action is similar to an initial strategy level, as it would use only a summary of the small amount of historical knowledge it has available. As a result, in the typical case of RPS, we would expect the action distribution to be roughly uniform. A better approach would involve building a frequency table of the nine previous joint action combinations, which may lead to more information about the response structure. An even more efficient model, from the standpoint of maximum generalizability, would use knowledge of reasoning to meta-learn the expected depth of thinking about each response. In essence, this model would harness the symmetry from each joint action to estimate how players respond to the opponent's previous action, the response to one's own action, and so on. This meta-reasoning method would provide considerable

transfer and speed up the model-building process on the order of the number of actions, as long as we assume symmetry among the actions. With this model of the population's reasoning behavior, our meta-reasoner can apply this knowledge to new games with a similar structure without seeing opponent behavior in them. The winners of previous runs of RPS tournaments applied a version of this strategy to out-think their opponents [Egnor, 2000], although they relied on an individually customized model to do so because it is possible to relearn an opponent model in a lengthy sequence of RPS rounds.

{Inputs: $U, \sigma^I, \sigma^S, \hat{\gamma}, \hat{\phi}$}
{Outputs: $v(a_i)$, value of a given action}
m: Number of actions
$n-1$: number of agents besides i
T: time horizon
for $j = 1$ **to** $j = n-1$ **do**
  for $k = 0$ **to** $k = 3$ **do**
    $\pi_j^{t_0}(a_j) = \pi_j^{t_0}(a_j) + \sigma_k^I \pi_k^I(a_j)$
  end for
end for
for $a_i \in A_i$ **do**
  for $a_{-i} \in A$ **do**
    for $j = 1$ **to** $n-1$ **do**
      Calculate ρ_j, G_j
      $\tau_j = \lceil log_{\hat{\gamma}}(\rho_j/(\rho_j + G_j)) \rceil$
    end for
    Find $j_{\min}$ where $\tau_{j_{\min}} = \min_{j'}(\tau_{j'})$
    for $k = 0$ **to** K **do**
      $\pi_{j_{\min}}^t(a_{j_{\min}}) = \pi_{j_{\min}}^t(a_{j_{\min}}) + \sigma_k^S \pi_k(a_{j_{\min}}, a_{-j_{\min}})$
    end for
    $v(a_i) = \sum_{t=0}^{\tau_{j_{\min}}-1} \gamma_i^t u(a_i, a_{-i}) + \sum_{t=\tau_{j_{\min}}}^{T} \gamma_i^t \pi_j^t(a_j) u(a_i, a_{j_{\min}}, a_{-i,-j})$
  end for
end for

Algorithm 1: Long-run Value Estimation in Populations

2.6 Related Work

From a theoretical standpoint, people have used the concept of best response to discover and analyze equilibria since the beginnings of game theory. The iterated best response technique (IBR) consists of an initialization of opponent strategies π_j and then a recursive application of best response until the opposing strategies are in equilibrium:

- $\forall j$ Initialize π_j
- repeat until π are in equilibrium
  - $\forall i$ player i finds the best response $\pi_i = BR(\pi_j)$

Given proper beliefs over initial strategies, IBR provably arrives at the Nash equilibrium for a normal form game [Stahl and Wilson, 1995]. The repeated setting, which is our main focus, makes analysis more complicated because there may be an infinite number of equilibria, considering the endless number of possible strategies [Aumann and Maschler, 1995]. The hierarchy output by IBR in this case will have a high dependence on the initial non-intentional or non-reasoning strategies, because higher levels derive from earlier levels. The lack of unique equilibria makes it clear that modelers must be cautious in the choice of initial assumptions.

The concept of IBR has also been explored in the field computer science under the guise of designing agents to act in an environment with others. The focus is usually on building algorithms to model others as experience accumulates in a repeated interaction, as in the RPS tournaments [Egnor, 2000]. Another approach is to construct an optimal model given prior beliefs about

the other agent combined with some computational cutoff. However, this focus is either used in an online learning setting [Koller and Milch, 2003; Gal and Pfeffer, 2008] or where the priors are known [Gmytrasiewicz and Doshi, 2005], and not where there is some established history of multiple interactions. See the sections below for further details.

2.6.1 Network of Influence Diagrams

One related development is the Network of Influence Diagrams, or NIDs [Gal and Pfeffer, 2008]. This construct incorporates the Multi-Agent Influence Diagram (MAID) [Koller and Milch, 2003] as a building block to model the behavior of other players. MAIDs were introduced to formalize a multiagent system as a kind of Bayesian network consisting of chance nodes, decision nodes, and utility functions. The chance nodes are simple random variables. Decision nodes are the actions of the agent, and utilities are the reward. A decision node is not pre-specified, but must be provided with some strategy inferred by the algorithm, at which point it becomes a chance node. Given conditional probabilities of the chance nodes, these diagrams can be solved for the optimal strategy profiles. Each MAID has at least one Nash equilibrium, and there are proposed exact and approximate algorithms for finding solutions [Koller and Milch, 2003]. An NID is basically a collection of these MAIDs that represent the models the players have of each other, so that each node is a MAID. The modularity provided by this higher-order model makes construction of the underlying decision process less difficult. In practice, the parameters of the NID (the conditional probability distributions at each nested MAID section) need to be estimated from data so that the whole MAID equilibrium can be found. One way to use this tool is with random initialization and then a variation of the EM

algorithm. The accuracy of this model depends on the level of detail of the network, and finding the right probabilities may require a large amount of data. Furthermore, this model is best applied in situations where there is a single opponent. With multiple opponents (two or more), the complexity of inferring the conditional probabilities in the model increases dramatically, as each player requires a model of the interactions between the others. The population-based meta-reasoning framework introduced above can also be considered as a graphical model of sorts, but in many cases it is not necessary to represent other agents in this way. A probability vector over types, encoded as a linear model, is often sufficient to capture the distribution over strategies, as we will see in later chapters.

2.6.2 Cognitive Hierarchies/Level-K Reasoning

This work also borrows both ideas and terminology from the emerging field of behavioral economics. Such work provides significant empirical grounding for the work done here. Using experiments on humans playing games, researchers have found a great deal of evidence that people use strategic reasoning to make decisions, but only up to a point. Indeed, this reasoning conforms to a well-defined cognitive hierarchy, or a related level-k model, composed of levels of thinking [Stahl and Wilson, 1995; Costa-Gomes et al., 2001; Camerer, 2003]. This model can apply to games with two agents or larger population games. The components and structure of these models was introduced above, and they have been applied in a growing number of behavioral experiments.

Lately, this line of investigation has been finding its way into computer science and neuroscience research. Recent work [Wright and Leyton-Brown, 2010,

2012] provides a comprehensive review on which existing models are the likeliest explanations of a series of previous experiments with human participants, from the standpoint of prediction. The conclusion of this survey is that a quantal response cognitive hierarchy model appears to be the most likely model of a population of players. The parameters for the population were trained using either max-likelihood or Bayesian model-fitting, and assume a homogeneous hierarchy for all players, who nonetheless play according to a different degree of sophistication.

2.6.3 I-POMDPs as Level-based Reasoners

From the computer-science or machine-learning perspective, the most relevant work when it comes to optimizations on a known opponent model the as an Interactive Partially Observable Markov Decision Process model, or I-POMDP [Gmytrasiewicz and Doshi, 2005]. This development synthesizes the considerable work done on single agent POMDPs [Kaelbling et al., 1998] with multiagent approaches such as the Recursive Modeling Method (RMM) [Gmytrasiewicz and Durfee, 1995]. By partially observable, we mean that the agent does not directly observe its state of the environment, but can only adjust its beliefs over the set of states. RMM is a tree-based framework that starts with a Zero Knowledge strategy at the leaf nodes to represent the lack of knowledge about other players. The method propagates strategies up the tree if reasoning can be performed on the lower nodes. This combined formulation is ideal for sequential or repeated games where the opponents' strategies are unknown—hence the partial observability—and have limited reasoning capabilities so that recursive modeling makes sense. The I-POMDP formalism allows for a partially observable environment as well, but we will consider games that limit

the state uncertainty to the other agents only, for the sake of simplicity. There is some literature that examines human behavior in a sequential game (the Investor/Trustee game) using I-POMDPs and fits parameters of the underlying models to the data [Ray et al., 2008]. Below is a version of the I-POMDP formalism that could be applied to games.

Let us assume two agents, i and j and associate I-POMDP_i with agent i. An $\text{I-POMDP}_i = \langle IS_i, A, T_i, \Omega_i, O_i, R_i \rangle$ has the following features:

- IS_i is the set of interactive states $IS_i = S \times \pi_j$ where S is the set of states from the environment and π_j is the set of policies for agent j.
- A is the set of joint actions $A_i \times A_j$
- T_i is the transition function $T_i : S \times A \times S \rightarrow [0, 1]$. The transition also combines with the internal decisions for the model π_j to lead to a new interactive state, but we assume that agent i does not directly control this part of its environment.
- Ω_i is the set of observations
- O_i is the observation function $O_i : S \times A \times \Omega \rightarrow [0, 1]$
- R_i is the reward function $R_i : IS_i \times A \rightarrow R$

Define for agent i rule-based policy $H : IS \rightarrow A_i$ to be a basic rule that maps states to actions. One example of a rule would be $A_i^t = A_i^{t-1}$, signifying constant action. Then, a parameterized policy $\pi : \mathbf{R} \rightarrow H$ maps real vector $X \in [0, 1]$ to some rule. X could be used to represent any adjustable feature of an agent, but we will assume that X_r is the probability of playing rule H_r. Note the rule is not fixed for the whole game, but rechosen every time step. Knowing

that our opponent is tied to a single value of X, we can utilize POMDP solving methods to estimate this value and respond to it to build the next level. In certain games, the computed policy will be constructed of a new set of rules, which composes a parameterized policy.

Policies at each level k are derived from the beliefs $b_{j,k-1}$ over the policies and states of the previous level $k-1$. Define the following spaces:

- $IS_i^0 = S, \qquad \pi_j^0 = IS_i^0 \rightarrow A_j \in H_0$
- $IS_i^1 = IS_i^0 \times \pi_j^0, \qquad \pi_j^1 = b_{j,1}(IS_i^1) \rightarrow A_j \in H_1$

  .

  .

  .

- $IS_i^L = IS_i^{L-1} \times \pi_j^{L-1}, \quad \pi_j^L = b_{j,L}(IS_i^L) \rightarrow A_j \in H_L.$

With this definition, we have a way of constructing a strategic hierarchy given limited knowledge about other agents.

While the I-POMDP formalization introduces a complete encapsulation of the state space of other reasoning agents, it suffers from several drawbacks. First, it makes no attempt to exploit problem structure or reduce the size of the state space, which is likely to grow large given the presence of outside agents—a considerable source of new uncertainty for each new joint action. Second, while a two-player scenario invites an intuitive back-and-forth reasoning process, with three or more players even describing how reasoning might work becomes difficult. There is nothing to stop the interactive state space itself from growing exponentially in terms of both agents and levels, as a planner attempts to model each player as it tries to model each player, etc.

A bigger problem arrives when we must decide upon the distribution (henceforth population) of models to include in each subsequent level. A default assumption for I-POMDPs or RMMs is to assume uniform random behavior over actions at the lowest level of sophistication. However, a sole focus on this particular policy eliminates other potentially relevant and also justifiable choices. For example, for the repeated case we might consider other strategies, such as repeating the same action (constant) or unchanging probabilistic transitions between actions (static). See Section 2.2 for more details about non-reasoning strategies. The less restrictive assumption is to set the bottom layer to a distribution of these types of policies. Some of these policies would provide orthogonal directions of reasoning to discover unique properties about the game in question. As a result, the rest of this work examines a simplified framework where the base non-reasoning level is a mixture of a specified set of components, and the optimization proceeds without attempting to classify a strategy but instead responds to the known state of the game.

2.7 Summary

This chapter has introduced some basic game theory concepts and definitions, along with some related work in the field of iterated best response models. The literature about iterated reasoning models is rooted in the fields of behavioral economics and multiagent learning (within computer science) but suffers from several shortcomings. First, previous models are only practical in multiagent environments with a single opponent, so that the system consists of two agents. The computation becomes much less tractable once the setting has more than one opponent. Second, the settings are not optimized for planning in repeated domains. In the Rock-Paper-Scissors tournaments, for example, the models

try to predict the next action, without regard to how future time periods are affected by actions. This type of reasoning is sufficient for zero-sum games, but general-sum games require a more nuanced approach.

To address these issues, this document proposes a systematic meta-reasoning modeling algorithm for populations playing repeated games. It is meant to be robust to noise by introducing flexibility in classification of agent strategies. This chapter introduced a broader definition of base strategy, to include repetition and imitation to the typical assumption of randomness. Finally, the meta-reasoning model allows for opponents to take planning steps by estimating a parameter for the discount rate. Combined with a level-based identification process over the expanded set of base strategies, the discount rate can capture the capacity for future lookahead that has not been a part of previous models.

Chapter 3
Learning Algorithms in Simple Games

This chapter investigates the dynamical behavior of agents that are learning over time in response to their perceived environment. Ultimately, we will see how the modeling framework introduced in Chapter 2 can be applied to this problem. First, we will review some learning algorithms that have been explored in game settings. Some of these will be covered briefly, but Q-learning deserves greater detail due to its remarkable properties in the game space.

In the single-agent context, a number of algorithms have been developed [Sutton and Barto, 1998] that are guaranteed to converge to the expected values of actions in the reachable states, and therefore achieve an optimal policy, when the environment and rewards are stationary. An environment with multiple agents does not meet this requirement, because the agents themselves have the capacity to generate non-stationarity. As a result, the same algorithms can arrive at inaccurate estimations of the values of strategies, because the observations become obsolete when the other players change their policies. This notion of optimal policy is therefore less well-defined because it does not apply in non-stationary environments. Even so, in some cases, we can expect that the learning algorithms that are present will behave consistently at a higher level. This assumption allows for a modeling agent to capture these dynamics, and by applying an adequate model, approximate this situation as a stationary environment at a higher level. Knowing the properties and possible outcomes of

learning algorithms will aid in the modeling process.

It is important to note the distinction between a repeated game and repeated learning trials. A repeated game is when a single-shot game is repeated more than once, and often many times. Players may hold the outcome of the previous round of the game in memory for future use. Unless otherwise stated, this chapter will operate under the assumption that repeated games are infinitely repeated. If a game is simply repeated because a learner is attempting to gain experience, then the game can be viewed as the non-repeated case. Most of these algorithms are used with the latter in mind, while this chapter (and the thesis itself) is primarily concerned with the former. Nevertheless, the behavior of these algorithms in the truly repeated setting reveals some interesting phenomena, and in any case the convergence result is often the same.

If our goal is to model a learning algorithm for the purpose of predicting and responding to it, then learning takes place in the space of repeated trials of multiple rounds in each trial. The next chapter will focus on how to build a model from an observed history using the base strategy with discounting framework.

Below is a detailed account of how learners behave will demonstrate how non-equilibrium play can emerge naturally from the resulting dynamics. The first few sections (3.1-3.3) are background that define different variants of learning algorithms that have been used in a multiagent context. The remainder of the chapter is then devoted to ϵ-greedy Q-learning, which is a challenge for modelers because it can be induced to play out-of-equilibrium, but is unstable and so its behavior is a function of its opponent's strategy. The rest of this

chapter is an original contribution [Wunder et al., 2010]. Section 3.4 derives local learning dynamics for the ϵ-greedy Q-learning algorithm. Using a dynamical systems approach, Section 3.5 explores the types of asymptotic behavior corresponding to different classes of games. Section 3.6 goes into greater detail about the conditions for convergence to a cooperative equilibrium, as well as conditions for non-convergent behavior, of ϵ-greedy Q-learning in a specific subclass of games. Section 3.7 compares two learning algorithms, ϵ-greedy Q-learning and Infinitesimal Gradient Ascent, to demonstrate this divergent behavior.

3.1 Fictitious Play

Fictitious play was one of the earliest multiagent learning algorithms developed and it is based on the premise that our opponent is playing a stationary strategy at each decision point [Brown, 1951]. In essence, the algorithm directly models the other player according to the observed distribution over actions and takes it as a fixed model of its world, giving no intentionality or learning ability to its opponent. Then, an agent using fictitious play simply calculates the values of its actions based on this stationary model, and responds using the best one. While it can work well in a variety of environments, this model will be flawed if the underlying assumptions do not hold. In terms of our reasoning framework, this algorithm corresponds to a strict level-one player that actually builds a simplified representation of its opponent and adapts its base beliefs over time.

3.2 Gradient-based Algorithms

Gradient-based methods have become one of the most important approaches to optimization problems of all kinds, where searching a parameter space exhaustively is computationally intractable. The principle is to slowly change the values of the parameters in the direction of steepest gradient change of the target objective. In some fields, such as machine learning, this objective is typically to minimize some cost, such as training error, and so the adaptation proceeds according to gradient descent. Most neural network update methods work by moving along the error gradient, for example [Rumelhart and McClelland, 1986]. The goal in a game setting is to maximize utility, and so the proper term is gradient ascent.

The biggest shortcoming of gradient-based methods is their tendency to converge to a local optimum when a better globally optimal point exists in the space. Some functions are not differentiable or well-defined, and therefore require more sophisticated search methods. If the convex problem is poorly conditioned, the gradient can zig-zag, leading to substantial time to convergence. For the simple games considered in this chapter, the surface is always smooth. However, it is possible that more than one equilibrium exists, particularly for repeated games, and so special care must be taken to consider the initial conditions that will result in one maximum or another.

3.2.1 Infinitesimal Gradient Ascent (IGA)

Infinitesimal Gradient Ascent (IGA) [Singh et al., 2000] defines the joint strategy of the players i and j by a pair (p, q), the probabilities of the first action for both players. Strategies are updated in the direction of the gradient of the reward V

at time t:

$$\begin{aligned} p_{t+1} &= p_t + \alpha \frac{\partial V_i(p,q)}{\partial p} \\ q_{t+1} &= q_t + \alpha \frac{\partial V_j(p,q)}{\partial q}. \end{aligned}$$

It was shown that IGA either leads the strategy pair to a Nash equilibrium or an orbit yielding an average reward equal to the Nash equilibrium.

3.2.2 Win-or-Learn-Fast Infinitesimal Gradient Ascent (WoLF-IGA)

A modified version of IGA, called WoLF-IGA, always converges to the Nash in these games [Bowling and Veloso, 2001]. The central idea behind *Win-or-Learn-Fast Infinitesimal Gradient Ascent* is to make a distinction between winning and losing, and therefore to define separate learning rates for each situation, say, α_W when winning and α_L when losing, so that $\alpha_W < \alpha_L$. So, if player i is scoring above the win threshold θ_i and j is below θ_j, the update formula becomes:

$$\begin{aligned} p_{t+1} &= p_t + \alpha_W \frac{\partial V_i(p,q)}{\partial p} \\ q_{t+1} &= q_t + \alpha_L \frac{\partial V_j(p,q)}{\partial q}. \end{aligned}$$

The learning rates will be set to their appropriate values. There is some room to set θ, but it is important that it is greater than the Nash value. This asymmetry causes losing players to learn faster while winners are slower to adapt, and the net effect is for the dynamical system to spiral into the equilibrium point, when there are two actions and two players in the game.

3.2.3 Cross Learning

Cross Learning is a type of gradient-based algorithm that directly uses rewards to update its policies [Borgers and Sarin, 1997]. The rule for updating action a is

$$p_{a,t+1} = p_{a,t} + R_a p_{a,t} + R_a I(a),$$

where $I(a)$ is an indicator function that takes a value of 1 when action a is played and 0 when it is not. In general, actions that are selected with higher probability are reinforced more often. The idea behind this algorithm is that actions with higher rewards will gradually become preferred, all else being equal. In self-play, it has been found that pure Nash equilibria are stable while mixed equilibria are not.

3.3 Q-Learning

Q-learning [Watkins and Dayan, 1992] was developed as a reinforcement-learning (RL) algorithm to maximize long-term expected reward in multistate environments. It is known to converge to optimal values in environments that can be formulated as Markov decision processes [Tsitsiklis, 1994]. Its elegance and simplicity make Q-learning a natural candidate for application to multi-player general-sum games, leading to questions about its asymptotic behavior in this context. While the study of simultaneous learning agents has generated much interest, characterization of their behavior is still incomplete. Algorithms such as Q-learning that use accumulated data about the values of actions are of interest beyond RL, as similar mechanisms are hypothesized to exist in mammalian brains [Dayan and Niv, 2008].

The Q-learning algorithm is defined by the elegant update equation:

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha(R_t + \gamma \max_{a'} Q_{t-1}(s',a') - Q_{t-1}(a)).$$

R_t is the reward received at time t, $Q_{t-1}(s,a)$ is the estimated value of action a from state s at time $t-1$, s' is the new state at time t, α is the learning rate, and γ is the algorithm's discount rate of future rewards.

In the following sections, we examine the behavior of two players executing ϵ-greedy Q-learning in a repeated general-sum game. Although some applications of Q-learning have used state representations that include recent history [Littman and Stone, 2001], we focus on a simpler representation consisting of just a single state. The idealized algorithm is one that consists of infinitely small learning steps making it possible to apply ideas from dynamical systems theory directly to the algorithm [Tuyls et al., 2003]. Later sections map the varied behavior of this algorithm, using much of the same terminology and methods as has been applied to other multiagent dynamical approaches.

As opposed to a purely *value-based* approach like Q-learning, past work using dynamical systems to analyze multiagent learners has centered on *policy-search* algorithms [Singh et al., 2000] or a mix of the two [Bowling and Veloso, 2001]. In cases where learning is equivalent to or resembles policy-gradient algorithms, researchers have found that adaptive methods tend to converge to a Nash equilibrium [Tuyls et al., 2003] or "orbit" a Nash equilibrium [Singh et al., 2000]. In the mold of this earlier work, the rest of this chapter fully describes the long-run convergence behavior of ϵ-greedy Q-learning—a commonly used algorithm that had not yet been analyzed in this way. A surprising finding [Wunder et al., 2010] is that when Q-learning is applied to games, a pure greedy value-based approach causes Q-learning to endlessly "flail" in

some games instead of converging. For the first time, we have a detailed picture of the behavior of Q-learning with ϵ-greedy exploration across the full spectrum of 2-player 2-action games. While many games finally converge to a Nash equilibrium, some significant games induce behavior that averages higher reward than any Nash equilibrium of the game. Since some of these games have a dominant action, this outcome is somewhat counterintuitive. Nonetheless, this behavior is not merely an empirical quirk but a fundamental property of this algorithm, which holds potentially profound implications.

3.3.1 Reinforcement Learning/Opponent Model Hybrids

Because general-sum games require sophisticated learners to adapt their strategies to their opponents, it makes sense to directly apply lessons from the literature on zero-sum games to this space. Some algorithms have been proposed that supplement the Q-learning update with a shaping term, so that it can better anticipate or determine opponent response.

The first algorithm is called *Minimax-Q* and the idea behind it is to construct a strategy that maximizes the security value, that is, the highest payoff that can be attained assuming the opposing player maximizes his own interests [Littman, 1994]. With this algorithm, the Q-values are set according to the usual update equation in an extended table $Q(a, b)$ for both players' actions, but the strategy for player i is selected by assuming that opponent j chooses the worst payoff for i:

$$\pi_i = \arg\max_{\pi_i}(\min_b \sum_a (\pi_i(a), Q(a, b))).$$

The strategy is constructed by using linear programming. This approach was generalized to all stochastic games under a similar algorithm, *Nash-Q* [Hu and

Wellman, 2003], where the minimax function is replaced by a function that computes the Nash equilibrium. However, there are convergence problems that arise due to the fact that the output equilibrium can change as the Q-values change, as well as the proven intractability of finding a Nash equilibrium in the first place. As such, Nash-Q would only be expected to work in coordination or zero-sum games. A related algorithm, *Friend-or-Foe-Q*, works by using either the assumption that our opponent is our friend (by maximizing our own score) or our foe (by minimizing our score) [Littman, 2001].

3.3.2 Q-Learning with Boltzmann Exploration

Q-learning with ϵ-greedy decisions represents a fixed exploration rate, with a value of ϵ. Another widely used exploration method is known as Boltzmann exploration, or in the economics literature as quantal response. With this method, actions are chosen according to an exponential weighting of the perceived values of the actions. Taking Q_{a_i} as the value for action a_i, strategy p_{a_i} is chosen with the following formula:

$$p_{a_i} = \frac{exp(\lambda Q(a_i))}{\sum_a exp(\lambda Q(a))}.$$

Parameter λ is a precision value which determines how much actions with high values are favored. If $\lambda = 0$, actions are chosen uniformly random. If $\lambda = \infty$, the best action is always selected. Although the values for the actions are updated exactly as in the Q-learning update, this algorithm shares properties with gradient-based methods because the strategies gradually change towards the better actions.

It has been shown that with certain modifications, the behavior resulting from these rules converges to replicator dynamics [Tuyls et al., 2003], which is

a mathematical representation of evolutionary game theory [Borgers and Sarin, 1997]. That is, the strategies adopted by two Boltzmann-type learners change in a way similar to that of two large populations of atomic (infinitesimal) agents who play each action in the game with a proportion equal to the corresponding probabilities in each learner's strategy. Then, the agents are selected to remain in the game in proportion to their success in the game. If A is the payoff matrix of the row player, the rate of change of its strategy p_a is:

$$\dot{p}_a = p_a\alpha(\lambda[e_aAq - pAq] - \log p_a + \sum_{a'} p_{a'} \log p_{a'}).$$

3.3.3 Frequency-Adjusted Q-Learning

Although, formally speaking, the dynamical equations of Boltzmann-Q and replicator dynamics are the same, in practice, the dynamics do not match due to the fact that the update rates vary based on the strategy. To address this issue, a modification called *Frequency-Adjusted Q-Learning* was proposed [Kaisers and Tuyls, 2011] that normalizes the update rates:

$$Q(a) = Q(a) + \alpha \min(\frac{\beta}{p_a}, 1)(R + \gamma \max_{a'} Q(a') - Q(a)).$$

The β term is meant to allow for a maximum update of 1 if the strategy p_a is below a minimum value β. This change allows the learning dynamics to conform to the evolutionary model and as a result, the values, will converge to the equilibrium prescribed by the evolutionary stable point.

While this property provides some guarantee of convergence [Kaisers and Tuyls, 2011] to the single-shot Nash equilibrium, we turn our attention back to look at ϵ-greedy Q-learning, which demonstrates behavior that can lead to globally efficient outcomes without explicit memory or representation of state.

3.4 Infinitesimal Q-Learning

This section investigates Q-learning dynamics more thoroughly, through the behavior of the continuous dynamical system resulting from the underlying update rules.

3.4.1 ϵ-Greedy Infinitesimal Q-learning (IQL-ϵ)

The ϵ-greedy Q-learning algorithm selects its highest valued (greedy) action with some fixed probability $(1 - \frac{\epsilon(k-1)}{k})$ and randomly selects among all other $k-1$ actions with probability $\frac{\epsilon}{k}$. Earlier papers have demonstrated superior performance of this algorithm in games [Sandholm and Crites, 1995; Zawadzki et al., November 2008] relative to similar learners and carried out dynamical systems analysis [Gomes and Kowalczyk, 2009] as a model for ordinary Q-learning. However, these have not systematically documented the resulting range of outcomes of the dynamical model itself, mostly because convergence to an equilibrium is not assured. More recent work [Wunder et al., 2010] has fully described the behavior of a deterministic model of the algorithm for all possible games within the 2-person 2-action space, which is detailed below.

The (one-state) Q-learning update rule when an agent takes action a and receives reward R

$$Q(a) = Q(a) + \alpha(R + \gamma \max_{a'} Q(a') - Q(a))$$

becomes $\frac{\partial Q(a)}{\partial t} = R + \gamma \max_{a'} Q(a') - Q(a)$ when $\alpha \to 0$. We call this deterministic algorithm IQL-ϵ for Infinitesimal Q-learning with ϵ-greedy exploration. The discount rate is γ and when set to 0 the update equation becomes simply $Q(a) = Q(a) + \alpha(R - Q(a))$. We write

- Q_{a_i} and Q_{b_j} for the Q-values of action a_i for row player RP, and action b_j for column player CP,
- $\dot{Q}_{a_i}$ for $\frac{\partial Q(a_i)}{\partial t}$, the update of action a_i for RP and $\dot{Q}_{b_j}$ for the update of action b_j for CP,
- $(r_{a_i,b_j}, c_{a_i,b_j})$ for the respective payoffs, or rewards, for RP and CP when RP plays a_i and CP b_j.

Due to the fact that there can be only one greedy action at a time, IQL-ϵ's updates lead to semi-continuous dynamics best classified as a piecewise-smooth, or hybrid, dynamical system [Di Bernardo et al., 2008]. A *general hybrid dynamical system* (GHDS) is a system $H = [P, F, J]$ with the following parts:

- P is the set of index states or discrete dynamical system states;
- $F = \bigcup_{p \in P} F_p$ is the set of ordinary differential equations (or *flows*) for index state p;
- $J = J_{p \in P}$ is the set of jump transition maps.

The simple IQL-ϵ GHDS can be represented as an automaton whose nodes are four complete and separate dynamical system flows and where transitions between the nodes, or index states, must be taken when certain conditions along them are met. When the values for one of the players' actions change ordering, the system jumps to the index state containing the dynamics corresponding to the new greedy policies. For the following analysis, only one state exists in the agents' environment—all state transitions are jump transitions in this model. In this case, a transition to a new flow occurs when the values cross a boundary $B(Q) = 0$, so that different flows operate when $B(Q) < 0$ and $B(Q) > 0$.

Using this notation, we examine the following equations for a combination of possible greedy actions for the two players. Consider what happens when a_1^* and b_1^* are greedy. RP chooses a_1^* $1-\frac{\epsilon}{2}$ of the time and $\hat{a_2}$ $\frac{\epsilon}{2}$ of the time, making its expected reward $R_{11} = r_{11}(1-\frac{\epsilon}{2}) + r_{12}\frac{\epsilon}{2}$ where ϵ is the exploration rate. In this case, RP will update Q_{a_1} according to:

$$\begin{aligned}\dot{Q}_{a_1} &= r_{11}(1-\frac{\epsilon}{2}) + r_{12}\frac{\epsilon}{2} + (\gamma-1)Q_{a_1} \\ &= R_{11} + \gamma \max_{a'} Q_{a'} - Q_{a_1}.\end{aligned}$$

However, this equation only describes the rate of update *when the value of a_1 is updated.* To capture the exact rate, consider that the greedy action is taken a fraction $(1-\frac{\epsilon}{2})$ of the time. In contrast, the non-greedy action is taken $\frac{\epsilon}{2}$ often. Weighting the updates appropriately, when Action a_1 is greedy for both players, the four Q-values obey the following system of differential equations [Gomes and Kowalczyk, 2009], $F_{a_1^* b_1^*}$:

$$\begin{aligned}\dot{Q}_{a_1} &= (R_{11} + Q_{a_1}(\gamma-1))(1-\frac{\epsilon}{2}), \\ \dot{Q}_{a_2} &= (R_{21} + Q_{a_1}\gamma - Q_{a_2})\frac{\epsilon}{2}, \\ \dot{Q}_{b_1} &= (C_{11} + Q_{b_1}(\gamma-1))(1-\frac{\epsilon}{2}), \\ \dot{Q}_{b_2} &= (C_{12} + Q_{b_1}\gamma - Q_{b_2})\frac{\epsilon}{2}.\end{aligned}$$

We can find the solutions for the above equations using linear dynamical systems theory [Di Bernardo et al., 2008]. While the solutions define a single dynamical flow $F_{a_1^* b_1^*}$ where a_1^* and b_1^* are the greedy actions for RP and CP, similar equations can be defined for the other three joint greedy policies. Note that because the system can switch flows, the values may not converge to the end point dictated by this flow alone. We say that the learning algorithm has *converged* if the ratio of strategies in an extended period of time stays within

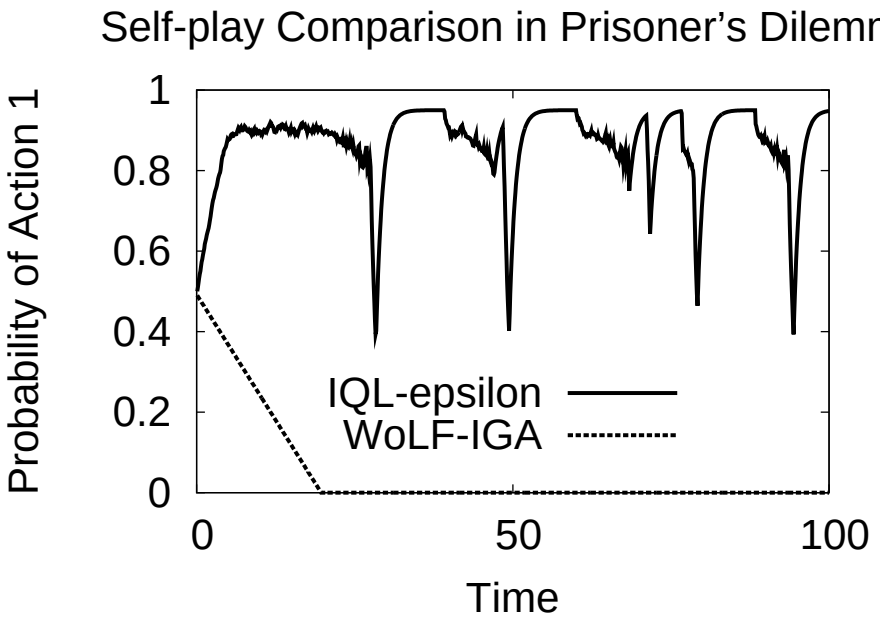


Figure 3.1: Probabilities for Action 1 (Cooperate) for RP in two self-play scenarios both in the Prisoner's Dilemma game. WoLF-IGA is seen to converge to the defection action (Nash), while IQL-ϵ oscillates around a mix of both actions, mostly cooperation. See Figure 3.2 and Section 3.7 for more details.

an infinitesimally small range. See Figure 3.1 for examples of converging and non-converging policies. Also, note that the equations are deterministic, in spite of the random exploration, because of the infinitesimal learning rate.

3.4.2 One-player Sliding Greedy Update

In cases in which the convergence points of the flows lie within the index state of a single flow, the above IQL-ϵ analysis is sufficient to disclose the final destination of the algorithm's values. If there is disagreement, the IQL-ϵ GHDS can end up with dynamics that slide along the boundary between two or more index states. An investigation of the resulting dynamics, known as a *Filippov sliding system* [Di Bernardo et al., 2008], is crucial for analyzing these more

complex situations.

When one player has two equal Q-values and both sums of discounted rewards are lower than the current value, this player has a *sliding* greedy action. The values may change in lockstep, although the two actions are selected at different rates. Consider what happens when CP has one clear greedy action. Figure 3.2(inset) shows an illustrated example of this update dynamics. Here, the two actions for RP have the same value and the Q-values for both players drop until CP's greedy action switches. The term "greedy" does not fully capture this type of dynamics for RP because, essentially, its greedy action alternates infinitely often over a given interval so it has no particular greedy action. Instead, define the current *favored action* to be the action f with the higher expected reward during a sliding update (let $\bar{f}$ be the other action). It turns out that f also has a higher probability of play than $\bar{f}$ when both values are dropping. Therefore, f is played by RP more often. Define ϕ_f to be the fraction of time where RP plays f. The updates $\dot{Q}_{\bar{f}}$ and $\dot{Q}_f$, taken from the definition of Q-learning, capture the change of respective Q-values over continuous time, observed separately. The formula for ϕ_{fb^*} is the ratio of the non-favored action's update rate to the total update rate while CP's greedy action is b^* and its non-greedy action is $\hat{b}$:

$$\phi_{fb^*} = \frac{\dot{Q}_{\bar{f}}}{\dot{Q}_{\bar{f}} + \dot{Q}_f} = \frac{r_{\bar{f}b^*}(1-\frac{\epsilon}{2}) + r_{\bar{f}\hat{b}}\frac{\epsilon}{2} + Q_{\bar{f}}(\gamma - 1)}{(r_{\bar{f}b^*} + r_{fb^*})(1-\frac{\epsilon}{2}) + (r_{\bar{f}\hat{b}} + r_{f\hat{b}})\frac{\epsilon}{2} + (Q_{\bar{f}} + Q_f)(\gamma - 1)}.$$

There is a natural intuition behind the ratio ϕ_{fb^*}. Ignoring exploration, if each update is different and negative, the algorithm more often selects the one that decreases more slowly because it is more often the greedy action. In fact, the ratio selected is identical to the ratio of the other value's standalone rate of decrease to the combined rates for both actions. If RP plays f with this proportion, then both values actually decrease at the same overall rate as the

faster one is selected less frequently. As a result, the update rates for CP depend on this fraction ϕ_{fb^*}:

$$\begin{aligned}\dot{Q}_{b^*} &= ((1-\phi_{fb^*})c_{\bar{f}b^*} + \phi_{fb^*}c_{fb^*})(1-\frac{\epsilon}{2}),\\ \dot{Q}_{\hat{b}} &= ((1-\phi_{fb^*})c_{\bar{f}\hat{b}} + \phi_{fb^*}c_{f\hat{b}})\frac{\epsilon}{2}.\end{aligned}$$

This reasoning only applies to falling values. If values rise, the arbitrarily chosen greedy one will be updated more rapidly resulting in a positive feedback loop.

3.4.3 Two-player Sliding Greedy Update

At times, if both players have Q-values at parity, the GHDS may comprise a *dual sliding system*. In the language of hybrid dynamical systems, this situation is equivalent to very low thresholds of switching between index states, meaning that no single flow describes the behavior in this regime. While some definable patterns show up during these periods, researchers in this field acknowledge the potential for unpredictable or chaotic behavior as $\alpha \to 0$ [Di Bernardo et al., 2008].

In some instances, the close distance between values can mean that decisions regarding how to implement continuous estimation can also affect long-run convergence, even for $\alpha \to 0$. There are several ways to define the idealized continuous version of Q-learning in this situation. For the rest of the analysis, we follow the convention of assuming discrete updates, but keep $\alpha \to 0$. This definition is consistent with the original setup and does not require new assumptions. It also recognizes that the two updates are always separate, even if values are equal. As a result of multiple sliding updates, a solution is no longer provided by solving a single hybrid system of differential equations,

Table 3.1: Payoffs for representative games in each subclass. RP's rewards are listed first in each pair.

Subclass 1a
Matching Pennies

MP	H	T
H	1, 0	0, 1
T	0, 1	1, 0

Subclass 1b
Spoiled Child

SC	B	M
S	1, 2	0, 3
P	0, 1	2, 0

Subclass 2a
Bach/Stravinsky

B/S	B	S
B	1, 2	0, 0
S	0, 0	2, 1

Subclass 2b
Chicken

CH	D	H
D	15, 15	1, 20
H	20, 1	0, 0

Subclass 3a
Deadlock

DL	b_1	b_2
a_1	1, 1	0, 3
a_2	3, 0	2, 2

Subclass 3b
Prisoner's Dilemma

PD	C	D
C	3, 3	0, 4
D	4, 0	1, 1

thereby complicating exact prediction of behavior. Fortunately, we are still able to clearly predict whether the system moves into a steady attractor for this particular GHDS (Sections 3.5 and 3.6).

3.5 Classes of 2-player 2-action games

The space of games can be divided according to characterizations of their equilibria and this section shows how IQL-ϵ behaves in each of these classes. For simplicity, we assume all reward values are distinct. (Ties can make games belong to multiple subclasses, complicating exposition.) Table 3.1 gives payoff matrices for some of the games we mention. The main results of this section are summed up by Table 3.2.

Subclass 1 covers games that only have a single mixed Nash equilibrium, meaning that the players play their actions with probabilities p and q such

that $0 < p, q < 1$. The space includes games that meet all of the following conditions: $(R_{11} - R_{21})(R_{12} - R_{22}) < 0$, $(C_{11} - C_{12})(C_{21} - C_{22}) < 0$, and $(R_{11} - R_{21})(C_{11} - C_{12}) < 0$. Zero-sum games like Matching Pennies (MP) are in this category, as is the new Spoiled Child (SC) game. Subclass 2 contains games that have two pure Nashes and one mixed Nash. These games satisfy the following conditions: $(R_{11} - R_{21})(R_{12} - R_{22}) < 0$, $(C_{11} - C_{12})(C_{21} - C_{22}) < 0$, and $(R_{11} - R_{21})(C_{11} - C_{12}) > 0$. Examples of games in this category include Bach/Stravinsky (B/S), Chicken (CH), and some Coordination games. Subclass 3 is the set of games in which at least one of the players has a pure dominant strategy, if $(R_{11} - R_{21})(R_{12} - R_{22}) > 0$ or $(C_{11} - C_{12})(C_{21} - C_{22}) > 0$. Examples in this class include all variants of Prisoner's Dilemma (PD) and Deadlock (DL).

The following results explicate the behavior of IQL-ϵ in these various classes, taking exploration into account. Some games that appear to be in one subclass can actually be changed into other classes once opponent exploration is considered, so we use those adjusted values for the analysis. For clarity, the analyses generally assume that the initial Q-values are their maximum possible values given the payoff matrix. It is common practice [Sutton and Barto, 1998] to initialize learning algorithms this way and it ensures that the algorithms play all actions greedily for some amount of time before settling down. IGA has its own classes based on the level of payoff sensitivity to the other player's strategy [Singh et al., 2000].

In each of these subclasses, IQL-ϵ further divides the space according to a simple rule so that on one side the algorithm always converges while on the other it is not guaranteed. Define Subclasses 1b, 2b, and 3b such that $\exists i, j$ $R_{ij} > R_N$ and $C_{ij} > C_N$, where R_N and C_N are either the unique expected Nash

Table 3.2: A summary of the behavior of IQL-ϵ and a taxonomy of games.

Subclass	**1a**	**1b**	**2a**	**2b**	**3a**	**3b**
# Pure Nashes	0	0	2	2	1	1
# Mixed Nashes	1	1	1	1	0	0
RP Action 2 is dominant?	No	No	No	No	Yes	Yes
$\exists i, j\ R_{ij} > R_N$ & $C_{ij} > C_N$?	No	Yes	No	Yes	No	Yes
Example game	MP	SC	B/S	CH	DL	PD
IQL-ϵ converges?	Yes	No	Yes	Y/N	Yes	Y/N

payoffs for RP and CP, or the lower security Nash payoffs in Subclass 2. Thus, there is some pure non-Nash strategy combination that is a higher payoff than the Nash equilibrium value for both players, much like the cooperative payoff in PD. While IGA gradually alters its strategy toward a best response, IQL-ϵ, in contrast, switches its greedy action suddenly, starving one value of updates. As a result, sometimes an action retains a relatively high value even when not a best response.

Theorem 2. *IQL-ϵ converges to the mixed Nash equilibrium when playing any game in Subclass 1a.*

Proof It is clear that no pure strategies will ultimately be successful because the other player can exploit any non-Nash strategy. IQL's Q-values converge to tied values where the ratio of actions selected matches that of the Nash equilibrium. In Q-learning, however, the agents can only approximate a mixed strategy by continually shifting their greedy actions. We can consider this behavior as a converged strategy because over short time spans the strategy is identical to the predicted Nash strategy, and the probability of each action is the same that the Nash strategy would play.

During learning, the values and strategies seem to cycle repeatedly at points removed from the mixed Nash, only to slowly approach it in the long

term. As the greedy strategy combination loops around the payoff table, exploration causes a slight convergence between the greedy and non-greedy value. On the next iteration, the values are closer to the Nash by some factor that depends on ϵ. In the limit, therefore, the policies close in on this equilibrium. □

Theorem 3. *In all Subclass 1b games, IQL-ϵ never converges to a single strategy, pure or mixed.*

Proof This result arises because the "cooperative" joint strategy that has higher values than the Nash equilibrium acts temporarily as a better attractor than the Nash whenever both of those actions are greedy. The Q-values periodically reset to the cooperative payoff, and the values for at least one of the players are always diverging from each other. The metaphor of a Spoiled Child (SC) illustrates the dynamics of this subclass, where the parent is RP and the child is CP. There is no pure Nash equilibrium in this class of games, so the IQL-ϵ players first drift towards playing the actions with a ratio resembling the mixed Nash strategy. As the values drop toward the value of this equilibrium, parent and child eventually greedily play the cooperative actions (Spoil and Behave, respectively). These values rise above the lower Nash values toward the cooperative payoff. However, this new attractor is not stable either, because the child would rather Misbehave, at which point the parent prefers the harmful Punish action. Thus, the system switches to a new set of dynamics and the cycle repeats. Unlike Subclass 1a, both greedy actions move away from the Nash during cooperation and therefore prevent convergence to a mixed strategy. □

Theorem 4. *IQL-ϵ converges to one of the two pure Nash equilibria in all games of Subclass 2a.*

Proof Consider the behavior of the dynamics once all Q-values have dropped below or are at the level of one of the Nash equilibria. At any point, the values move toward the payoffs resulting from the current greedy actions. This payoff either represents one of the pure Nashes or it does not. If it does, the greedy actions remain greedy, as neither player gets a higher value by exploring. If the greedy actions do not correspond to a pure Nash, then at some point one player switches greedy actions. The new combination is necessarily a Nash equilibrium by the payoff structure of this class. In addition, the mixed Nash is unstable because the dynamics of the payoff structure deviate from any mixed strategy to one of the pure strategies, returning to the earlier argument for convergence. □

Theorem 5. *IQL-ϵ may or may not converge to one of the two pure Nash equilibria in Subclass 2b.*

Proof While IQL-ϵ does not converge to the mixed Nash for reasons described above, some values lead the dynamics to one of the stable pure Nashes, while others cycle much like Subclass 1b. The key parameter for this class is ϵ, which can alter the payoff matrix and in essence put the game in a different class. □

3.6 Analysis of Convergence in Dominant Action Games

This section delves into the convergence behavior for Subclass 3 games, which have a dominant action for at least one player. Intuitively, this class seems the simplest—no matter what actions its opponent plays, a player always has an unchanging preferred response. In fact, IGA behaves this way by increasing its

dominant action probability until it reaches a Nash strategy. The gradient always points in the direction of the dominant action, and so no other behaviors are possible in the long run.

In the IQL-ϵ system, the effect of dominant actions depends on the method used to implement the learning algorithm. Under one regime where updates are applied with constant and discrete steps (however small), the dynamics can be unstable and lead to sudden shifts of fortune, or even chaotic behavior. The PD time series in Figures 3.1 and 3.2 show the strange, non-repeating pattern of updates possible in Subclass 3b, which persists at all learning rates and is an intriguing property of ϵ-greedy Q-learning.

However, in the infinitesimal, continuous case, we can show convergence to a cooperative, non-Nash equilibrium that stands as a counterpoint to the broad theoretical work that has been established showing convergence to the single-shot Nash equilibrium under a number of reinforcement learning algorithms [Bowling and Veloso, 2001].

3.6.1 Dominant Action Games: Subclass 3a

Call RP's action a_2 *dominant* when $R_{11} < R_{21}$ and $R_{12} < R_{22}$. If $\neg(\exists i, j\, R_{ij} > R_N$ and $C_{ij} > C_N)$, the game is a member of Subclass 3a.

Theorem 6. *In Subclass 3a games, IQL-ϵ converges to the pure Nash equilibrium identified by one player's dominant action and the other player's best response.*

Proof If there is no payoff preferable to the Nash for both players involving RP's non-dominant action a_1, it simply plays a_2. Once RP's Q-values drop below $\frac{R_{21}}{1-\gamma}$ or $\frac{R_{22}}{1-\gamma}$, no other payoff can attract the algorithm to play a_1.At that point, CP is faced with a static optimization problem and its values inevitably

converge to $\frac{C_{21}}{1-\gamma}$ and $\frac{C_{22}}{1-\gamma}$. Therefore, IQL-ϵ converges to the Nash equilibrium by definition. □

3.6.2 Cooperative Equilibrium Analysis

Define Subclass 3b as the remaining dominant action games, those for which $\exists i, j\ R_{ij} > R_N$ and $C_{ij} > C_N$). Prisoner's Dilemma resides in this subclass, which as we will see demonstrates special behavior owing to the unique properties at work in this game.

To investigate the convergence behavior of the infinitesimal Q-learning algorithm, we will make use of the concept of Lyapunov stability, which is the standard framework for working with dynamical systems [Lyapunov, 1992].

Definition 18. Lyapunov stability. *An equilibrium x^* is Lyapunov stable if, for some function $x(t)$, the derivative at this point with respect to time is zero: $x'(t) = \dot{x} = 0$. Furthermore, there is a neighborhood around x^* where for every ϵ, there is a $\delta = \delta(\epsilon) > 0$ so that if $||x'(0) - x^*|| < \delta$, then $\lim_{t\to\infty} ||x'(t) - x^*|| < \epsilon$. Conceptually, solutions that start close to the equilibrium, within any distance δ, will remain within ϵ of the equilibrium forever.*

First, let us establish that such an equilibrium Q^* exists in the IQL in IPD case, in which the probability of cooperation is over half. Then we will see that it is a Lyapunov stable equilibrium, when the Q-values for both players A and B are equal ($Q_C^A = Q_D^A = Q_C^B = Q_D^B$). Furthermore, when $Q_C^A = Q_D^A \neq Q_C^B = Q_D^B$ at time t, there exists a time $T > t$ where $Q_C^A = Q_D^A = Q_C^B = Q_D^B$. Finally, for all points in the four-dimensional space of Q-values, either the updates converge to this cooperative equilibrium or the mutual defection equilibrium.

Before proving the first conjecture, some useful observations and definitions should be made regarding identical algorithms in self-play.

Definition 19. Joint value update. *A joint value update is a change that is made to the component values of a joint action as a result of some reward per unit time the players take the joint action.*

If $Q(t)$ is the Q-value vector at time t, then $Q'(t) = \dot{Q}(t)$ is the differential equation that describes the joint update of the entire Q-value vector. Because the update equation depends on the entire vector, we need to use the joint value update.

According to the IQL setup, the players are taking actions and updating simultaneously and continuously. In this scenario, it is clear that their strategies depend on each other to the extent that the updates occur in the joint action space rather than independent of opponent actions, even as the values are separate for the two players.

Fact 1. Identical algorithms behave identically under identical conditions. *If two identical learning algorithms using the same update method have the same parameters (such as learning rate and exploration) as well as the same current state (in this case, Q-values) and receive the same inputs (rewards or payoffs for Q-learners), then they will behave identically as measured by strategy and update rate. That is, if A and B are identical learners and their payoffs are identical, and* $\forall a_i Q^A_{a_i} = Q^B_{a_i}$, *then* $\forall a_i \dot{Q}^A_{a_i} = \dot{Q}^B_{a_i}$.

This fact derives from the definition of update rate, which in this case is a set of differential equations. The implications of this fact is that when the values of two players with an identical payoff structure are equal to each other, the strategies and corresponding update rates are identical. For our purposes,

in this special case it means we can analyze how a single player behaves and conclude the same holds true for both players.

Before beginning to investigate convergence to stable points in this space, let us explore some additional properties about the updates that hold when $Q_C = Q_D$ for both players. In the context of hybrid dynamical systems, we need to be careful about our description of the dynamics at the boundary between two or more separate flows, especially when they point into this boundary (which we will consider a *stable boundary*). If one of the flows takes the values away from the boundary, then it serves only to change the direction of the flow. However, if along the boundary all updates flow into the boundary, then the values will remain at the boundary and we need a different way of thinking about the dynamics. One consistent way of thinking about the stable boundary case is that the flow updates are all simultaneously present, albeit with different relative frequencies.

Definition 20. Flow. *A flow $F(x)$ is the component of the differential equation governing the behavior of the dynamical system such that $\lim_{dt\to 0} x(t+dt) = x(t) + F(x(t))dt$. As such, it is a vector at point x.*

Definition 21. Stable boundary. *Let a hybrid dynamical system have a function B where the system operates according to flow F_1 when $B(x) < 0$ and F_2 when $B(x) > 0$. A stable boundary $B(x) = 0$ between two separate flows F_1 and F_2 exists when for every dt there exists some t such that $B(x(t)) < 0$ and $B(x(t) + F_1(x(t))dt) > 0$, and $B(x(t)) > 0$ and $B(x(t) + F_2(x(t))dt) < 0$. That is, a boundary is stable when the dynamical equations from each side of the boundary can lead, at the next point in time, to values on the other side.*

We know that B is a line and all flows F are vectors in this case, and so we can use standard vector math to understand the relative forces at a stable

boundary. To do so, we define the dynamics at a boundary that is a weighted combination of the flows of the adjacent regions.

Definition 22. Flow weight at a stable boundary. *When a system state is located at stable boundary B, then the* flow weight ω_i *is the relative contribution of flow F_i to the net vector F. Because $F = 0$ at a stable boundary, ω_i is inversely proportional to the magnitude of the flow component that is perpendicular to the boundary, G_i. If θ is the angle between $F_i(x)$ and $B(x)$ at x, and $G_i(x) = F_i(x) - |F_i(x)| \cos\theta$ is the orthogonal projection of $F_i(x)$ onto $B(x)$, then $\sum_i \omega_i |G_i(x)| = 0$ because the flows cancel in the direction orthogonal to B by definition of a stable boundary.*

In practice, this formula for ω is constructed by forming a column for each flow and a row for each dimension, as well as a final row to enforce the constraint that the weights sum to one. This formulation leads to the possibility that we get an undefined result. Intuitively, this can happen when there are multiple cases of flows canceling each other, and there is no way to assign the amount of cancelation to each opposing set.

In the single dimension case, consider a situation where $\dot{x} = 1$ when $x < 2$, and $\dot{x} = -2$ when $x > 2$. Then, $x = 2$ is a stable boundary (here, $B(x) = x - 2$). Because the system takes both updates simultaneously, they must balance in order for $\dot{x} = 0$ at $x = 2$. Therefore, the positive update is taken with frequency $\frac{2}{3}$ and the negative one is taken with frequency $\frac{1}{3}$. Under this definition, a boundary could be a line or hyperplane. It is also possible for the updates to slide along this boundary B, which is stable as long as the system does not enter one side or the other.

We can go further by extending this notion of stable boundary to a single point in an n-dimensional space.

Definition 23. Stable boundary equilibrium. *A stable boundary equilibrium x^**

between two or more regions with separate flows exists when the boundaries between all adjacent regions are stable boundaries, and along each boundary where $B(x) = 0$, $\lim_{t\to\infty} |x(t) - x^*| \to 0$.

Imagine a two-dimensional hybrid dynamical system where $F_1(x) = \dot{x_1} = 0, \dot{x_2} = 1$ when $x_2 < 0$; $F_2(x) = \dot{x_1} = -1, \dot{x_2} = -1$ when $x_1 > 0, x_2 > 0$; and $F_3(x) = \dot{x_1} = 1, \dot{x_2} = -1$ when $x_1 < 0, x_2 > 0$. Under these conditions, there are several stable boundaries but only one stable point at the origin. Intuitively, the flow points up below the x axis, down and to the left in quadrant I, and down and to the right in quadrant II. The system will move, possibly changing quadrants, until the origin is reached. Here, all the flows keep the system stuck here. In this case we have the following system of equations, corresponding to the two boundaries and the constraint that $\sum_i \omega_i$:

$$X_i^\delta = \begin{bmatrix} 0 & -1 & 1 \\ 1 & -1 & -1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

By solving for ω, we find that the weights on the flows at this stable point are $\omega_1 = \frac{1}{2}, \omega_2 = \frac{1}{4}, \omega_3 = \frac{1}{4}$.

When the state of a system is located at a boundary, then the fraction of time it operates according to each flow is inversely proportional to the magnitude of the flow component into its respective region. If this requirement did not hold, then there would be an imbalance that shifts the state away from the stable point. This result would be inconsistent with the constraints that arise as required to operate with region-specific flows.

Notice that if there is no combination of flows into a particular region, then the weighting for the flow corresponding to that region will be 0. Therefore, we can have situations where only two flows F_1, F_2 carry all the weight, even

when there are many more flows adjacent to x^* and F_1, F_2 are only adjacent at x^*.

The flows in our IPD/IQL setting correspond to the different joint updates that occur as a result of each possible payoff. Because of this correspondence, we can investigate how these dynamics work in this setting according to the general hybrid dynamical systems framework. Likewise, there is some distribution over the possible joint greedy updates that occurs as a result of the Q-learning algorithm in self-play. Let us call this update-frequency vector $X = [X_0, X_1, X_2, X_3]$ and allow for these values to be in the range $[0, 1]$ and sum to 1 such that X is a probability distribution.

Let us define the following values for the dynamics of IQL-ϵ in the Prisoner's Dilemma game.

- ϵ: the amount of exploration, taken here as the fraction of time spent playing the non-greedy action
- A, B: two players in this space
- R: the Reward payoff for mutual cooperation
- S: the Sucker payoff for cooperating against defection
- T: the Temptation payoff for defecting against cooperation
- P: the Punishment payoff for mutual defection
- Q_C: the Q-value for cooperation
- Q_D: the Q-value for defection
- Q: the single Q-value when $Q_C = Q_D$

- $\dot{Q}_C = \dot{Q}_{CC} + \dot{Q}_{CD}$: the Q-value update rate for cooperation
- $\dot{Q}_D = \dot{Q}_{DC} + \dot{Q}_{DD}$: the Q-value update rate for defection
- $\dot{Q}_{CC} = ((1-\epsilon)^2 + \frac{w}{c}\epsilon^2)(R-Q)$
- $\dot{Q}_{CD} = (\epsilon(1-\epsilon) + \frac{w}{c}\epsilon(1-\epsilon))(S-Q)$
- $\dot{Q}_{DC} = (\epsilon(1-\epsilon) + \frac{w}{c}\epsilon(1-\epsilon))(T-Q)$
- $\dot{Q}_{DD} = (\epsilon^2 + \frac{w}{c}(1-\epsilon)^2)(P-Q)$
- c: percent of updates occurring under mutual cooperation
- $w = 1-c$: percent of updates occurring under mutual defection

Although the dynamics technically take place in a four-dimensional space, it is perhaps more helpful to think of them in a reduced space of the differences in each player's values, so that $Q_C^A - Q_D^A$ is one dimension and $Q_C^B - Q_D^B$ is another. Then, the flows will change along each axis. At the origin, the differences are zero and we have *dual value parity*.

Definition 24. Dual value parity. *The situation that arises in Q-learners in self-play when $Q_C^A = Q_D^A$ and $Q_C^B = Q_D^B$ is a boundary equilibrium.*

Theorem 7. Existence of stable parity. *There exists some value Q such that the origin of this reduced space (where $Q_C^A = Q_D^A$ and $Q_C^B = Q_D^B$) is a stable boundary point, such as where $Q \geq (1-\epsilon)^2 R - \epsilon(1-\epsilon)T - \epsilon^2 P$.*

Proof: There are clearly values of $Q = Q_C = Q_D$ where $\dot{Q}_C - \dot{Q}_D = (1-\epsilon)((R-Q)(1-\epsilon) + (S-Q)\epsilon) - \epsilon((T-Q)(1-\epsilon) + (P-Q)\epsilon) \leq 0$ in the region where $Q_C > Q_D$ for both players. This inequality yields $Q \geq (1-$

$\epsilon)^2 R - \epsilon(1-\epsilon)T - \epsilon^2 P$. If this first inequality holds, then the reverse also does: $\dot{Q}_C - \dot{Q}_D \geq 0$ in the region where $Q_C < Q_D$.

Finally, when $Q_C^A < Q_D^A$ but $Q_C^B > Q_D^B$, then $\dot{Q}_C^B - \dot{Q}_D^B < 0$ and $\dot{Q}_C^A - \dot{Q}_D^A < 0$. In all three cases, the update vector points either towards the origin or towards a stable boundary that points towards the origin (such as when $Q_C^A < Q_D^A$ and $Q_C^B = Q_D^B$). □

There exists a stable equilibrium in terms of the four-dimensional Q-value space that is somewhat trivial if we do not allow for any updates of the non-greedy (or non-played) action. That is, there are two equilibrium points where the updates of all actions are zero: one where $Q = R$ and $c = 1$, and the other where $Q = P$ and $c = 0$. If we force updates due to exploration, then we need to recognize that there are two forms of greedy updating that can occur: mutual cooperation (say at rate 1) and mutual defection (at rate $\frac{w}{c}$), along with the necessary exploration. The reason we can use just these two weights for the four vectors is that all four of the updates push the dynamics either into the mutual cooperation space ($\dot{Q}_{CC}$ and $\dot{Q}_{DD}$) or the mutual defection space ($\dot{Q}_{CD}$ and $\dot{Q}_{DC}$), as described above.

As a result of these vectors and the definition of stable boundary point, we know that there will be no proportion of updates in the asymmetric joint action when $Q_C^A = Q_D^A = Q_C^B = Q_D^B$ because any linear combination of the updates lies along the $Q_C^A - Q_D^A = Q_C^B - Q_D^B$ line. Formally, the net update vector is balanced by using just these two update components, and there is no combination of these other vectors that creates a net vector into the asymmetric regions, which according to the definition of a stable boundary equilibrium eliminates the possibility that time is spent in this region due solely to the dynamics. On the other hand, exploration allows this update to occur by artificially adding

this update with weight ϵ.

Theorem 8. *With exploration, there are two equilibrium points, such that* $\epsilon << c < 1 - \epsilon$ *and* $c = 0$.

Proof: First, we know that if the Q-values are different, then $c = 0$ is a stable point because $Q_C < Q_D$ when $\dot{Q}_C = \dot{Q}_D = 0$.

To investigate the claim for $c > 0$, we use the update rates for every payoff and action, taking into account that some proportion of the updates will come from the mutual cooperation state, and some from mutual defection, with exploration updates coming from both.

The Q_C value will be stable when $\dot{Q}_C = 0$. Set $S = 0$ without loss of generality.

$$\begin{aligned} \dot{Q}_C &= 0 = ((1-\epsilon)^2 + \frac{w}{c}\epsilon^2)(R - Q) - (\epsilon(1-\epsilon) + \frac{w}{c}\epsilon(1-\epsilon))Q \\ Q_C^* &= \frac{((1-\epsilon)^2 + \frac{w}{c}\epsilon^2)R}{\frac{w}{c}\epsilon + 1 - \epsilon} \end{aligned}$$

Because these are the updates for the Q-values given received payoffs, the Q-value for C will remain constant at the level Q_C^*.

Now let us turn to the Q_D value.

$$\begin{aligned} \dot{Q}_D &= 0 = (\epsilon(1-\epsilon) + \frac{w}{c}\epsilon(1-\epsilon))(T - Q) + ((1-\epsilon)^2 + \frac{w}{c}(1-\epsilon)^2)(P - Q) \\ Q_D^* &= \frac{\epsilon(1-\epsilon)(1+\frac{w}{c})T + (\epsilon^2 + (1-\epsilon)^2\frac{w}{c})P}{(1-\epsilon)\frac{w}{c} + \epsilon^2} \end{aligned}$$

Setting these two values $Q_C^* = Q_D^* = Q^*$, we arrive at a quadratic equation for $\frac{w}{c}$:

$$\begin{aligned} 0 = {} & (\epsilon(1-\epsilon)^2 P + \epsilon^2(1-\epsilon)(T-R))(\frac{w}{c})^2 + \\ & ((\epsilon^3 + (1-\epsilon)^3)(P-R) + \epsilon(1-\epsilon)T)\frac{w}{c} + \\ & \epsilon(1-\epsilon)^2(T-R) + \epsilon^2(1-\epsilon)P \end{aligned}$$

And solving for $\frac{w}{c}$ we arrive at two solutions, but only the one where $\frac{w}{c} < 1$ is stable. The resulting strategy is mostly cooperation. □

This theorem tells us that there are two equilibria with $c > 0$. The higher one with $c_H > w$ gives us Q_H^*, and the lower c_L gives Q_L^*. However, we can now show that Q_H^* is stable while Q_L^* is not, under the definitions of Lyapunov stability. Because $Q_L^* < \frac{1}{1-2\epsilon}[(1-\epsilon)((1-\epsilon)P + \epsilon T) - \epsilon((1-\epsilon)S + \epsilon R)]$, greedy defection is a viable strategy and the cooperation values can decrease due solely to exploration, as explained above.

To complete the validation that there is a zone of initial Q-values that will end up in the cooperation equilibrium, we need to examine several cases. The first is when both players' values are the same, but not at the equilibrium. The second case considers what happens when each player's values are different, and not at the stable point. The final possible starting point is where all four values are different. If convergence is inevitable within some range of values, then the updates in each of these cases must, over time, move the Q-values towards the equilibrium.

For the first part of the analysis, we will use the observation that for $Q > Q_L^*$, if the Q-values for a player are equal, then either the players will cooperate greedily ($\dot{Q}_C > 0$) or the values will remain equal due to Theorem 7 regarding update dynamics. That means that for each value Q, either $c = 1$ or there is a specific joint strategy $c < 1$ that keeps $\dot{Q}_C = \dot{Q}_D$. When $Q_C < \frac{1}{1-2\epsilon}((1-\epsilon)^2 R - \epsilon(1-\epsilon)T - \epsilon^2 P)$, we expect that the values will not remain equal because of the greedy update. Therefore, we only need to look at the complementary cases, when $Q_C \geq \frac{1}{1-2\epsilon}((1-\epsilon)^2 R - \epsilon(1-\epsilon)T - \epsilon^2 P)$.

Lemma 1. *When all four Q-values are equal to Q and $Q \geq \frac{1}{1-2\epsilon}((1-\epsilon)^2 R - \epsilon(1-\epsilon)T - \epsilon^2 P)$ and $Q < Q_H^*$, the updates $\dot{Q}_C$ and $\dot{Q}_D$ are positive.*

Proof: The values of the updates are set by the ratio of the outcome frequencies c and $w = 1 - c$ and the value Q.

Recalling the updates for each outcome:

- $\dot{Q}_{CC} = (c(1-\epsilon)^2 + (1-c)\epsilon^2)(R - Q)$
- $\dot{Q}_{CD} = -(c\epsilon(1-\epsilon) + (1-c)\epsilon(1-\epsilon))Q$
- $\dot{Q}_{DC} = (c\epsilon(1-\epsilon) + (1-c)\epsilon(1-\epsilon))(T - Q)$
- $\dot{Q}_{DD} = (c\epsilon^2 + (1-c)(1-\epsilon)^2)(P - Q)$.

We set $\dot{Q}_C = \dot{Q}_D$ to get c:

$$\begin{aligned}(c(1-\epsilon)^2 + (1-c)\epsilon^2)(R-Q) &= \epsilon(1-\epsilon)T + (c\epsilon^2 + (1-c)(1-\epsilon)^2)(P-Q) \\ c &= \frac{-\epsilon^2 R + \epsilon(1-\epsilon)T + (1-\epsilon)^2 P - (1-2\epsilon)Q}{(1-2\epsilon)(R+P) - 2(1-2\epsilon)Q}.\end{aligned}$$

The numerator and denominator are negative when $Q \approx R$. When $Q < Q^*_H$, then $c > c^*_H$ which implies $\frac{c}{\epsilon(1-\epsilon)} > \frac{c^*_H}{\epsilon(1-\epsilon)} > 1$. Therefore, $\frac{c}{\epsilon(1-\epsilon)}(R - Q) + S - Q > \frac{c^*_H}{\epsilon(1-\epsilon)}(R - Q^*_H) + S - Q^*_H = 0$, and so $\dot{Q}_D = \dot{Q}_C > 0$. □

Lemma 2. *When all four Q-values are equal to Q and $Q > Q^*_H$, the updates $\dot{Q}_C$ and $\dot{Q}_D$ are negative.*

Proof: For the same reason as in the above lemma, $c < c^*_H$ when $Q > Q^*_H$. As a result, $1 < \frac{c}{\epsilon(1-\epsilon)} < \frac{c^*_H}{\epsilon(1-\epsilon)}$. Therefore, $\frac{c}{\epsilon(1-\epsilon)}(R - Q) + S - Q < \frac{c^*_H}{\epsilon(1-\epsilon)}(R - Q^*_H) + S - Q^*_H = 0$, and so $\dot{Q}_D = \dot{Q}_C < 0$. □

A more in-depth analysis is required when the players' values are not the same as each other, because they will have different updates and possibly different strategies as a result.

Theorem 9. *When $Q^A_C = Q^A_D = Q^A$ for player A and $Q^B_C = Q^B_D = Q^B$ for player B and $Q^A > Q^*_H$ and $Q^B < Q^A$, $\lim_{t\to\infty} |Q^A(t) - Q^B(t)| = 0$.*

Proof: There are two cases here. One where $Q^B > Q^*_H$ and one where $Q^B \leq Q^*_H$. Let us start with the first case. For reasons similar to those described above in Lemma 2, the updates of both players will be different but negative. This situation persists until $Q^B \leq Q^*_H$, to which we now turn.

While the amount of update coming from mutual cooperation and mutual defect will be the same for both players, we must now allow for a divergence in strategy between the asymmetric outcomes. The reason is that a linear combination of the joint update vectors can possibly add to a vector located in the value space where the asymmetric joint action is greedy. To account for this possibility, we will add a defection/cooperation factor μ to each set of updates.

Recalling the updates for each outcome:

- $\dot{Q^A}_{CC} = (c(1-\epsilon)^2 + (1-c)\epsilon^2)(R - Q^A)$
- $\dot{Q^A}_{CD} = -(c\epsilon(1-\epsilon) + (1-c)\epsilon(1-\epsilon) + \mu)Q^A$
- $\dot{Q^A}_{DC} = (c\epsilon(1-\epsilon) + (1-c)\epsilon(1-\epsilon))(T - Q^A)$
- $\dot{Q^A}_{DD} = (c\epsilon^2 + (1-c)(1-\epsilon)^2)(P - Q^A)$.
- $\dot{Q^B}_{CC} = (c(1-\epsilon)^2 + (1-c)\epsilon^2)(R - Q^B)$
- $\dot{Q^B}_{CD} = -(c\epsilon(1-\epsilon) + (1-c)\epsilon(1-\epsilon))Q^B$
- $\dot{Q^B}_{DC} = (c\epsilon(1-\epsilon) + (1-c)\epsilon(1-\epsilon) + \mu)(T - Q^B)$
- $\dot{Q^B}_{DD} = (c\epsilon^2 + (1-c)(1-\epsilon)^2)(P - Q^B)$.

We have

$$c^A = \frac{-\epsilon^2 R + \epsilon(1-\epsilon)T + (1-\epsilon)^2 P - (1-2\epsilon)Q^A - \mu)Q^A}{(1-2\epsilon)(R+P) - 2(1-2\epsilon)Q^A}$$

and

$$c^B = \frac{-\epsilon^2 R + \epsilon(1-\epsilon)T + (1-\epsilon)^2 P - (1-2\epsilon)Q^B + \mu(T-Q^B)}{(1-2\epsilon)(R+P) - 2(1-2\epsilon)Q^B}$$

Setting $c^A = c^B$:

$$\begin{aligned}
N^A &= -\epsilon^2 R + \epsilon(1-\epsilon)T + (1-\epsilon)^2 P - (1-2\epsilon)Q^A \\
N^B &= -\epsilon^2 R + \epsilon(1-\epsilon)T + (1-\epsilon)^2 P - (1-2\epsilon)Q^B \\
D^A &= (1-2\epsilon)(R+P) - 2(1-2\epsilon)Q^A \\
D^B &= (1-2\epsilon)(R+P) - 2(1-2\epsilon)Q^B \\
\frac{N^A - \mu Q^A}{D^A} &= \frac{N^B + \mu(T-Q^B)}{D^B} \\
D^B(N^A - \mu Q^A) &= D^A(N^B + \mu(T-Q^B)) \\
\mu &= \frac{D^B N^A - D^A N^B}{D^A(T-Q^B) + D^B Q^A}
\end{aligned}$$

Notice that this last equation is true in general, so that when $Q^A = Q^B, \mu = 0$. When $Q^B \approx R$, $D^A < N^A < 0$ and $D^B < N^B < 0$. We know that $D^B N^A - D^A N^B < 0$ because:

$$\begin{aligned}
0 &> D^B N^A - D^A N^B \\
0 &> ((N^A + (1-2\epsilon)Q^A) - (N^B + (1-2\epsilon)Q^B))(1-2\epsilon)(R+P) + \\
&\quad (2(1-2\epsilon)^2 - 2(1-2\epsilon)^2)Q^A Q^B + (1-2\epsilon)^2(R+P)(Q^B - Q^A) + \\
&\quad 2(1-2\epsilon)(N^A + (1-2\epsilon)Q^A)(Q^A - Q^B) \\
0 &> (2(1-2\epsilon)(N^A + (1-2\epsilon)Q^A) - (1-2\epsilon)^2(R+P))(Q^A - Q^B)
\end{aligned}$$

For $P << R$, the factor in front of $(Q^A - Q^B)$ is negative, and $Q^A - Q^B$ is positive. Since the denominator $D^A(T-Q^B) + D^B Q^A$ is also negative, then $\mu > 0$, suggesting that the player who has the lower Q-values (in this case, B)

will be the one who defects more in order to keep parity. As a result, $\dot{Q^B} > \dot{Q^A}$ and convergence will occur because $\forall t, |Q^A(t-1) - Q^B(t-1)| > |Q^A(t) - Q^B(t)|$. □

Theorem 10. *When $Q_C^A \neq Q_D^A$ or $Q_C^B \neq Q_D^B$ at time t, then at some time $t+\tau$ $Q_C^A < Q_D^A$ and $Q_C^B < Q_D^B$ or $Q_C^A = Q_D^A$ and $Q_C^B = Q_D^B$, the* dual parity *condition.*

Proof: The inequalities will be true in the case when the Q-values converge to the greedy mutual defection outcome.

Otherwise, to show how the dual parity happens, we must examine each starting configuration of the Q-values. The possible cases are when $Q_C^A < Q_D^A$ and $Q_C^B < Q_D^B$, $Q_C^A < Q_D^A$ and $Q_C^B > Q_D^B$, or $Q_C^A > Q_D^A$ and $Q_C^B > Q_D^B$. In the first case, either $Q_D^A, Q_D^B < P$ or $\dot{Q_D^A} < 0$ or $\dot{Q_D^B} < 0$. If $Q_D^A < P$ and $Q_D^B < P$ then convergence to mutual defection is assured. If $\dot{Q_D^A} < 0$ or $\dot{Q_D^B} < 0$, one of two outcomes is possible. Either $Q_D^A < P$ and $Q_D^B < P$ will hold at some point, or the updating enters a second phase when $Q_C^A = Q_D^A$. At this point either $\dot{Q_D^B} > \dot{Q_C^B}$ or $\dot{Q_D^B} < \dot{Q_C^B}$ due to the onset of the sliding update. As long as the first inequality holds, then $\dot{Q_D^B} \neq \dot{Q_C^B}$ and mutual defection is again assured. If the second inequality occurs, then at some point the updating enters a third phase such that $Q_C^B = Q_D^B$, reaching dual parity.

The second starting configuration represents the defection vs. cooperation arrangement, and will lead to $Q_C^B = Q_D^B$ because $\dot{Q_D^B} > \dot{Q_C^B}$. Then, this situation will be the same as the second phase in the situation above, and the same analysis holds.

In the third starting configuration, mutual cooperation is the current update regime. Eventually, $\dot{Q_D^A} > \dot{Q_C^A}$ which causes $Q_C^A < Q_D^A$. The dynamics then transition into a phase that is the same as the second starting configuration, and the analysis is identical as before. □

Theorem 11. *Given an initial value setting of $Q_C^A, Q_D^A, Q_C^B, Q_D^B$, the update dynamics eventually converge to either $Q_C^A = Q_C^B < Q_D^A = Q_D^B = (1-\epsilon)P + \epsilon T$ or to $Q_C^A = Q_C^B = Q_D^A = Q_D^B = Q_H^*$ where Q_H^* is the value at the mutual cooperation equilibrium described above.*

Proof: We have seen that every starting configuration either reaches the mutual defection equilibrium at $Q_C^A = Q_C^B < Q_D^A = Q_D^B = (1-\epsilon)P + \epsilon T$ or it ends at the dual parity condition such that $Q_C^A = Q_D^A$ and $Q_C^B = Q_D^B$.

When dual parity is achieved, the updates eventually result in mutual defection or $Q_C^A = Q_D^A = Q_C^B = Q_D^B$. From there, the dynamics converge to Q_H^* due to Lyapunov stability. □

The final theorem encapsulates the prior theorems to show how convergence to a cooperative equilibrium is not only possible, but guaranteed in a large fraction of the total space of initial Q-value settings. This surprising and unconventional result demonstrates that there is still much to be explored when it comes to understanding the behavior of multiagent learning algorithms, even in a game as well-known as the Prisoner's Dilemma. The nonlinear dynamics of algorithms like IQL allow for multiagent researchers to break new ground both in theory and in empirical studies. The next section expands on how the algorithm behaves in practice under a typical set of conditions and implementation choices. This behavior can be radically different from the convergence demonstrated by the idealized algorithm.

3.6.3 Prisoner's Dilemma Phases

In this Prisoner's Dilemma subclass, simulations may not converge for numerical reasons owing to discrete updates occurring within a computational

system. We will explore the reasons why this phenomenon arises next. Under these conditions, the convergence result shown above does not hold, due to the nonlinear updating that occurs that leads to a greedy action no matter what the difference between Q_C and Q_D. As a result, we do not see the dual parity conditions described in the previous section.

Instead, the nonlinear operator causes unexpected dynamics to proceed according to a predictable sequence of update *phases* (Figure 3.2 and Table 3.3). Each phase represents a temporarily stable combination of index states and flows that persist until conditions change. The phases arise from a conflict between the selfish incentive of the dominant action and the mutual benefit of cooperative action. In a pattern similar to human conflict, the flows transition from I, *peaceful cooperation*, to II, *aggression*, to III, *domination*, to IV, *rebellion*, then either back to I or to DD, total war. One player's act of *aggression* breaks the peace so that one player's defection values gain advantage, while the *rebellion* results from the reaction of the long-dominated player who will not be exploited any longer. These repeated phases form a chaotic attractor due to the dual sliding update (Figure 3.3).

1. *Peaceful Cooperation:* $Q_{a_1} > Q_{a_2}$ and $Q_{b_1} > Q_{b_2}$. Starting from low values, both players may happen to find themselves playing C greedily. When this condition arises, C's Q-value updates $(\dot{Q}_{a_1}, \dot{Q}_{b_1})$ will be higher than those for defections $(\dot{Q}_{a_2}, \dot{Q}_{b_2})$. Over time as Q_{a_1} rises, $\dot{Q}_{a_1}$ will fall until it is under $\dot{Q}_{a_2}$, even when D is non-greedy. This phase ends when $Q_{a_1} < Q_{a_2}$.

Consider that RP is the aggressor who initiates defecting so that $Q_{a_2} > Q_{b_2}$.

2. *Aggression:* $Q_{a_1} < Q_{a_2}$ and $Q_{b_1} \geq Q_{b_2}$. Although this phase says little about which player's actual values are higher, we use RP to refer to the player who is more apt to defect and CP to describe the player closer to its Nash

values. The algorithm's behavior as a dynamical system ensures that the two players will be different within this class of games. When RP's dominant action becomes greedy, for a time it will face CP's greedy C until one player's values become equal again. Usually Q_{a_2} will rise until $Q_{b_1} = Q_{b_2}$, and then start to fall. This phase ends when $Q_{a_1} \leq Q_{a_2}$ again.

3. *Domination:* $Q_{a_1} = Q_{a_2}$ and $Q_{b_1} = Q_{b_2}$, and $\dot{Q}_{a_2} \geq \dot{Q}_{a_1}$. During this period, RP will gain nothing from cooperating further, allowing it to exploit with its dominant action to some degree. As a result, CP's values enter a slow decline. While the dynamics during the other observed phases can be easily described, the behavior here does not appear to fit a fixed set of equations. Instead, a complex set of forces keeps each player's values together. The unit of update "step" during this phase is actually a sequence of several other greedy phases resulting in a return to parity for both players. If either player's D action becomes greedy, the Q-value for D will drop against the other player's D, or cause the other player's C to drop until it plays D. The more complicated step arises when both players play C. The values of both players rise, but CP rises faster. Because $\dot{Q}_{a_2} \geq \dot{Q}_{a_1}, Q_{a_2}$ will catch up and become greedy. When that happens, $\dot{Q}_{b_1}$ becomes negative and Q_{b_1} drops to equal Q_{b_2}. Next, $\dot{Q}_{a_2}$ becomes negative and RP's values will be equal again. This period ends when CP drops far enough that it is able to play its dominant action greedily against RP.

There are two important things to notice about this chain of events. One is that RP's values may end either higher or lower at the end of each miniature sequence of updates, but CP's values will definitely be lower. The second observation is that any difference between separate starting points for CP's value will be magnified upon rejoining the values. A lower starting value for

CP gives RP more time to defect, and therefore will accelerate the drop in the value.

4. *Rebellion:* $Q_{a_1} = Q_{a_2}$ and $Q_{b_1} < Q_{b_2}$ and $\dot{Q}_{b_1} < \dot{Q}_{b_2}$. Figure 3.2(b) shows the dynamics of this phase. At a certain point, CP's greedy defection update $\dot{Q}_{b_2}$ will be higher than its non-greedy cooperation update $\dot{Q}_{b_2}$ against RP's mixed strategy ϕ_{22}. During this phase, RP's values drop rapidly, and accordingly ϕ_{22} will slowly rise. In some cases, Q_{b_1} will become greedy along with Q_{a_1}, thus returning the dynamics to phase I before Q_{a_1} drops far enough to cause the victim to switch. In other cases, this phase resumes with the players switching roles. The payoffs and exploration parameter determine the behavior of this phase, deciding whether the dynamics end in convergence or endless repetition of these phases. The conditions for convergence are explored further in the next section.

3.6.4 Conditions for Convergence in Subclass 3b

For the dynamics to converge to the Nash equilibrium, one of the players (CP, for instance) must sustain the dominant action greedily against *both* actions of RP so RP's values can fall. Figure 3.3(inset) shows an example of this condition in phase IV. To keep the values decreasing to the Nash point, the players must switch roles before both cooperative actions become greedy again, thereby perpetuating phase IV. Once one of the players can defect greedily against the other's greedy defection, convergence to Nash is assured. The value below which mutual defection (DD) is inevitable is the following threshold Q_{DD}, found when (non-greedy update) $\dot{Q}_{\hat{b_1}}$ is less than (greedy update) $\dot{Q_{b_2^*}}$:

$$\frac{\epsilon}{2}(C_{21} + (\gamma - 1)Q_{DD}) < (1 - \frac{\epsilon}{2})(C_{22} + (\gamma - 1)Q_{DD})$$

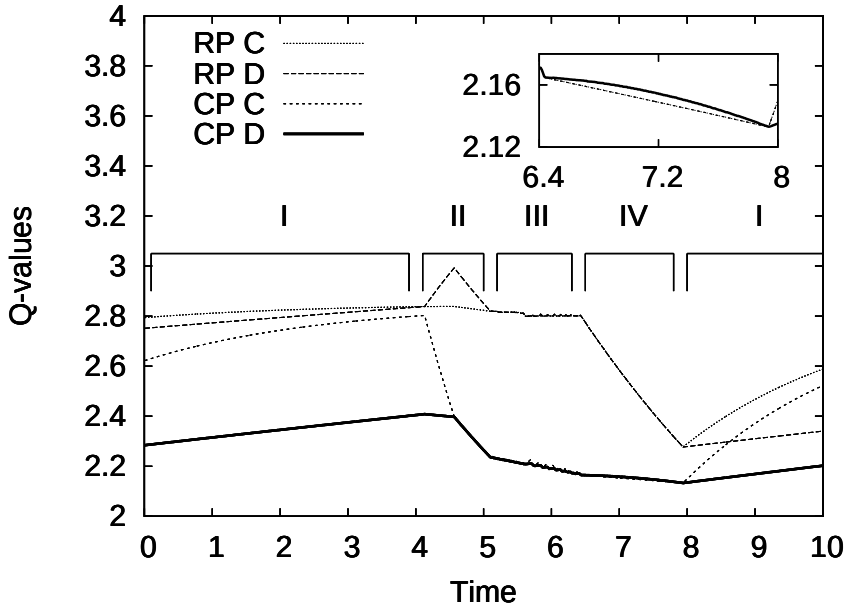


Figure 3.2: The sequence of phases I-IV during PD with IQL-ϵ agents. The value ordering is documented in Table 3.3. Some of these phases exist entirely within a single index state of the GHDS (I), while others rotate between all four index states (III). In the **peaceful cooperation** phase I, both agents cooperate greedily. Eventually, via exploration, the defection value appeals to one of the players, RP, leading to **aggression** (II). In II, RP forces both of CP's values to drop until neither player has a clear greedy action. Phase III, **domination**, is the dual sliding update, so that the algorithm alternates between mutual cooperation and one player defecting. When CP's values drop below the Q_{DT} threshold, it becomes profitable to defect against both actions of the other player, initiating **rebellion** (IV). After this final phase, both players reenter peaceful phase I, thereby renewing the cycle. (inset) Close-up of phase IV of the cycle.

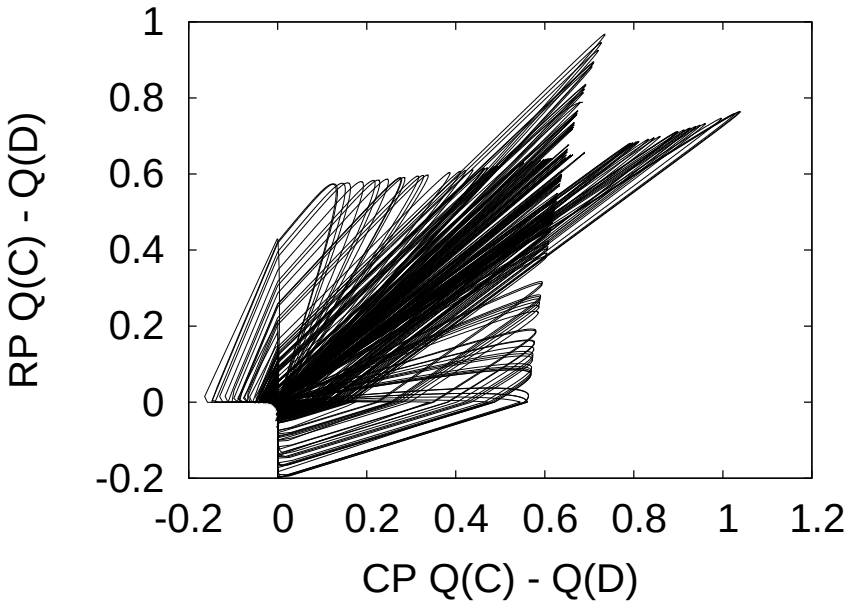


Figure 3.3: A 2-D projection of the chaotic attractor over many cycles in the IQL phase space. The system switches flows along the x=0 and y=0 axes. Note how at these values the system avoids the unstable mutual defection flow in the lower left quadrant. However, there are some updates made from the flows in this region.

Table 3.3: Properties of phase dynamics during repeated Prisoner's Dilemma by phase. These phases repeat in the order identified in Figure 3.2. (Arrows denote transitions.)

Comparison	I	II	III	IV
RP Value Q_{a_1} ? Q_{a_2}	$>$	$<$	$=$	$=$
CP Value Q_{b_1} ? Q_{b_2}	$>\rightarrow<$	$<\rightarrow=$	$=$	$<$
RP Update $\dot{Q}_{a_1}$? $\dot{Q}_{a_2}$	$>\rightarrow<$	$<\rightarrow>$	$\leq$	$=$
CP Update $\dot{Q}_{b_1}$? $\dot{Q}_{b_2}$	$>$	$<\rightarrow=$	$=$	$<\rightarrow>$

$$Q_{DD} < \frac{-\frac{\epsilon}{2}C_{21} + (1-\frac{\epsilon}{2})C_{22}}{(1-\gamma)(1-\epsilon)}.$$

As CP's values decrease during phase III, they drop below a defection threshold (DT) where exploring $Q_{\hat{b_1}}$ drops faster than greedy $Q_{b_2^*}$. In this case b_2^*, D, is greedy in response to mixed actions of RP. Say $C_{\phi 22}$ is the D reward against RP's sliding update. Like above, Q_{DT} is defined by the inequality $\dot{Q}_{\hat{b_1}} < \dot{Q}_{b_2^*}$:

$$Q_{DT} \quad < \quad \frac{-\frac{\epsilon}{2}C_{\phi 21} + (1-\frac{\epsilon}{2})C_{\phi 22}}{(1-\gamma)(1-\epsilon)}.$$

These dynamics from Section 3.4.2 imply ϕ_{22} is equivalent to the percentage of time that RP spends playing a_2 when its dropping values are equal and CP is playing b_2 greedily. In general, ϕ_{22} rises as values decrease. An important cooperation threshold (CT), Q_{CT}, relates to the level where $\dot{Q}_{\hat{a_2}} > \dot{Q}_{a_1^*}$. Essentially, if both of a player's values are very close and above Q_{CT}, it cannot cooperate for long before $Q_{\hat{a_2}}$ overtakes $Q_{a_1^*}$:

$$\frac{\epsilon}{2}((\gamma-1)Q_{CT} + R_{21}) \geq (1-\frac{\epsilon}{2})((\gamma-1)Q_{CT} + R_{11})$$

$$Q_{CT} \geq \frac{(1-\frac{\epsilon}{2})R_{11} - \frac{\epsilon}{2}R_{21}}{(1-\epsilon)(1-\gamma)}.$$

As long as $Q_{CT} \leq Q_{DT}$ for some player, then convergence to the Nash equilibrium is assured because it has nothing to lose by defecting. If this condition is true for long enough, the other player may be in a position to trigger a chain of defections leading to the Nash.

Phase IV, observed in the closeup Figure 3.2(inset), either leads to convergence or back to I, depending on its length and its values when it commences. IV begins when CP plays D greedily against greedy C below the threshold Q_{DT}, thereby dropping RP's values. If phase IV begins with Q_{b_2} just below Q_{DT}, then it will be too short and convergence cannot happen, as the flow returns to peaceful cooperation and the cycle restarts. However, IV might not begin as soon as it crosses the threshold if the possibility of transitioning to the crucial index state is zero, regardless of the continuity of the updates. Delaying phase IV makes CP eventually defect for longer periods, increasing the likelihood of convergence to Nash. In the case of PD, this question is settled during phase III, the dual sliding update. To illustrate this process, consider what happens for discrete updates, shown in Figure 3.4. Essentially, CP must first erase RP's gains made when RP defected against its C with two Ds. After two defections, RP cooperates, but now so does CP, so phase III continues. This canceling still occurs at an arbitrarily small scale, but the updating is not continuous in the sense that the flow in each region does not operate until the boundary. Instead discrete updates are made.

In single-agent Q-learning, varying the learning rate α has no effect on convergence, but it can make a big difference in the multiagent case of Prisoner's Dilemma. A uniform (unchanging) learning rate might delay phase IV from its onset if the index state essential to causing a transition to the next phase (here, a CP greedy defection and RP greedy cooperation) is skipped again and again. This effect occurs for uniform continuous updating as well, as mutual defection makes both defection values non-greedy, delaying phase IV. From an empirical perspective, this effect also appears while estimating the continuous trajectories of IQL-ϵ, where one typically computes small updates in the

Q-values. One way to alter this outcome is to vary the size of α stochastically to approximate noisy continuous updating, thus ensuring every index state a chance to be visited. However, given that uniform (unchanging) learning rate IQL is the standard way to implement Q-learning, its behavior should be fully documented.

Some games prevent the onset of IV below defection threshold Q_{DT} until ϕ_{22} rises above its own threshold. Specifically, the condition just described implies that phase IV can begin when

$$\phi_{22} \geq \frac{-\frac{R_{22}}{1-\gamma} + Q_{a_2}}{-\frac{R_{22}}{1-\gamma} + Q_{a_2} + \frac{R_{21}}{1-\gamma} - Q_{a_2}} = \frac{-R_{22} + (1-\gamma)Q_{a_2}}{R_{21} - R_{22}}$$

Once it is known where phase IV must begin as $\alpha \to 0$, one iteration is enough to show whether the system converges. In Figure 3.5, we have mapped the region of symmetric games where uniform IQL-ϵ does or does not converge to Nash.

Theorem 12. *In Subclass 3b games, certain starting values guarantee the IQL-ϵ dynamics converge to the pure Nash. For other values, the dynamics do not permanently converge to any particular strategy, but average rewards of both players are higher than the Nash.*

Proof There exists a defection threshold Q_{DD} below which two-player greedy defection ($F_{a_2^* b_2^*}$) is a sink and does not jump to another index state. Starting values that meet this condition, or lead later to values that do, converge to the Nash. In addition, a high R_{21} value that delays phase IV while $Q_{DT} \geq Q_{CT}$ encourages convergence as RP can defect freely.

Other starting values enter a series of phases. If phase IV always occurs immediately after the Q-values drop below Q_{DT}, mutual cooperation ($F_{a_1^* b_1^*}$) temporarily attracts the cooperation Q-values away from dominant action a_2

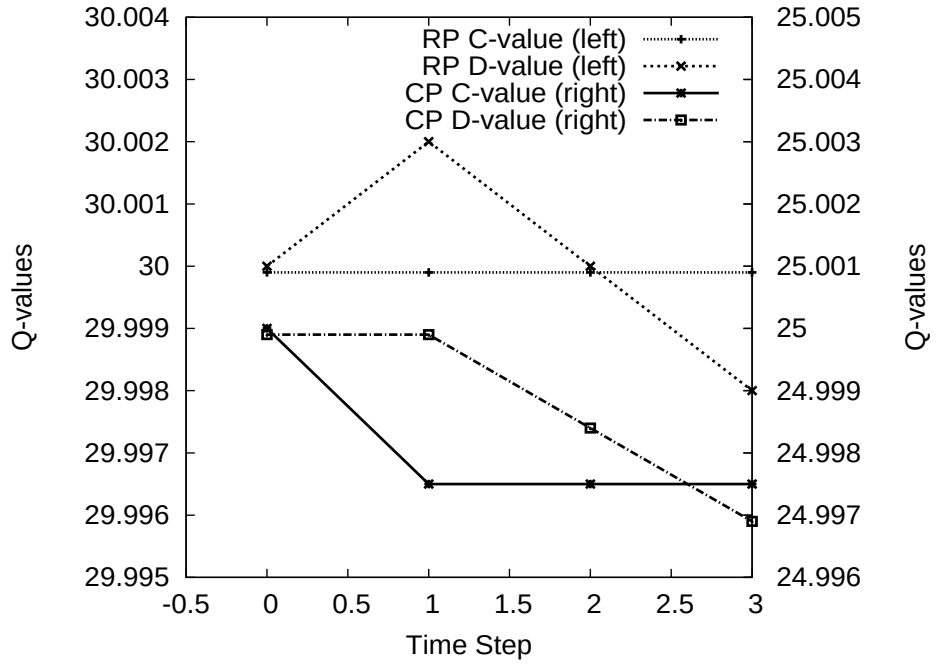


Figure 3.4: This plot demonstrates how greedy defections cancel each other out to prevent CP (bottom) from defecting against RP's cooperation (top), even as $\alpha \to 0$. Imagine that CP's values are below the threshold Q_{DT} and RP's values are close together. CP first erases the gains made when RP defected against its cooperation (the first update pictured), with two defections. After two defections, shown in the next two updates, RP cooperates. However, if $\phi_{22} < \frac{2}{3}$ as it is here, CP always cooperates because it cannot defect more than 2 times in a row. Therefore, phase IV does not begin until $\phi_{22} \geq \frac{2}{3}$. This sequence of events still takes place at infinitesimal scale.

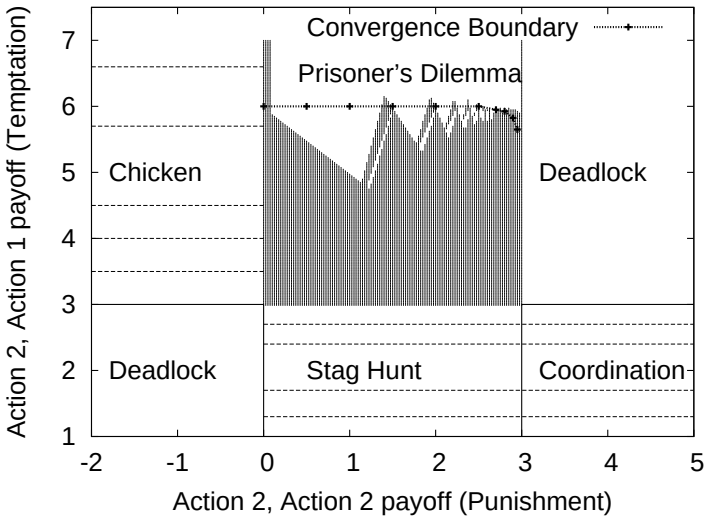


Figure 3.5: The shaded area marks the space of symmetric games where uniform IQL-ϵ does not converge. x is the payoff when both agents play Action 2, and y is the RP payoff when RP plays a_2 and CP b_1. The a_1, b_1 payoff is fixed at 3 for both players and the a_1, b_2 payoff is 0. The space divides into several well-known games with pure strategies that attract IQL-ϵ. The dashed lines mark the regions with two attracting equilibria, and plain white for one. The dotted line in the PD region represents a high temptation boundary above which noisy continuous IQL-ϵ always converges. The notched boundary in the PD section arises from the value of ϕ_{22} that initiates phase IV. The first notch includes those games where it takes at least two CP defections to undo RP's gains from CP cooperation, so that $\phi_{22} \geq \frac{2}{3}$ is required to trigger phase IV. The second notch contains those games where $\phi_{22} \geq \frac{3}{4}$ is needed to start phase IV, and so on.

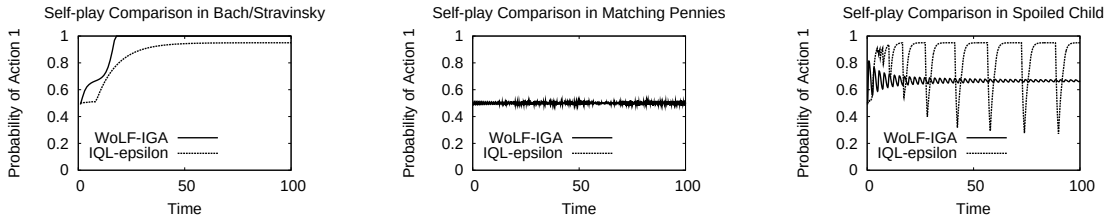


Figure 3.6: Action probabilities for RP in two self-playing algorithms in representative games (Table 3.1). The policies of WoLF-IGA converge, while the IQL-ϵ dynamics do not for some starting values in the PD or SC games. Both agents converge to one of the pure Nashes in B/S, and the mixed Nash in MP. In SC, the IQL-ϵ players oscillate periodically while WoLF-IGA reaches the fixed point defined by the mixed Nash of the game. See Section 3.7 for more details.

and convergence does not result. Delayed onset of phase IV, meanwhile, can lead to sustained greedy defection and convergence. If neither players' values ever drop below the threshold Q_{DD}, by the construction of Q-learning the players must be receiving higher average values than the Nash values. □

These experimental findings were initiated with different Q-values for the two players. If we were to initialize the values of the actions of one player to be equal to the other player, we would in fact see the convergence results shown in the theoretical setting explored in Section 3.6.2 above.

3.7 Empirical Comparison

The experiments show the result of running IGA and IQL-ϵ in a representative game from each class, using the payoffs in Table 3.1, for 100 simulated units of continuous time. We approximated the solutions numerically ($\alpha = 0.0001$) and used parameters of $\gamma = 0$ and $\epsilon = 0.1$. To allow the algorithms to demonstrate their full behavior, it is necessary to choose starting Q-values distinct from the Nash values. Figure 3.6 provides a time-series plot of the Q-values for representative games. Larger values of α show the same patterns, but with more

noise [Gomes and Kowalczyk, 2009].

In Matching Pennies, the two algorithms essentially behave the same way, ultimately converging to Nash. Deadlock (not shown) converges simply and similarly for both algorithms due to the cooperative dominant action equilibrium.

Both algorithms converge in B/S but identical starting points may lead IQL-ϵ and IGA to find different equilibria (coordinating on B vs. S). IGA converges to a pure Nash in Chicken (not shown), and IQL-ϵ sometimes converges. In the case of cyclic activity, it manages a minimum average reward of 5.7, higher than either the mixed Nash (3.3) or lower pure Nash (1).

IQL-ϵ never converges in Spoiled Child, but IGA will converge to the mixed Nash. Once again, we see IQL-ϵ attaining higher reward than IGA; around 2 and 1.2 for the two players instead of 1.5 and 1. These observations provide clues about the diverse and sometimes beneficial nature of non-convergence as well as important similarities within classes. In contrast to Prisoner's Dilemma, games in this subclass reach a periodic attractor.

Finally, the PD series (Figure 3.1) compares the policies of IQL-ϵ with IGA. IGA converges to the Nash (DD). While low initial values will lead IQL-ϵ to DD, here IQL-ϵ does not converge for the chosen starting values. When simulated with discrete, fixed-α updates, IQL-ϵ meets both conditions that describe a chaotic pattern. Specifically, it never returns to the same point, and small initial differences lead exponentially to arbitrarily large gaps later on. The average reward obtained by the IQL-ϵs is around 2.75 and 2.45, exceeding IGA's value of 1.0.

3.8 Conclusion

Motivated by the important and unique role of Q-learning in multiagent RL, it is a valuable exercise to catalog the dynamics of a continuous-time variant. This chapter documented a wide range of outcomes where two agents learn to play two-action games, varying from rapid convergence to Nash to unceasing oscillations above the Nash. Of particular interest is the complex behavior of Q-learning with ϵ-greedy exploration in Prisoner-Dilemma-like games, since the algorithm is able to achieve higher-than-Nash outcomes in this previously uncharted chaotic system. The increasing prevalence of mutually cooperative non-Nash strategies as exploration is decreased to zero is itself worthy of investigation. There is no reason to eliminate the possibility that this result would arise in games with more players or actions.

What is the purpose of this detailed description? A modeler should be prepared to execute its strategy with the aim of pursuing the highest possible reward. It is a non-trivial task to model and train these types of non-convergent learners. If a cooperative outcome is possible against a learning algorithm, then the goal of a modeler should be to induce the learner to behave this way. It may be possible to extract some extra reward while keeping the learner in a mostly cooperative state, but a modeler should be careful not to cause this frequency of cooperation to decrease in response, thereby negating the extra advantage sought after. The following chapter will describe how to build a model from the strategic components in Chapter 2, without explicitly reconstructing the dynamical system of updates.

Chapter 4
Modeling Learning Algorithms

In repeated games, adaptation is possible whenever prior experience can be applied to future play. As a result, the ability to model learners is an essential part of any comprehensive multiagent theory. The preceding chapter investigated in detail the dynamical behavior resulting from interacting Q-learners with ϵ-greedy exploration. This particular algorithm demonstrates unstable behavior as a result of the learning process. The discretized IQL-ϵ algorithm in self-play will not converge in certain types of games, as one might expect. Instead, it shifts its greedy actions repeatedly, going through a number of stages of varying length and never retracing the same values, the revealing signs of a chaotic system. This type of learner presents a unique challenge when trying to predict opponent behavior because of its natural instability.

This chapter proposes and evaluates a modeling algorithm that uses historical data to build and fit a predictive structure of learners or fixed history strategies. The modeler described below consists of the components from the meta-reasoning model, with an addition that incorporates learning from deviations of reward.

For the Prisoner's Dilemma game, the central message we can take from the previous chapter is that learners with certain properties can unexpectedly deviate from the single-round Nash equilibrium. The goal of a modeling agent is to identify opponents with this capacity, as well as the strategy that keeps

them from the Nash equilibrium while still taking maximum advantage. Later in the chapter, a special strategy will be introduced that is meant to exploit learning agents in this way, as long as the learning is constrained. The modeler will relax those constraints to compute an exploitative response against any evolutionary opponent, if it is possible.

The next section outlines the inextricable relationship between learning and teaching, and how teaching can be considered as a level of thinking beyond pure learning. Section 4.2 goes into more detail about how different learners need to be taught differently, centering on the issue of discounted future payoffs. In Section 4.3, I adapt the meta-reasoning framework for adaptive strategies by adding a separate model that changes the weights on each type according to a linear function. I present the results of experiments using the adapted meta-reasoning modeler in Section 4.4, and use a pre-existing model-free teacher to evaluate its performance.

4.1 Learning and Teaching

Since the earliest research into learning in games, the concepts of teaching and learning have been linked [Leyton-Brown and Shoham, 2009]. Indeed, in a multiagent context one can not easily separate *learning* from *teaching*, because just as an agent will necessarily learn from others' behavior, others are also influenced by the agent. Clearly, if an opponent has the ability to adapt its behavior over time, and some mutually beneficial arrangement exists, then it pays off to guide the learner to this arrangement. In many cases, it is necessary to communicate a credible threat of punishment to enforce good behavior. Likewise, if an opponent adopts a reasonable teaching posture, it is beneficial to attempt to learn the correct behavior that will lead to the highest reward.

Because the concept of teaching can mean several different things, let us define it for this context.

Definition 25. *A* teaching agent *or* teacher *in a game is an agent that adopts a strategy with the view that its opponent will behave with a (discounted) best response to that strategy. The teaching strategy is designed to select that learner's discounted best response with some goal in mind, such as whether it is optimal for the teacher.*

In cases of learning opponents, an optimal teaching agent will attempt to shape the environment the learner perceives in order to guide it into decisions the teacher prefers. Theoretically, a teacher is able to enforce any opponent strategy that results in a higher average payoff for the opponent than the minimax value for the game, due to the Folk Theorem, defined in Chapter 2.

In the discrete form of this problem, a teaching agent wishes to construct a deterministic state-machine strategy that aims to punish opponent behavior deviating from the enforceable strategy. It is possible for an algorithm, given a set of enforceable strategies, to compute the strategy machine that will enforce those behaviors, based on techniques dubbed the Computational Folk Theorem, or CFT [Littman and Stone, 2003]. The algorithm works by calculating the number of punishment rounds required to counteract the deviation from the desired opponent strategy, and therefore making it suboptimal to do so. An agent operating according to the CFT assumes that its opponent has the capacity to notice this pattern and adopt the desired strategy.

Another advance came in the form of *MetaStrategy* [Powers and Shoham, 2005], which combined a teaching stance to force acquiesence along with a background learning process to provide a best response if the teaching strategy does not work after a certain number of rounds. This approach can work well against a wide variety of opponents, although there is still no explicit formation

of an opponent model.

Researchers have investigated other combinations of learning and teaching, such as Q-learning with shaping [Babes et al., 2008], whereby a learner's policy is nudged towards some predefined policy by additional shaping rewards in the hopes of reaching an equilibrium. The logic behind this algorithm is similar to that of a correlated equilibrium, except that only one player is starting with the equilibrium behavior. The shaped strategy is not fixed because learning is still possible, so the learning agent may still learn a different strategy despite the shaping.

4.2 Teaching a Learner with Unknown Discount

The discounted sum of rewards for some discount rate γ is

$$U_i = \sum_{t=1}^{\infty} \gamma^t u_{i,t}.$$

The regret of player i with respect to the current strategy π_i is

$$\rho_i = \max_{\hat{\pi}_i} u(\hat{\pi}_i, \pi_{-i}) - -u(\pi_i, \pi_{-i}).$$

However, in a repeated game, the actions of one round may affect the rewards of future rounds. Therefore, let us define the future expected regret.

The *future expected regret* of a strategy is the difference of discounted future rewards between one course of action as opposed to another:

$$\varphi_i = \max_{\hat{\pi}_i} \sum_{t=1}^{\infty} \gamma^t (u_t(\hat{\pi}_i, \pi_{-i}) - -u_t(\pi_i, \pi_{-i})).$$

In a repeated game context with participants who look ahead some amount, it makes more sense for players to attempt to minimize future regret to the best

of their ability. This goal requires the player to be able to anticipate, to some degree, the effect of the current round's action.

Let us consider the Iterated Prisoner's Dilemma game, which, as we saw in the last chapter, leads certain types of learners to spontaneously arrive at mutually beneficial arrangements. Press and Dyson 2012 describe a method for computing teaching strategies that simultaneously keep a learning opponent j in a cooperative state and extract an arbitrarily large extra advantage for i when measured by $\frac{\bar{u}_i - P}{\bar{u}_j - P}$ (where P is the security value, or the Nash equilibrium value for a single round). This type of strategy will be discussed in more detail below, but we will hold onto this observation that evolutionary agents can be trained to produce good behavior for the teacher.

One-round memory strategies, in two-action, two-player games, consist of a four-valued vector $\pi = [\pi_1, \pi_2, \pi_3, \pi_4]$ corresponding to the probability of cooperation after each of the four joint action outcomes of Prisoner's Dilemma: both i and j cooperate, i cooperates/j defects, i defects/j cooperates, or both i and j defect. The strategy is selected to be the vector with the lowest amount of cooperation required to induce the opponent to cooperate at all times. That is, what is the minimum amount of required conditional cooperation?

For the all-cooperation enforcement to be successful, it is possible to use nonzero values for just π_1 and π_3, because the resulting transition efficiently communicates the future reward for cooperating and the strictest punishment for defecting. Furthermore, these values can be set equal to each other by finding the eigenvector of the transition matrix yielded by π. In this way, we can find a mixed Tit-for-Tat variant, expressed as single probability value π that fits into the vector where $\pi_1 = \pi$ and $\pi_3 = \pi$, that is designed to induce cooperation from a learner with almost arbitrarily long lookahead. Tit-for-Tat is just a

strategy where $\pi = 1$, but it is possible that a learner will still find cooperation to be the best strategy even when $0 < \pi < 1$.

This strategy simplification from four-dimensional vector to single dimensional value now yields a much more accessible decision process for the teaching agent, when faced with a learner of known γ. We can use the expected discounted regret to frame the problem. Over a two round window, the learner is faced with the choice of receiving $(1+\gamma)(\pi R + (1-\pi)S)$ from cooperating twice, or $\pi T + (1-\pi)P + \gamma S$ by defecting once and then cooperating to get back to the max reward state of mixed mutual cooperation. The following inequality represents the decision process where cooperating is beneficial, giving a break-even value for π:

$$\begin{aligned} (1+\gamma)(\pi R + (1-\pi)S) &> \pi T + (1-\pi)P + \gamma S \\ (1+\gamma)(\pi R - -\pi S) - \pi T + \pi P &> P + (\gamma - 1 - \gamma)S \\ \pi &> \frac{P+S}{(1+\gamma)(R-S) - -T}. \end{aligned}$$

The natural intuition behind this formula is that as γ increases, the lower bound for π decreases, allowing for more exploitation of the learner. The values for π are lower bounded by $\frac{P+S}{2(R-S)--T}$ for $\gamma = 1$ and $\frac{P+S}{R--S--T}$ when $\gamma = 0$. Using the payoffs of $[3, 0, 4, 1]$ gives a bound of $\pi > \frac{1}{3}$ for $\gamma = 1$ and $\pi > \infty$ for $\gamma = 0$, which means that when the learner does no lookahead, it is impossible to induce cooperation.

This formula also provides a way to distinguish between learners with different values of γ. If cooperation is observed for one value of π but not another, that information is evidence that the learner has a γ that would be consistent with the observation. To find this γ, observe that

$$\begin{aligned}(1+\gamma)(\pi R + (1-\pi)S) &> \pi T + (1-\pi)P + \gamma S \\ \gamma &> \frac{-\pi R + (\pi - 1)S + pT + (1-\pi)P}{\pi R + \pi S}.\end{aligned}$$

So, if γ is greater than the RHS of the inequality for mixed TFT ratio π, then the agent learns to cooperate, and otherwise the learner will defect. This information is useful to the model builder because it provides a way to distinguish between learners with different discount values. If a player cooperates with one value of π and not another, then γ is between the values output by the above formula by nature of the discounted rewards received by the learner.

4.3 Modeling a Learning Process with a Dynamic Meta-reasoning Model

If we wish to successfully build a model of a learning agent, we should first identify what we mean by learning and how we expect experience to alter behavior. In the field of reinforcement learning, agents use some method for identifying good behavior through reward feedback over many rounds of experience in various states of the world.

If the agent has access to the available state information, then a learning algorithm like Q-learning is guaranteed to eventually converge to the optimal policy for the given MDP, at least for the discount rate built into its update method. A learner with the four states mentioned in the previous section would update its values for each state it visits, and this separation creates room for the rewards to be different in each state. However, it is not obvious that any given learner *automatically* has the correct state representation to learn the optimal policy in response. For example, in the IQL-ϵ learner described in the last

chapter, there is only one state, and the action values will be updated for all states simultaneously. Nevertheless, this learner is still able to cooperate and so an opponent should attempt to achieve this outcome. Clearly, this learning structure creates an obstacle for a modeler that knows the best response for the state-based learner, and it cannot be ruled out that an agent would operate in this way. After all, simpler, more compact learning mechanisms might be preferred by agents, due to a large number of states or other domain-specific reason.

To make the setting more concrete, imagine a scenario where a single agent receives an income of 1 over an infinite number of time periods. It is repeatedly faced with a choice about whether to spend or invest its income. The agent derives utility of 1 from spending the income. The twist is that if the agent decides to invest, then it receives additional income equal to $I > 0$ of the invested amount X in the next T turns, at which point it has to spend and collect the higher utility. If I and T are large enough, it will be in the long-run interest of the agent to invest every time. For the moment, let us assume that the invested return is received in the very next round, and the amount is significantly larger than the spending lost by investing. However, in any given time period the reward is higher for spending. If investment was chosen last turn, the agent would receive that return plus the spending of the income for this turn.

A learning algorithm with just one round of memory, so that the state is the action from last turn, will quickly learn that the reward in the time period after investing is better than the one after spending. As a result it will gravitate towards a investing strategy, as long as the future rewards have a sufficient discount rate. The surprising thing is that the no-memory learner can learn to

invest also, as long as the rewards for doing so are high enough. That detail is the important one—the learned behavior can depend on the relative rate of reward between spending and investing, and not the single-turn absolute comparison (whether spending or investing generates a higher short-term reward).

Because the values for all states are aggregated in the no-memory scenario, the learner's policy will not converge to any particular solution. Instead, there will be instability, as investing becomes the greedy policy for a time until spending catches up as the short-term winner. At that point both values will suffer from the lower overall reward, followed by a switch back into investing as the value for the invest action is able to temporarily outpace the spending value once it invests several rounds in a row. (See previous chapter for more details in the context of IPD.) As a result, the learner's behavior appears as a mixture between several γ-optimal policies, and this ratio is by and large determined by the recent reward received. This reward is determined by the investment return. The mechanism by which this process works is somewhat complex, but it is related to how much of the lower spending reward (minus the investment income) is required to keep the value designated to spending in check. Lower investment rewards force the amount of spending to be relatively higher because the regret due to lost investment is also lower.

This discussion relates to repeated games because in games like IPD, a TFT-like strategy can emulate this exact same decision surface, albeit probabilistically. The defection payoff is analogous to the immediate higher payoff of spending, while cooperating takes the role of investing. In turn, an opponent facing a learner can select the return for cooperating by tuning the ratio π that determines how conditional the teacher is.

The link between average reward for the desired action and frequency that

a memoryless learner plays that action suggests a method for adapting the framework presented in Chapter 2. Along with fitting the default parameters of the model over all history, we can add more features to test the sensitivity of the model to rewards received by the modeled agent. If the agent appears to use a different strategic component more often as a result of higher (or lower) reward, we can incorporate this observation as additional features.

4.3.1 Applying the Meta-reasoning Framework to Model Learners

We want the model to capture how often the opponent's behavior is explained by each strategic component. That is, each time the opposing agent makes a choice, the model imagines that the decision is attributed to one of these factors, with an associated probability for each one. We would like the model to estimate these probabilities. If the behavior is in fact represented by a mixture of these factors, then the result should be a good predictive model. Formally, we can represent the default strategy vector as

$$\beta = [\epsilon, \phi, \mu, \omega_{\gamma_0}, \omega_{\gamma_1}, ..., \omega_{\gamma_n}]^T$$

to capture the base strategies (ϵ, ϕ, μ) representing the weights on randomness, repeating, and imitation respectively, and ω parameters for the optimal response against the teaching distribution for various values for γ. In many interesting games, all of these components are necessary for an accurate model.

The features corresponding to each of these weights are defined by the the

input matrix

$$X_i = \begin{bmatrix} \frac{1}{|A|} & x_{\phi,1} & x_{\mu,1} & x_{\omega_{\gamma_0},1} & \dots & x_{\omega_{\gamma_n},1} \\ \frac{1}{|A|} & x_{\phi,2} & x_{\mu,2} & x_{\omega_{\gamma_0},2} & \dots & x_{\omega_{\gamma_n},2} \\ \dots & & & & & \\ \frac{1}{|A|} & x_{\phi,T} & x_{\mu,T} & x_{\omega_{\gamma_0},T} & \dots & x_{\omega_{\gamma_n},T} \end{bmatrix}.$$

The values at row t corresponds to the feature values at time t. The values of each column are encoded as bits that are positive when the action at time t is equivalent to the action predicted by the relevant feature, and zero otherwise. These binary values in fact represent the probability that each feature is the same as the relevant action. For the actions that are completely determined, the values are always either 100% or 0%. An exception is the random action which is set $x_{\epsilon,t} = \frac{1}{|A|}$ to best fit a random variable with probabilities $0 < x_{\epsilon,t} < 1$. Intuitively, this feature value is meant to capture the probabilistic nature of the predicted action, because a random choice has $\frac{1}{|A|}$ chance of choosing any action, whereas the other features are determined to be 0 or 1. This component will gain weight in cases where the probability of the others are less than this uniform value.

The target output b is always set to one. Once the modeler has gathered sufficient historical data, the features for each time step become columns for fitting the weights by regression so that:

$$X_i\beta = b.$$

This setup works to discover the probabilities of each strategic component because the regression will attempt to make the weights add to one and the idea is that these strategies are sufficient to cover the space. Of course, in this case, the weights need to be bounded between zero and one, so the problem is one

of constrained optimization minimizing the squared error. Practically speaking, if the learner uses a discount rate that is not included in the set of policies, we would expect a mixture of two adjacent strategies to be part of the output model.

While there are features that can respond directly to actions (imitation) or even the long-run best response (the γ-optimal policies), the astute reader will notice that the model fit by this procedure is not designed to change with experience. To allow for adaptation of the model itself we are forced to make the alteration described earlier in this section, namely to come up with additional weights that depend on rewards. To achieve this outcome, we define the delta features as a scaled version of the previous matrix:

$$X_i^\delta = \begin{bmatrix} \frac{1}{|A|} & \bar{u}_{i,1}^\delta & 0 & 0 & ... & 0 \\ \frac{1}{|A|} & 0 & \bar{u}_{i,2}^\delta & 0 & ... & 0 \\ ... & & & & & \\ \frac{1}{|A|} & 0 & 0 & 0 & ... & \bar{u}_{i,T}^\delta \end{bmatrix} \begin{bmatrix} \frac{1}{|A|} & x_{\phi,1} & x_{\mu,1} & x_{\omega_{\gamma_0},1} & ... & x_{\omega_{\gamma_n},1} \\ \frac{1}{|A|} & x_{\phi,2} & x_{\mu,2} & x_{\omega_{\gamma_0},2} & ... & x_{\omega_{\gamma_n},2} \\ ... & & & & & \\ \frac{1}{|A|} & x_{\phi,T} & x_{\mu,T} & x_{\omega_{\gamma_0},T} & ... & x_{\omega_{\gamma_n},T} \end{bmatrix}.$$

Here, we have the original features weighted by the deviation $\bar{u}_{i,t}^\delta$, which is the difference between the average reward up to time t, $\bar{u_{i,t}}$, and some anchor reward $\hat{u}$ taken to be some midpoint reward that separates winning from losing. This anchor has a similar function to the threshold of WoLF-IGA [Bowling and Veloso, 2001] and can be chosen by one of several methods, such as the average available payoff, average between best and worst payoffs, or half the best payoff. In the normalized games we will consider, all of these values are set to 0.5. The original feature matrix X_i is scaled by diagonal matrix $\bar{U}_i^\delta$ to form delta features X_i^δ. These delta features, combined with the default features, form a dynamic model that adapts with changes in the average reward received by the agent. To come up with the likely model given a teaching strategy, all that

is required is to determine the reward and deviation, scale the delta model by this amount, and form a new regression:

$$[X_i \quad \bar{U}_i^{\delta} X_i] \begin{bmatrix} \beta \\ \beta_i^{\delta} \end{bmatrix} = \begin{bmatrix} b \\ b \end{bmatrix}.$$

This operation may change the reward received by the resulting model, so it would then be necessary to iterate the process until convergence is achieved.

Let us return to the simple investment example to examine how the delta model captures a learning agent's behavior. We know that a stateless learner is inherently unstable in these cases. It will not converge to any particular pure strategy, and so its strategy will be mixed from the different strategic components. According to the analysis in Section 4.2, the probabilistic strategy will vary in proportion to the teaching strategy employed. Another way of looking at this phenomenon is to say that the model of the learner changes along with the reward it receives. For instance, imagine that when the next-round return is 1.9 for every 1 invested in the current round, the learner invests with probability 0.9 and spends the rest of the time. If the probability drops to 0.5 when the return is 1.5, and to 0.1 when the return is 1.1, there is evidence of a linear relationship between reward and strategy. Take the default model to be $\omega_0 = 0.5, \omega_{0.9} = 0.5$. One possible model would give a delta weight $\omega_{0,\Delta} = -0.05$ for the no-lookahead policy. Conversely, a long-term lookahead policy, like when $\gamma = 0.9$, would have a positive delta weight $\omega_{0.9,\Delta} = 0.05$. If the midpoint reward is taken to be 1.5 for this learner, then the model would give the correct strategy for the three return payoffs. To get accurate estimates of these parameters, the modeler needs to test the response of the learner against a variety of possible payoffs.

It is valid to ask what would occur if this part of the learning model is

missing. Because the strategies vary dramatically as a function of the payoffs, a standalone default model would not be able to capture this variation. It is likely that the model would be very different given a different set of testing transitions, which is not an ideal property of a modeling process. Potentially, each strategic component would provide less predictive power than the random action, and so most or all of the weight could end up on ϵ, the probability of the randomness feature. Figure 4.1 demonstrates what happens to the default model as more experience is added. We observe rapid shifting among the factors, where some coefficients have all the weight and others have none, and this allocation is unstable as trials are added. This outcome contrasts with Figure 4.2, which employs the delta features. The resulting delta model is stable. Only the highest weighted features are shown.

Once the model is constructed, all that remains is to find the best strategy against it, whether it is designed to teach a learner, or the optimal policy learned from a fixed player. For fixed distributions that do not attempt to maximize reward against their opponent (level 0), the first four base strategies are sufficient to represent nearly any appropriate distribution, at least for 2-action games. These strategies are the base components of playing random, repeat, imitate, or one-round best response against a random opponent and the weights are contained in $[\epsilon, \phi, \mu, \omega_{\gamma_0}]$. To be precise, since the modeler only really cares about the ways it can influence the modeled agent, the important weights are μ (which may be positive for a fixed player) and the optimized policy weights (which will be zero for a non-learner). Furthermore, these fixed strategies will not adapt to the reward they receive, so the delta model should be expected to have only zeroes or have no effect on the distribution. Against these strategies, there is only a single MDP that will be faced, and the modeler

can easily find the best policy against it.

For the learners where the delta-model is relevant, the modeler needs to perform an extra step to determine the how the model reacts to the rewards resulting from the strategies it is considering, and find the best reward coming from them. That is, the modeler recognizes that the opponent strategy may change as a result of the strategy applied to the opponent. Therefore, the modeler needs to iteratively decide how the modeler strategy affects the opponent payoff, which in turn determines the reactive strategy. In the IPD scenario, this step amounts to using different TFT ratios of π and discovering where cooperation drops off. The end result of this model will be to find a best policy that reacts to a fixed opponent, and generates a payoff structure that leads a learning opponent to the desired behavior. See Algorithm 2 for more details.

$\bar{u_i} = 0$: average reward initialized to 0
$\hat{u_i}$: Win/loss threshold initialized to some midpoint payoff value
ω_i: Delta-model for agent i fit from data
for $\bar{u_i}$ not converged **do**
  $\pi_i = \omega_i + \omega_{i,\delta}(\bar{u_i} - \hat{u_i})$
  $\pi_M = \arg\max_\pi \sum_t U_t(\pi, \pi_i)$
  $\bar{u_i} = \frac{1}{T} \sum_t U_t(\pi_i, \pi_M)$
end for

Algorithm 2: Iterative Delta-model response

4.4 Experiments

This section will show how the above model-building algorithm performs against a selection of simple learning algorithms along with assorted teaching strategies. The evaluated setting will be Iterated Prisoner's Dilemma, a well-studied game that continues to fascinate theorists in this space.

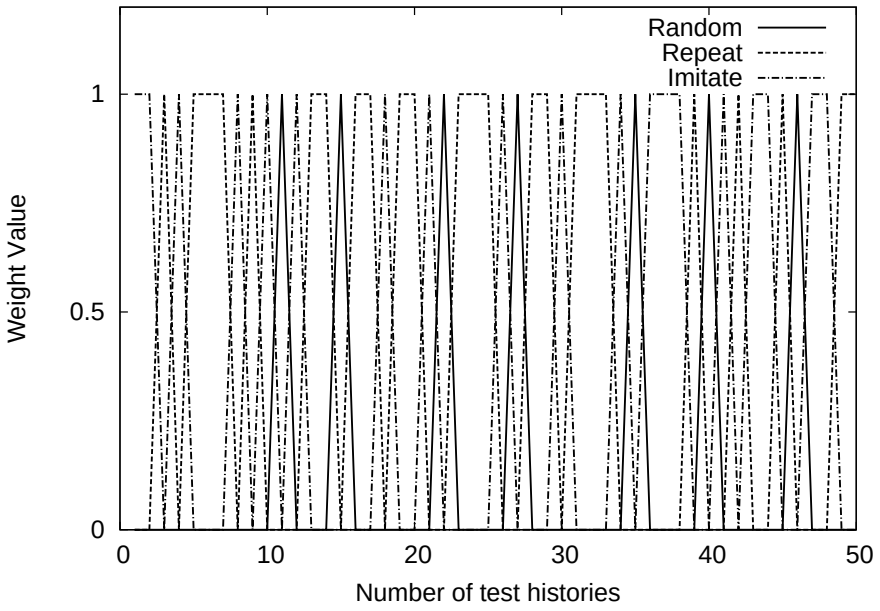


Figure 4.1: Effect of additional experience on the default model, without the support of the delta features. Model does not converge.

4.4.1 Hypothesis

Recent work has taken the teaching question further by taking into account the observation that a probabilistic strategy can exploit a learner through ensuring that the optimal response is still the one desired by the teacher. To be specific, Press and Dyson 2012 have identified a class of strategies called *zero-determinant* (ZD) that have the property that the learning opponent's score is set at the discretion of the agent executing a ZD strategy. The central idea behind this type of strategy is that the stationary state of the ZD strategy's transition matrix is such that the payoffs take on a linear relationship, becoming in essence a constant-sum game. The opponent has the option of accepting some portion of the available reward, or rejecting it and taking the minimax

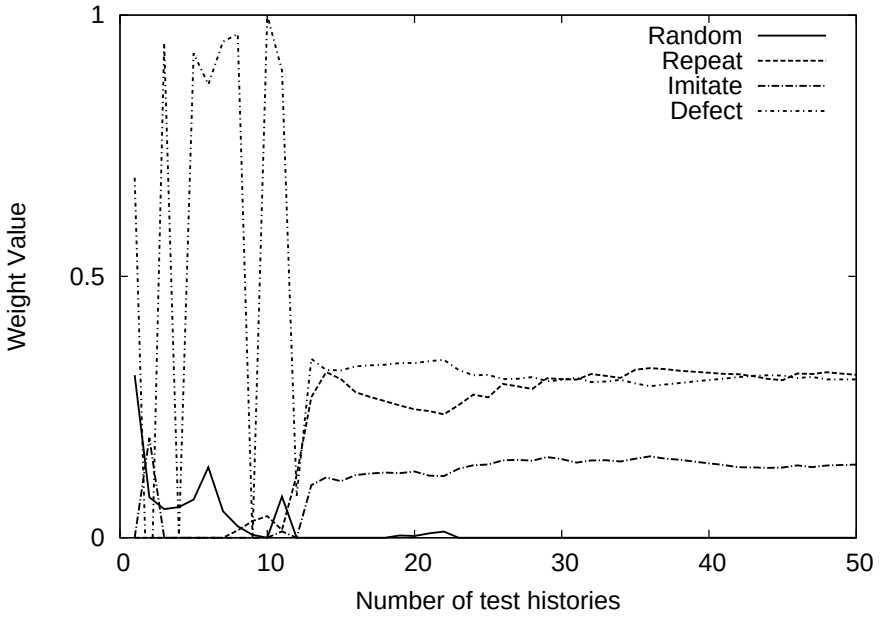


Figure 4.2: Effect of additional experience on the basic model, with the support of the delta features. Note the convergence over time.

reward. In effect, a ZD strategy allows a player to exploit an adaptive opponent in games like Iterated Prisoner's Dilemma, creating an asymmetric payoff structure from a symmetric game with a dominant action.

According the ZD formula, the result is four different values that, when combined, are guaranteed to lead an opponent who maximizes average reward to a policy that benefits the ZD-teacher. Strangely, while π_4 (mutual defection) is zero, the formula is usually a positive number for π_2, which would not appear to be useful given that the learner will always be cooperating anyway, as long as it is choosing the best policy.

This finding is valuable because it presents a way for a teaching agent to select an optimal teaching strategy. However, there are several weaknesses

in the analysis. First, as the authors themselves mention, the teacher, agent i, can be thwarted by a player j who decides to take the same teaching stance, or alternately who has a "theory of mind." In other words, if j reasons that i will eventually adapt to j, then it is in j's interest not to conform to the exploitation proposed by i. A second, more subtle issue is that the evolutionary player is assumed to want to maximize the average score over the infinite horizon. Therefore, if the player is acting with respect to future discounted rewards, then rejecting the extortionate proposal could actually appear to be the more rational response. A related problem is that the learner may not have the proper state representation that it would need to learn the optimal response.

The simple meta-reasoning model introduced above is meant to address this second issue, which centers on recognizing that the opponent operates under some discount rate and thus cannot be fully exploited in the way that Press and Dyson assume. Therefore, the hypothesis for these experiments is that a modeling agent that can correctly predict the amount of lookahead in these learners will outperform a teaching strategy that assumes a learner that achieves an optimal average reward, as opposed to future discounted reward. The model proposed here is a direct instantiation of the claims in the thesis in terms of the features it extracts and the behaviors it captures.

4.4.2 Evaluation

An important characteristic of multiagent environments is that the population selected as the opponent pool has a major effect on the performance of the evaluated algorithms. In recognition of the teaching/learning dynamic, half of the experimental set of agents will be learning players, and the other half will consist of fixed strategies. For the pool to make distinctions between stronger

and weaker strategies, the agents in these experiments should be diverse and cover the space of behaviors.

Some basic opponents can be used to identify whether players choose the best one-round action if there are no bad consequences for doing so. Two obvious contenders are players who play just one of the actions in a repeated sequence. Another is a random player, exhibiting the type of behavior typically assumed as base play in the cognitive hierarchy literature.

Conditional strategies add a new layer of sophistication, because they attempt to modulate short-term and long-term action choices. The simplest of these in IPD is the Tit-for-Tat strategy which achieved surprising success in early tournaments [Axelrod, 1984]. In addition, we can add some variety to this basic imitation strategy by allowing for some probability of dominant action play, to test whether players still learn a policy of cooperating despite the noise.

Finally, we have a Zero-Determinant (ZD) strategy to use as a baseline. This strategy has been shown to extract a great deal of extra value from evolutionary opponents, and so should be included in the list of contenders. Because a ZD strategy is constructed with this goal of teaching in mind, we should expect it to have a big advantage when learners are present. While these fixed strategies are by no means exhaustive, they cover a wide range of behaviors from pure actions to random actions to conditional reactions.

The learning population contains a diversity of methods as well as parameter values. Some learners (specifically IQL-ϵ) use a strict maximum threshold to choose the action, and add some extra exploration to make sure that the best action remains so. In contrast, the Boltzmann learning agents take a proportional approach to exploration, by exponentially weighting the estimated

values. Because a Boltzmann learner shares properties with gradient-based algorithms like gradually tuning its strategy, it will be considered a representative for the class of gradient-ascent learners. Within these algorithms, the learning can be subdivided into agents with zero or one round of memory. For the agents with state representation, the discounting factor γ can be varied. These experiments will use two values of $\gamma, 0.75$ and 0.9.

The meta-reasoning modeler is the only entrant that has the capacity to gain knowledge from putting other agents through a test suite of strategies and then using the model to customize a response. Therefore, the experiment will allow the modeler this opportunity to extract the relevant data by playing a number of games against each player before the actual tournament begins. For Prisoner's Dilemma, the testing takes the form of a number of stochastic zero- and one-round transitions. The zero-round transitions are a simple probability for playing one action or the other, $p_C = [0.0, 0.1, 0.2, ..., 1.0]$. The one-round transitions hold a spectrum of values of $\pi = [0.0, 0.1, 0.2, ..., 1.0]$, which means that there is some probability of the modeler imitates the opponent when the opponent cooperates, and the rest of the time it defects. This preliminary play will therefore consist of observing opponents against 22 strategies, which will be played over a period of time sufficient to see convergence to a particular policy.

4.4.3 Results

First, it is instructive to demonstrate how the modeler is able to represent each of the opponent strategies. Tables 4.1 and 4.2 introduce the opponent pool

and the derived model for each agent. These values represent probabilities responding to each of the strategic components that were output from the regression described above. That is, if a history can be best described by imitation, then it should have a high value for μ. If it always defects, then the $\omega_{\gamma 0}$ should be high.

The fixed strategies are fairly straightforward to interpret. All of the probability mass is concentrated in the first four columns, corresponding to the three base strategies and the single-round best response (Defection). The agent who only cooperates is classified as a pure repeating strategy, which makes sense given that the classification strategy is initialized for both players to cooperate. An all-defecting agent is correctly identified as such, with all the probability in column four. The random strategy is found to have $\epsilon = 1.0$, and the pure Tit-for-Tat is seen as 100% imitating. The mixed TFT agents also have nearly exactly correct ratios.

The learning agent models tell a richer story. These agents demonstrate a broader range of behavior, with large weights in the optimal discounted policies. Here, the delta-model component, shown in the second row for each agent in Table 4.2, is key. The stateless IQL-ϵ agent, for example, has positive delta-values for $\omega_{\gamma=0.9}$ and $\omega_{\gamma=0.75}$, and negative delta-values for $\omega_{\gamma=0}$ and $\omega_{\gamma=0.5}$. What that means is that this type of learner acts like it has a far lookahead when it is receiving high reward, and does more defection as the reward is decreased. Concretely, if it were receiving 0.5 average reward, it would defect 0.22 of the time and cooperate 0.43 of the time, with the rest split between repeating and imitation. If it receives 0.4, its best response vector would become $[\omega_{\gamma=0}, \omega_{\gamma=0.5}, \omega_{\gamma=0.75}, \omega_{\gamma=0.9}] = [0.22, 0, 0.16, 0.27] + (0.4 - 0.5) * [-1.00, -0.35, 0.78, 1.00] = [0.32, 0.03, 0.08, 0.17]$ which demonstrates a

Table 4.1: Output default models for each of the fixed strategies in the opponent pool. A value of 0.00 is denoted by $-$.

Strategy	Random ϵ	Repeat ϕ	Imitate μ	$\omega_{\gamma=0.0}$	$\omega_{\gamma=0.5}$	$\omega_{\gamma=0.75}$	$\omega_{\gamma=0.9}$
All-cooperate	–	1.00	–	–	–	–	–
All-defect	–	–	–	1.00	–	–	–
Random	1.00	–	–	–	–	–	–
Tit-for-Tat (100%)	–	–	1.00	–	–	–	–
Tit-for-Tat (70%)	–	–	0.70	0.29	0.01	–	–
Tit-for-Tat (50%)	–	–	0.50	0.50	–	–	–
ZD exploiter $[0.8, 0.5, 0.3, 0.0]$	–	0.49	0.30	0.21	–	–	–

large shift towards defection (the first and second entries represent defect in this case and the third and fourth are cooperation). On the other hand, if it receives 0.6 instead, then this vector becomes $[0.22, 0, 0.16, 0.27] + (0.6 - 0.5) * [-1.00, -0.35, 0.78, 1.00] = [0.12, 0.0, 0.24, 0.37]$, so that it cooperates relatively more than the baseline.

The other learners with at least one round of memory demonstrate similar patterns, although a somewhat less pronounced relationship between model and reward. To emphasize the importance of the delta weights, consider that without them the zero-memory learner will be classified as a random agent. With another set of testing strategies, the default model alone will arrive at other settings but with the delta-model the same estimates are found.

The other learners are fairly consistent with the expected behavior. If the rewards are such that cooperation is optimal given the agent's γ, then it cooperates so that the high-discount features are more predictive. The obvious exception is the memoryless Boltzmann learner, which converges to the defection action regardless of what the opponent does, and while it converges there is a lot of randomness in its action choice.

The overall results of the pairwise tournament between the agents are

Table 4.2: Output default (first row, sum to 1) and the associated delta models (second row, bounded by -1 and 1) for each of the adaptive learning strategies in the opponent pool. A value of 0.00 is denoted by $-$.

Strategy	Random ϵ	Repeat ϕ	Imitate μ	$\omega_{\gamma=0.0}$	$\omega_{\gamma=0.5}$	$\omega_{\gamma=0.75}$	$\omega_{\gamma=0.9}$
IQL-ϵ	0.02	0.23	0.10	0.22	–	0.16	0.27
δ-model	–	-0.24	-0.13	-1.00	-0.35	0.78	1.00
No memory							
Boltzmann	0.38	0.27	0.13	–	–	0.22	–
δ-model	–	0.44	0.81	-1.00	0.64	-1.00	-0.32
No memory							
IQL-ϵ	0.20	0.03	–	0.46	–	0.06	0.24
δ-model	-1.00	-0.04	-0.15	-1.00	0.26	0.34	0.93
$\gamma = 0.75$							
Boltzmann	0.22	0.46	–	–	–	–	0.32
δ-model	-1.00	1.00	-0.39	-1.00	0.23	-0.03	0.33
$\gamma = 0.75$							
IQL-ϵ	0.33	0.17	0.05	–	–	0.01	0.43
δ-model	–	0.50	0.02	-0.72	1.00	-1.00	0.11
$\gamma = 0.9$							
Boltzmann	0.06	0.07	0.02	–	–	0.29	0.55
δ-model	-1.00	-0.63	0.39	-0.14	-0.81	1.00	0.77
$\gamma = 0.9$							
Meta	–	0.26	0.27	–	–	0.15	0.33
δ-model	–	-0.60	0.13	-0.97	0.32	0.21	1.00

shown in Figure 4.3 and Table 4.3. Average payoffs are shown for the normalized Prisoner's Dilemma where $R = 0.75, T = 1.0, P = 0.25, S = 0.0$. The first column shows the score of the agent we are evaluating against all the rest, listed by name. The second column of the table gives an idea of how well the others do against each opponent. There is a fairly tight distribution, but the *Modeler* agent in the final row is an obvious outlier to the upside.

Pairwise results provide a more detailed look at the dynamics between the different types of teaching and learning agents. First, the teacher versus teacher games (Table 4.4) demonstrate that two fixed strategies that aim to exploit each

other will often end up punishing each other instead. Only the 100% Tit-for-Tat agent stays in a state of cooperation with itself. As one would expect, the strategies that teach conditional cooperation are more successful against the learning agents, shown in Table 4.5. The learning agents in these match-ups show some mixed success (against teachers, see Table 4.6). In general, they do better against teachers than other learners (Table 4.7), which are not necessarily programmed to enforce cooperation. Furthermore, the learners with higher discount rates appear to perform more strongly than those with lower γ. The most likely explanation for this phenomenon is that the learners with longer lookahead get the slightly higher reward from the exploiting fixed players, instead of merely reverting to mutual defection. That is, they are learning what the exploiters are trying to teach them: that it pays off to cooperate, albeit in the exploiting agent's favor.

4.4.4 Transferability to Other Games

One goal of building meta-reasoning models is that they can be used to predict behaviors in alternate situations where payoffs differ from the original training data. This transferability task is meant to provide a check that the agent models are somewhat generalizable, and not specially built for a given setting.

As a proof of concept, let us begin by using the models trained on the Prisoner's Dilemma with payoffs $[3, 4, 0, 1]$ in a game with payoffs $[2, 3, 0, 0.5]$. This test represents a game with the same payoff structure but different payoffs. In this instance, the models are completely transferable and the performance of the transferred model and trained model in this game are statistically equivalent.

A final set of experiments was conducted to explore how the models trained

Table 4.3: Mean performance of agents against the population in normalized IPD. Note that the modeler scores the best compared to other agents in the population.

Name of Agent X	Agent X's mean scores vs. others	Others' mean scores vs. Agent X
Teachers (fixed)		
Always defect	0.36	0.21
Always cooperate	0.35	**0.88**
Random	0.34	0.55
Tit-for-Tat, 100%	**0.53**	0.53
Tit-for-Tat, 70%	0.50	0.41
Tit-for-Tat, 50%	0.48	0.30
ZD teacher	0.49	0.33
Learners		
IQL-ϵ 0-round memory	0.49	0.40
IQL-ϵ 1-round, $\gamma = 0.75$	0.37	0.29
IQL-ϵ 1-round, $\gamma = 0.90$	**0.52**	**0.64**
Boltzmann 0-round memory	0.48	0.49
Boltzmann 1-round, $\gamma = 0.75$	0.45	0.43
Boltzmann 1-round, $\gamma = 0.90$	0.49	0.47
MetaStrategy 1-round, $\gamma = 0.75$	0.46	0.45
Modeler	**0.62**	0.54

Table 4.4: Pairwise performance of fixed teaching agents against other teachers in IPD. The agents are the same but the column headings are abbreviations of the row headers.

Name	All D	All C	Rand.	TFT-100	TFT-70	TFT-50	ZD	Mean
Always defect	0.25	1.00	0.63	0.25	0.25	0.25	0.25	0.41
Always cooperate	0.00	0.75	0.38	0.75	0.52	0.38	0.45	0.46
Random	0.12	0.87	0.50	0.50	0.38	0.31	0.35	0.43
Tit-for-Tat, 100%	0.25	0.75	0.50	0.75	0.25	0.25	0.25	0.43
Tit-for-Tat, 70%	0.25	0.83	0.54	0.25	0.34	0.25	0.32	0.40
Tit-for-Tat, 50%	0.25	0.87	0.56	0.25	0.25	0.25	0.25	0.38
ZD Teacher	0.25	0.85	0.55	0.25	0.47	0.25	0.29	0.42

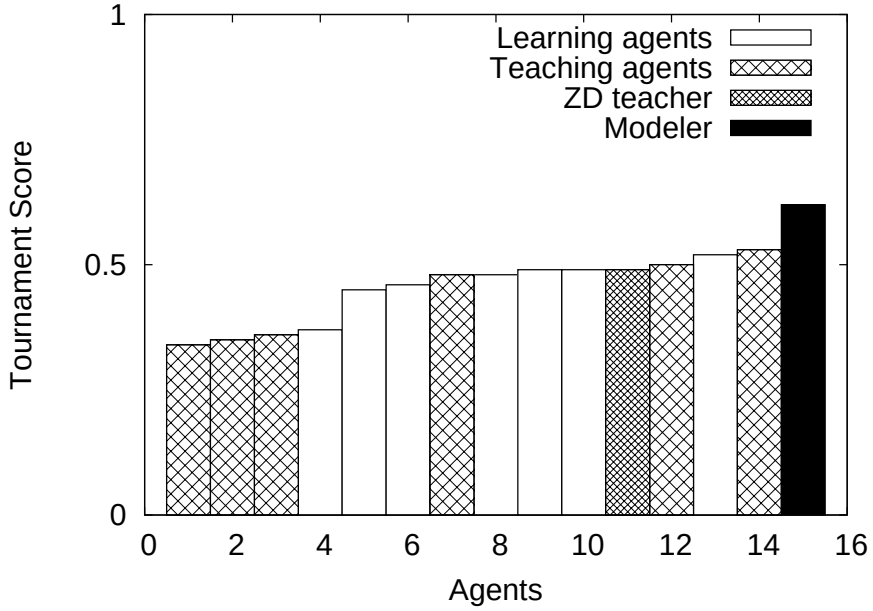


Figure 4.3: Mean performance of agents against the population in normalized IPD. The modeler (black bar) outperforms the nearest competitor by over 10%, a significant margin in this type of tournament.

Table 4.5: Pairwise performance of fixed teaching agents against learners in IPD.

Name	IQL-ϵ	IQL-ϵ	Boltz.	Boltz.	Meta	IQL-ϵ	Boltz.	Mean
Rounds of memory	0	1	0	1	1	1	1	
γ	0	0.75	0	0.75	0.75	0.90	0.90	
Always defect	0.30	0.31	0.34	0.36	0.28	0.33	0.40	0.33
Always cooperate	0.10	0.35	0.06	0.07	0.74	0.73	0.01	0.29
Random	0.16	0.51	0.18	0.14	0.24	0.62	0.14	0.28
Tit-for-Tat, 100%	0.63	0.69	0.32	0.64	0.54	0.74	0.58	0.59
Tit-for-Tat, 70%	0.29	0.62	0.32	0.60	0.48	0.76	0.76	0.55
Tit-for-Tat, 50%	0.27	0.55	0.31	0.59	0.52	0.77	0.63	0.52
ZD Teacher	0.34	0.57	0.33	0.56	0.46	0.79	0.55	0.51

Table 4.6: Pairwise performance of learners against fixed teaching agents in IPD.

Name	γ	All D	All C	Rand.	TFT-100	TFT-70	TFT-50	ZD	Mean
IQL-ϵ	N/A	0.23	0.97	0.61	0.63	0.27	0.26	0.28	0.46
IQL-ϵ	0.75	0.22	0.98	0.60	0.32	0.29	0.26	0.27	0.42
IQL-ϵ	0.90	0.22	0.76	0.46	0.74	0.66	0.36	0.42	0.52
Boltz.	N/A	0.23	0.88	0.50	0.69	0.42	0.31	0.35	0.48
Boltz.	0.75	0.21	0.98	0.62	0.64	0.42	0.32	0.35	0.51
Boltz.	0.90	0.20	1.00	0.61	0.58	0.67	0.33	0.35	0.53
Meta	0.75	0.24	0.75	0.59	0.54	0.36	0.31	0.32	0.44
Modeler	N/A	0.25	1.00	0.62	0.75	0.53	0.38	0.46	0.57

Table 4.7: Pairwise performance of learners against learners in IPD.

Name	γ	IQL-ϵ	IQL-ϵ	IQL-ϵ	Boltz.	Boltz.	Boltz.	Meta	Modeler	Mean
Rounds		0	1	1	0	1	1	1		
γ		0	0.75	0.90	0	0.75	0.90	0.75		
IQL-ϵ	N/A	0.65	0.37	0.68	0.33	0.36	0.72	0.26	0.68	0.51
IQL-ϵ	0.75	0.36	0.67	0.46	0.32	0.46	0.36	0.52	0.60	0.47
IQL-ϵ	0.90	0.67	0.37	0.74	0.30	0.51	0.75	0.54	0.36	0.53
Boltz.	N/A	0.27	0.42	0.30	0.31	0.36	0.36	0.25	0.31	0.32
Boltz.	0.75	0.26	0.36	0.45	0.31	0.41	0.34	0.42	0.64	0.40
Boltz.	0.90	0.70	0.31	0.73	0.28	0.34	0.29	0.39	0.62	0.46
Meta	N/A	0.29	0.58	0.74	0.32	0.34	0.46	0.53	0.54	0.48
Modeler	N/A	0.68	0.70	0.80	0.31	0.72	0.77	0.58	0.75	0.66

in Prisoner's Dilemma perform in the Spoiled Child game introduced in Chapter 3. Briefly, Spoiled Child is an asymmetric game where one player is the child and the other is the parent. The child can choose to behave or misbehave, while the parent decides to spoil the child or punish him. The parent is happy to spoil the child when he is behaving, but prefers to punish when he is misbehaving. The child's payoffs are similar to Chicken, where he is better off getting spoiled, and prefers to misbehave most of all. However, unlike with the symmetric payoff in Chicken, if the parent is punishing, then the child would rather behave. This game has a single mixed Nash equilibrium, but the unstable Spoil/Behave payoff is higher than the mixed Nash payoffs. As a result, learners can cycle in and out of this outcome over long periods.

Because these dynamics are similar to Prisoner's Dilemma while also diverging somewhat, it is a valuable test case for the transferability of models learned in IPD. Indeed, with models trained on agents in IPD, the modeler performs nearly identically (within 0.01) of the models trained on the Spoiled Child.

A note of caution is still in order, however, because changing the complexity or information content of a game can have an effect on learning and reasoning algorithms as the internal computational and cognitive costs of making decisions varies. For example, an agent might be observed playing the best response in a simple game but resort to a random strategy in a harder one.

4.5 Conclusion

The takeaway message of this chapter is that the proposed method for modeling learning agents achieves the goal of discovering the optimal policy against both learners and fixed players. Other algorithms have not been optimized for this purpose, and instead do well against either fixed or adaptive players under certain assumptions.

Chapter 5

Reasoning Models in the Lemonade Stand Game Tournaments

Imagine a setting where a population of agents play a multi-round game in which the payoffs are different every time a new match begins. Many real world situations are likely to have much in common with these *non-experienced* domains, as the players respond to a changing world. This problem has been thoroughly studied in classification/regression [Pan and Yang, 2010] and reinforcement-learning problems [Taylor and Stone, 2007] under the guise of transfer learning, but it is rarely mentioned in either traditional game theory or the field of multiagent systems. Many questions that might have straightforward answers in static multiagent environments, such as what constitutes a best response to a given opponent strategy, become much more challenging in previously unseen situations. In single-agent learning problems, transfer learning involves mapping a feature representation or a decision rule between tasks or domains, while in the multiagent case we must carry across how a strategic thinker solves slightly different problems. The underlying question is the same, however: what can be gained from experience when the training data differs from the test set in some limited way? The meta-reasoning framework provides a method to address these issues.

5.1 Introduction

This chapter applies the previously introduced meta-reasoning algorithm to a setting known as the Lemonade Stand Game (LSG) [Zinkevich, 2009]. The LSG is a simple, constant-sum location game with three players and many possible equilibria. Traditional, unstructured learning algorithms fail to perform adequately because the act of intra-match learning puts an agent at a disadvantage. To test the overall thesis, for learning data we will analyze a series of actual competitions to figure out which computerized strategies arise within a community of researchers and project them into the iterated best response (IBR) level space. The LSG tournament follows in the spirit of previous agent competitions from Rock-Paper-Scissors [Egnor, 2000], a simple two-player three-action zero-sum game, to the Trading Agent Competition [Stone and Greenwald, 2005], where multiple actors interact using complex optimizations. On the complexity spectrum of games, LSG lies between these two competitions, with three players, constant-sum payoffs, and a total of 12 actions. However, scores in this game can vary widely depending on the particular agents present, giving an edge to meta-reasoners with an effective model of the game and their opponents.

It is worth noting here that the concept of strategic levels of reasoning has not, for the most part, been applied to repeated games. An exception is the Rock-Paper-Scissors tournament, where meta-reasoning strategies were used to great effect [Egnor, 2000]. In human behavioral experiments, it is difficult to control for learning effects shown by participants because people learn in different ways in different contexts. There are also certain theoretical problems that arise when deciding upon what is meant by a step of reasoning in a repeated game. Some of these issues result from the conflict of whether to

consider reasoning as the computation of a single action in time, or a longer sequence of actions. In situations where this scope is in doubt, we will prefer the latter definition but reduced to a repeated strategy, due to ease of representation. In addition, the issue of how to anchor the reasoning by base strategy is much less obvious when the game and population have a state. Games where actions imply a state, such as location games, are convenient domains because the impression of being in a location immediately suggests constancy as a basic action choice from which reasoning can flow. Therefore, even if there is no environmental state, we can speak of the population as occupying a state in location space.

This chapter is organized as follows. First, in Section 5.2, we introduce the concept of a location game as a special case of normal form games and address some issues that arise if these games are repeated.The rest of the chapter is an original contribution because the Lemonade Stand Game is such a new and unexplored domain. Following, in Section 5.3, is the Lemonade Stand Game setup as played by the agents in several of its variations. In Section 5.4, we look at how depth of reasoning and time horizon capture strategies in the basic symmetric game. Next, we do the same for the asymmetric version of the game, in which game payoffs are altered between separate matches so that experience from one game must be transferred to the future ones. Prediction has a critical role in this type of game, as utilities are strongly determined by the actions and reactions of opponents. Finally, we analyze the full group of submitted agents from each of the four annual competitions to provide an empirical validation for the utilized methods for the purpose of prediction and accurate modeling. A strategy informed by the population meta-reasoning model is compared against both submitted agents and alternative learners.

5.2 Related Work: Location Games

Hotelling games, also known as location games, were devised as a way to represent the actions of a small number of firms in a market [Hotelling, 1929]. Typically, the players of these games choose a location in some space that represents the demand of a customer base, and the firms "sell" their products to the closest customers. (In more advanced versions, economists study games where a pricing mechanism also exists, but we will focus purely on the location aspect here.) In the simple two-player case played on a uniform-density line, it is fairly straightforward to show that a pure equilibrium exists with both players at the halfway point.

These types of games have been well studied in the economics literature [Gabszewicz and Thisse, 1992; Kilkenny and Thisse, 1999]. From a practical standpoint, location games like the LSG have obvious applications for retail establishments in a physical space, but they can also be applied in abstract spaces like on the web or social networks. In addition, this model can apply to political parties on an ideological spectrum, which has been studied under the guise of median voter theory [Lopez et al., 2007]. Hotelling games have another interesting property, which is that they have a *price of anarchy* equal to $(2n-2)/n$ where n is the number of players and the social optimum is considered to be the minimum score of the n players [Blum et al., 2008]. The concept of a price of anarchy relates the ratio of the optimal average social welfare to the Nash equilibrium values. Therefore, if many Nash equilibria exist, the resulting payoffs of an equilibrium can severely disadvantage any given player, which makes it all the more urgent for individuals to act according to an accurate population model in these settings.

When these games are repeated, one emergent phenomenon is that, depending on the current configuration of players, certain well-known bimatrix payoffs can operate between two of the opposing players in response to a presently fixed third (or nth) player. In particular, when two players are near each other or at the same location, a standoff occurs that resembles the game of Chicken. To illustrate this point, consider the following simple example.

Let there be a simple one-shot game with three players (Alex, Bill, and Carla) and three actions (1, 2, and 3) on a number line, so that 2 is connected to 1 and 3, but 1 is not connected to 3. The utilities of such a game for player i are defined as

$$U(a_i) = I(a_i)/\#(a_i) + d(a_i, a_{\neg i})$$

where I is the identity function, $\#(a_i)$ is the number of agents playing action a_i, and d is a distance function that returns the minimum distance to the "closest" player in the action space identified by the numbers ($i.e. d(a_i, a_{\neg i}) = \min_{j|j\neq i}(|a_i - a_j|)$). Let's imagine that Alex and Bill choose action 3 and Carla chooses action 2. In this scenario, Carla receives $2 + 1 = 3$, while Alex and Bill get 1.5 each. The payoff matrix for Alex and Bill, assuming Carla stays at 2, is displayed in Table 5.2 and demonstrates how this situation can be considered a standoff when repeated. As a result, an analysis of a repeated game like this one must rely on the time horizon factor that players of the game Chicken use when deciding when to back down. See Section 2.4 for more details.

While Alex and Bill are both facing off on location 3, their regret equals $\rho = 2 - 1.5 = 0.5$ every round. The potential gain, on the other hand, is $G = 4 - 1.5 = 2.5$. If Alex moves to location 1 after 5 rounds, we estimate γ at $\hat{\gamma} \leq (0.5/(0.5 + 2.5))^{1/5} = 1/6^{1/5} \approx 0.7$. To repeat the justification for this formula, a learner facing a non-reasoning player would expect at least even

Table 5.1: Player A's payoffs for a three-action location game played on a number line.

A's Action	C1 B1	C1 B2	C1 B3	C2 B1	C2 B2	C2 B3	C3 B1	C3 B2	C3 B3
1	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{3}$	2	2	$\frac{1}{2}$	2	3
2	3	1	3	1	$\frac{2}{3}$	1	3	1	3
3	5	4	1.5	4	4	2.5	1.5	1.5	1

Table 5.2: A bimatrix game resulting from the situation where two players on the same location (3) must decide whether to remain or move. The third player is assumed to be constant at location 2, so there is no benefit to playing action 2. Either player can benefit from action 1, which increases the score from $\frac{3}{2}$ to 2. However, the remaining player benefits more from the change, increasing from $\frac{3}{2}$ to 4. This situation resembles Chicken, where there is a superior equilibrium but there is an incentive to wait for the other player to back down. The three columns on the right show the same game but with the dominated action 2 removed.

A/B	1	2	3	1	2	3
1	$\frac{1}{2}, \frac{1}{2}$	2,1	2,4	$\frac{1}{2}, \frac{1}{2}$	-	2,4
2	1,2	$\frac{2}{3}, \frac{2}{3}$	1,4	-	-	-
3	4,2	4,1	$\frac{3}{2}, \frac{3}{2}$	4,2	-	$\frac{3}{2}, \frac{3}{2}$

odds that the current state of affairs will continue for as long as it has been observed. Therefore, Alex would require a higher γ to justify holding out for the future payoffs (that may not arrive). The important thing is a consistent way to map patience to a particular value, and this method works as well as any other.

This simple game also illustrates the importance of robust beliefs over narrow ones. If Carla were to calculate a strategy using level-based reasoning starting at random L0, she would find that L1 should play 3. At L2, the action choice depends on how Carla picks the likely population of Alex and Bill. If Carla believes the population is composed of 100% L1s playing action 3, then she has no preference over action 1 or 2 as they both get utility 3, as long as she

does not pick 3 with a guaranteed score of 1. However, if she believes that Alex is an L1 but Bill is an L0, then she has a different choice. In this case, action 1 is worth $(0.5+2+3)/3 = 11/6$, action 2 is worth $(3+1+3)/3 = 7/3$, and action 3 is worth $(1.5+1.5+1)/3 = 4/3$. In either case, action 2 is optimal (for level 2). In the first case, the double L1 assumption caused Carla to completely misread the structure of the game, and ignore the crucial difference between action 1 and action 2, which is that action 2 has a higher base score.

5.3 Lemonade Stand Game

The Lemonade Stand Game was introduced as a domain for contestants to try out algorithms for outwitting opponents in a game with no unique equilibria. The LSG demonstrates the interaction complexity that can arise in a game from simple rules [Zinkevich, 2009]. The game is played by three lemonade vendors on a circular island with n beach locations, where typically $n = 12$, arranged like the numbers on a clock. Each morning, the vendors have to set up on one of the beach locations, not knowing where the other vendors will show up. The game is played repeatedly for 100 days and the joint action is observable.

Assuming the beach customers are uniformly distributed with 2 per location and buy their lemonade from the closest vendor (breaking ties evenly), the payoff for the day is equal to the distance to the neighboring lemonade vendors. For convenience, denote $D(A_i, A_j)$ as the distance function between agents i and j on the side with no other agent in between. Then, $R_i^t = \sum_{j \neq i} D(A_i^t, A_j^t)$ except when $A_0^t = A_1^t = A_2^t$ and all players receive 8. The 2009 and 2010 LSG tournaments took this *uniform-density* form.

In game-theoretic terms, the uniform LSG is a 12-action normal form game

on a ring, where the payoff function equals the sum of distances to the right and left neighboring vendor, so that there are no locations inherently better or worse than others. Because there are 24 customers in the uniform case, the cumulative payoff of the three players is 24. The only exceptional formations are when multiple agents conflict by choosing the same action (Collision). If two vendors choose the same action, they split the reward normally assigned in that situation, which is exactly half of the customers. They receive a reward of 6 and create the most favorable condition for the third agent who receives the maximum of 12. If all three vendors choose the same action, each receives 8. Because of this conflict, imitation/cooperation does not have the same interpretation that it would in games without this structure.

The Lemonade Stand Game is a special case of a Hotelling game [Hotelling, 1929], and also an ideal example of competitive collaboration. It appears that players have many repeated turns for observation and experimenting. In reality many matches are settled in the first several rounds, as agents seek to identify partners and non-partners and mutual history is established. Cooperation, however defined, is self-reinforcing. Therefore, strategies in this game put a premium on speed of action over time-intensive data collection when finding optimal actions. This property means that traditional learning methods, like gradient ascent or regret matching tend to be outperformed by very simple rules. Because there are many possible Nash equilibria in the game, it is also unclear which ones are optimal and how to reach them. Our aim is a model that can explain such phenomena and yield strategies that outperform at least the simplest heuristics. An alternative approach [de Côte et al., 2010] to this game identifies a stable equilibrium and classifies agents as leaders or followers according to who initiates the equilibrium pattern. While this strategy

works in some scenarios, in some cases it is possible to identify several levels of leading and following. It also makes no judgments about whether one is superior to the other, or how one might measure that performance. In a population setting, we might also want to learn the probability of a follower in any particular round, and as we will see that is what the meta-reasoning model sets out to do.

The dynamics of this game are particularly interesting because they involve a sense of competition, as the gains of one always have to be compensated by the loss of others, as well as a sense of cooperation, because two agents can coordinate a joint attack on the third. That is, a player able to convince another player to cooperate with it can achieve a higher average score to the disadvantage of the third player. Of course, each player would prefer to cooperate with the "friendlier" player, with the knowledge that any attempts may be used by the other players for their own classification tasks. Ultimately, two players who can work together best will achieve higher scores at the expense of the third, who becomes the "sucker". In this context, collaboration entails dividing up the constant space between two players, leaving the third a smaller overall fraction of the pie.

Because coalitions are easy to form and such an essential aspect of the LSG, this game has unique properties that distinguish it from other purely competitive settings like Rock-Paper-Scissors [Egnor, 2000] or even larger games like the Trading Agent Competition (TAC) [Stone and Greenwald, 2005]. TAC agents must perform a great deal of optimization to identify trading opportunities and estimate price values [Jordan et al., 2010]. However, the inherent symmetry of TAC creates a tendency towards Level-1 type strategies and a resulting equilibrium and away from significant modeling approaches. In the

LSG, the tension between collaboration and competition leads to a central emphasis on reasoning about the decision-making process of other players. That is, LSG agents must construct strategies to either guide others into partnership, or cause them to compete amongst themselves. The structure of the game therefore calls forth a modeling approach, even if much of the model-building or reasoning takes place before the tournament begins.

Figure 5.3 shows an overview of the key strategic patterns in the LSG. Each agent has to choose an action every round, and the simplest move is to choose the same action as before (Repeat), matching the same base strategy as before. Imitation can be understood through an abstracted game between two of the players, whereby the smaller game resembles coordination such that players agree to spread out from each other. In the extreme case, an Equilateral arrangement leads to all players getting an equal score of 8. In practice, it is easier to spread apart from a single opponent, so that the imitation is more like a mirror image than exact duplication, leading to the Across action. Once two agents coordinate on the action Across, they will share 18, relegating the third agent to 6 regardless of the action it chooses. As an illustration of its simplicity, □ must only find a predictable player ○ and use the action opposite to it. ○ can be completely oblivious as long as it is predictable (say, a pure Repeat player).

If an agent finds its opponents in a consistent Across pattern, it will lose unless it can entice at least one opponent to break formation. However, achieving this outcome requires a member of the coalition to deviate from the prior stable state, thereby risking retaliation as a result. It is also unclear what the series of steps are that would cause a player to move. For this reason, we can say that Across is a stable equilibrium in this game.

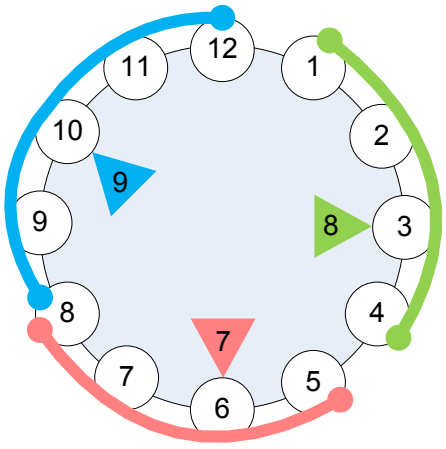


Figure 5.1: Sample actions and associated payoffs, computed by nearest distance.

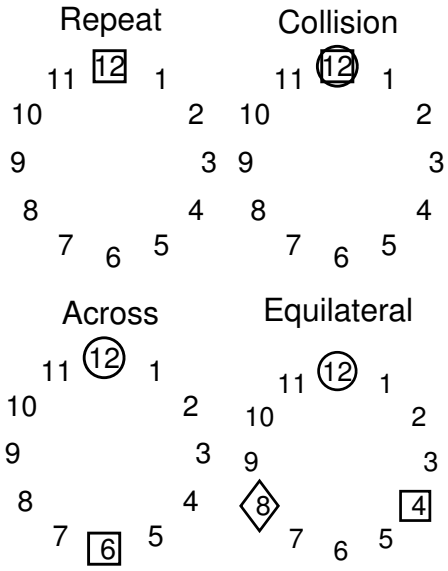


Figure 5.2: This figure depicts key strategic patterns of the Lemonade Stand game. Each of the diagrams refers to a (partial) joint action, and similarly a strategic move by □, expecting opponents to play ◇ and ○. As the domain is on a ring, the patterns are rotation and reflection insensitive.

5.3.1 Generalized LSG

A non-uniform, or generalized, version of this game results when we remove the constraint that the customers are evenly distributed. In the 2011 and 2012 iterations of the LSG tournament, game payoffs were randomly generated by independently drawing either 6, 12, or 18 customers for each location. See Figure 5.3 for a visual representation. This variation creates $3^{12} \approx 5 * 10^5$ possible utility functions. With a learning lifetime orders of magnitude smaller than this figure, no learning algorithm can expect to build case-specific models for each payoff configuration. This limitation holds true especially when considering the multitude of actions that can be played in the numerous rounds by the various agents, so that many observations would be required for each instance. Also, we see that this type of distribution typically leads to dense clusters of customers appearing in certain locations of the action space. The resulting asymmetry means that certain heuristics, such as finding the position with the highest surrounding densities, will outperform other reasoners who ignore these densities, even agents that do well in the uniform case.

5.4 Level-k Meta-reasoning in a γ-model

In this section we apply the reasoning γ-model to the LSG in its uniform and asymmetric non-uniform (non-identically iterated) versions. Because the symmetric game restricts the number of outcomes, the reasoning is drastically simpler, but the issue of time horizon remains. In the case where new payoff functions are randomly drawn each new game, the reasoning takes a more prominent role, and the temporal questions also become richer.

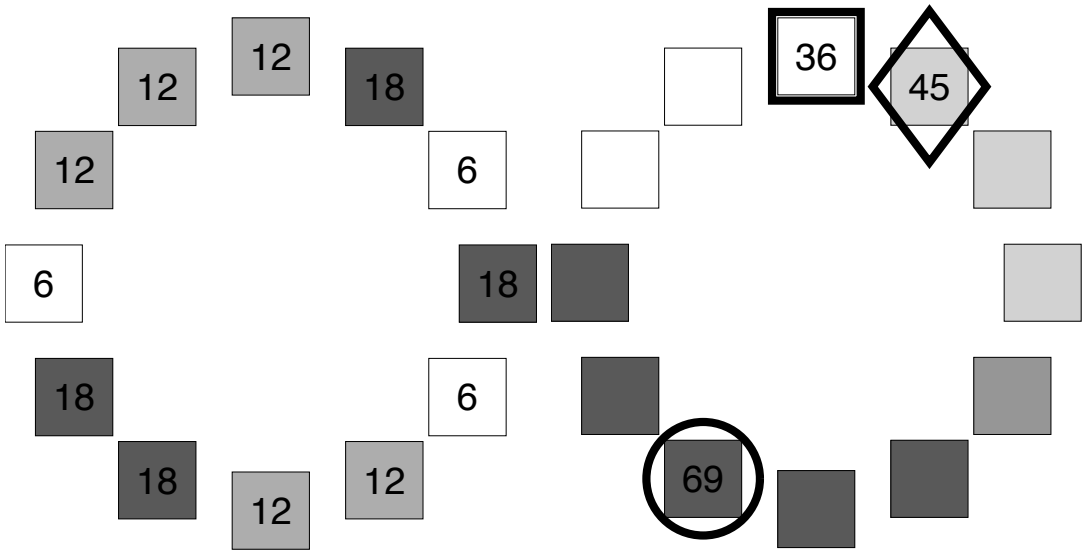


Figure 5.3: In the Generalized Lemonade Stand game, customers are unevenly distributed, with new distributions each new match. Each beach has 6, 12 or 18 customers with equal probability, and each customer gives a profit of 6 (left, darker colors represent higher values). Given lemonade stands ◯, □, and ◊, the utility for each can be computed (right, colors show to which stand customers go).

5.4.1 Uniform game levels

In the symmetric, uniform game, the first action is trivial because no actions are better than the others. As a result, the agents are assumed to play randomly on this step, and then the game truly begins at time $t = 2$ with knowledge of the location state (all agent actions). A reasonable starting point for the state-based level 0 (L0) strategy is the uniform random action. However, because this game is repeated, there is a spectrum of randomness, proceeding from complete randomness every turn to the opposite extreme of no randomness at all, or constant action. We can make use of the time parameter here in order to create a sense of continuity with the higher levels, even though L0 takes no strategic action. For now, we assume that the L0 strategy is expected to move randomly within τ time steps. One way to represent τ is as the point where the sum of a discounted geometric series reaches half of its total value.

In this instance, the base of the exponent can be interpreted as the probability of remaining in the same location. Therefore, τ is equivalent to the number where the following inequality is true for some value of ϕ, assuming $T >> \tau$:

$$\begin{aligned}
\sum_{t=0}^{\tau-1} \phi^t &\geq \sum_{t=\tau}^{T} \phi^t \\
\frac{1}{1-\phi} - \frac{\phi^{\tau-1}}{1-\phi} &\geq \frac{1}{1-\phi} - \frac{\phi^T}{1-\phi} - \frac{1}{1-\phi} + \frac{\phi^{\tau-1}}{1-\phi} \\
\frac{1}{1-\phi} - \frac{\phi^{\tau-1}}{1-\phi} &\geq \frac{\phi^{\tau-1}}{1-\phi} - \frac{\phi^T}{1-\phi} \\
1 - \phi^{\tau-1} &\geq \phi^{\tau-1} - \phi^T \\
1/2 &\geq \phi^{\tau-1} \\
\phi &\leq \frac{1}{2^{1/\tau}}
\end{aligned}$$

The result of this inequality means that if $\tau = 1$, a value of $\phi = 0.5$ yields equal sums, so that the expected wait time to move is 1. It is not important how we measure the degree of constancy of the L0 strategy types, as long as higher means more constant, as ϕ does here. In fact, ϕ turns out to be equivalent to the probability of repeating the last action under the above definition (so that $1 - \phi$ is the probability of motion). When examining strategic players who are deciding whether to delay gratification, the calculation is as above but the ϕ is still proportional to the probability of repetition.

The ensuing level 1 reasoner operating under a given belief of the opponents' ϕ should be aware that the two players may have different degrees of constancy, even though it is impossible to discern immediately without experience. L1 could therefore employ in-game learning and even some amount of generalization to optimize its approach, but it will not attribute any agency to its competitors. L1, then, takes its actions with no accounting for how those

actions change the state of the game for other agents. As a result, we can speak of values of ϕ for these strategies, but the value calculation will always return an action that is best for the next time step.

In the case of the uniform LSG, it is a simple calculation to see that the optimal L1 strategy, a best response to semi-random opponents, maximizes the distance from the other agents, weighted by the degree of constancy of each agent. If A_B is the action of player B and A_C is the action of player C, and ϕ_B and ϕ_C are the probabilities that B and C remain constant, respectively. They randomize with probability $1 - \phi_B$ and $1 - \phi_C$. then the optimal action for the third agent I is $A_I* = W_B\mathsf{Across}(A_B) + W_C\mathsf{Across}(A_C)$ where $W_B = \frac{\phi_B(1-\phi_C)}{\phi_B(1-\phi_C)+\phi_C(1-\phi_B)}$ and $W_C = 1 - W_B$. This formula results from basic function maximization, and as such exploration plays no useful role in determining the solution. It is worth noting that if $\phi_B = 1$ and $\phi_C < 1$, then $W_B = 1$ and the resulting strategy is for I to stand Across from B. This stable equilibrium has been shown to benefit the two players in the Across partnership (here, B and I) to the disadvantage of the third (C) for as long as it lasts.

To form the second level, we return to the earlier Chicken analysis. In the presence of an L1 agent, it is profitable for L2 to appear to be as constant as possible, so that the L1 moves far from L2. However, one can imagine a scenario where the random initial actions lead two players (B and I) to start on the same location, causing a conflict and a period of experienced regret since there may be a way to improve utility. Notice that if B moves to the higher half of the board, perhaps employing the Across strategy from I, there is a temporary boost in utility. This change in state will cause C, if it is now the sucker in the relationship, to set up Across from the I who remains in the initial position. I has seen its score improve significantly simply by out-waiting its opponent.

Table 5.3: A bimatrix game resulting from the situation where two players on the same location (3) must decide whether to remain or move. The third player is assumed to be constant at location 6 unless it receives the minimum payoff (after one of these players moves). In the uniform case, there is no benefit to moving first in this scenario.

R/C	Stay	Move
Stay	$\frac{1}{1-\gamma}(6,6)$	$(\frac{9}{1-\gamma}, 3+\frac{6}{1-\gamma})$
Move	$(3+\frac{6}{1-\gamma}, \frac{9}{1-\gamma})$	N/A

For this reason, we see how γ re-enters the situation in a way that directly impacts the final outcome of the game. See Table 5.4.1 for the resulting payoffs. It is clear that if the payoff in the current state is greater than six, there is no benefit to moving. If less than six, then a move should occur based on the observed (or expected) discount. A payoff of exactly six should lead to an infinite waiting period, because there is no long-term gain for a new action. On the other hand, if there is some positive likelihood that another player might act randomly as a result of a move by this agent, then it might be worth doing so depending on the relative level frequencies.

For the rest of this analysis, we will assume that in the uniform setting, reasoning strategies play the Across action with varying degrees of constancy, parameterized by γ, until Level 2 is reached, representing completely constant. So, for shorthand, a player with $\gamma = 0.5$ is considered Level 1.5, $\gamma = 0.9$ is considered Level 1.9, and so on. In effect, the estimated value of γ takes the same form as the case where ϕ is computed using the sum of geometric series. The main difference here is that the sum is the discounted rewards received, whereas earlier the rewards are ignored. Section 2.4 details the process of comparing past regrets with future expected rewards, given the expected response of an opponent.

5.4.2 Meta-learning Level-based Model Distributions over k and γ

The quantal level-k model (Section 2.2) provides a method for producing the strategies at each level given a precision value λ. To produce an estimate of the strategy executed by an agent, observe its action and the probability that each strategy would have selected it. Using this probability, we arrive via Bayes at the likelihood that this agent is acting according to this strategy, and therefore that it reasons at the level that would produce that strategy. The normalized likelihoods for all reasoning levels give an estimated meta-reasoning model of this agent. As described in the extended meta-reasoning model for populations (Section 2.5), the two sets of observations that determine the distribution over levels are initial actions and subsequent actions (made in the context of a state). Because the number of these observations may tend to be small in a single instance of the game at hand, we must gather data over many matches, with many different payoff functions, and hope that patterns arise. There are several ways to build models from this data. In the comprehensive survey and analysis done with human data [Wright and Leyton-Brown, 2010], the authors used a maximum likelihood method to make predictions of future behaviors. For our purposes, this predictive ability may not be enough because we would like to be able to generate strategies in response to a meta-learned model. A further difficulty is that limits on available computation time constrains the ability to re-compute entire distributions for each new decision. Therefore, a compromise solution is to simply average together the resulting likelihoods for every instance observed.

5.4.3 Planning with a Meta-learned Level-based γ-Model

Once the reasoning level distribution is estimated, we turn to finding optimal responses to the underlying population. The process used to accomplish this task is basically the mirror image of the learning mechanism. Given a new LSG instance, we discover the initial action levels, up to the third. In our case, three is sufficient because the reasoning only needs to handle three agents. It so happens that by choosing one action from each of the three level-based reasoners will yield a very good approximation to the likely final state of a series of best response steps, starting from any initial position. Note that this assumption amounts to a prior belief that the agents select their action with high precision, such that $\lambda = \infty$. In the event that the space cannot be divided in this way, there are probably a high number of equally good starting points distributed more or less evenly around the circle. We will suffice to call these final three actions *stable* for lack of a better term.

With these three candidate actions limiting the search space, we can easily find the scores assuming that there is one agent at each location, giving a ranking of these three, which will not necessarily correspond to the level that first produced them. (See Figures 5.4.3 and 5.5 for an example visual explanation.)

To review the workings of the modeling algorithm (Algorithm 1), each time a decision is observed, whether it is the first observed action or a subsequent change in action, this action is assigned a strategy vector σ of length K, where K is set to 4 in this case. To estimate the posterior probability that an observed agent i is operating at a given level, first calculate the likelihood of the observed action, for each level k, starting with $\pi_0(a_i) = \frac{1}{|A|}$:

$$\pi_k(a_i) = \frac{e^{\lambda u_{i,k}(a_i)}}{\sum_{a_i'} e^{\lambda u_{i,k}(a_i')}}.$$

Then. the probability that the agent reasons at level K can be computed using Bayes rule:

$$\hat{\sigma}_{i,k} = \frac{\pi_k(a_i)}{\sum_{k'} \pi_{k'}(a_i)}.$$

The same procedure can be applied to the state-based level strategies, but substituting the repeat action for random in Level 0.

It is unlikely that the three players will end up picking a different one of the three stable actions, but it depends on their relative reasoning level. More likely is that at least one of our two opponents will choose the highest ranked action, as all else equal it would be the best one. Fortunately, this situation is ideal for us to analyze using the regret-based approach mentioned in Section 2.5.

The amount of time that has passed before this move provides evidence about how long this agent would expect to wait in the future, as a function of the relative benefit it receives by moving to a new location. That is, if we have an estimated discount factor (γ) for our target population, then we know how long to expect an opponent to wait before switching to the lesser stable action, leaving our agent in command of the best one. If γ is sufficiently low, it will be worth the initial cost to be the lucky agent to wait to collect the higher score. However, if the population has demonstrated high γ and therefore a lot of patience, then it may in fact be optimal to take one of the lower ranked stable actions, and hope that our opponents end up fighting Chicken-style over the highly ranked location, to our agent's benefit. The parameterized model accounts for these opposing forces and combines them to compute the estimated values for each of these stable points over an entire game, prior to the start of

the game. Although we have described this process in words here, our agent was built according to the level-k γ-model. It is able to quantitatively discover this solution automatically, and thus fully implement a meta-reasoning model-based response.

In essence, this process uses data from prior games to construct a distribution over the strategy space given new payoff function. The shifting densities put the agents in a shifting landscape, but crucially the rules and structure of the game remain in place. Thus, the decision process is an ideal test case for the ability to transfer knowledge through a meta-reasoning framework. We now turn to evaluating this modeling approach through the performance of an agent that uses this method.

5.5 Generalized LSG Experiments

The mark of a successful model is to predict outcomes, and the state-based multiplayer IBR framework presented here is no exception. In this section, we test the correlation between the various model parameters and final competition scores. Our results can be summarized in Figure 5.7. We took the level distributions for each agent and found the average estimated level for both initial action and state action in this manner. Then, we examined the history leading up to a change in state and recorded the regrets of the participating agents, and thereby arrived at bounds on γ. This step gives three total model parameters, including the discount factor. All three are highly positively correlated with final performance. If we combine all three values to predict the score, we reach a 0.83 coefficient of correlation and over 0.9 for prediction of rankings. The initial action level has highest performance at L2, on account of other agents playing sub-L2 strategies. Recall that the level-k model prescribes

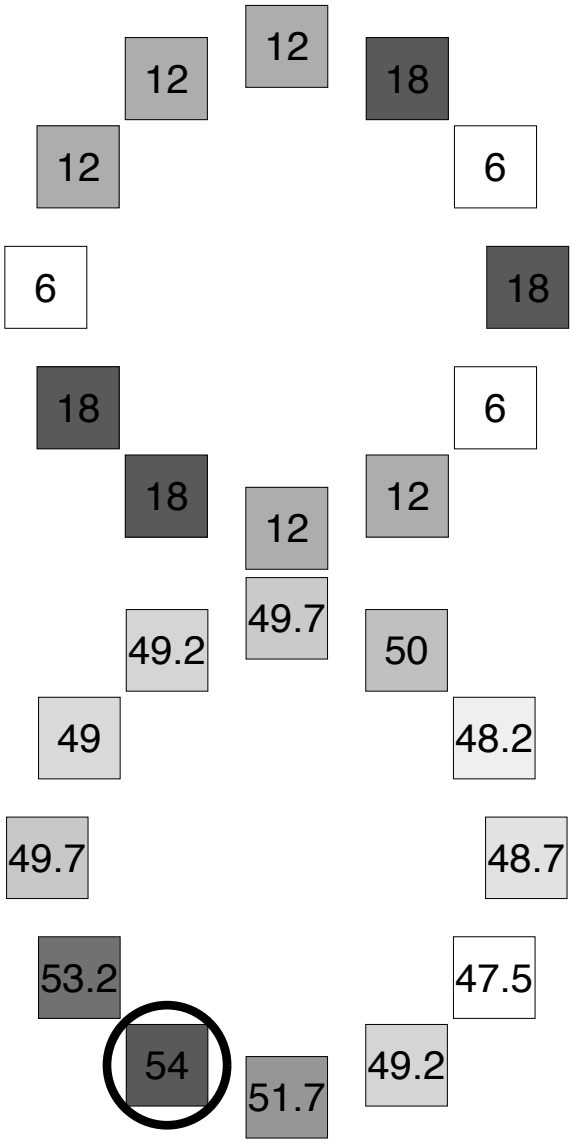


Figure 5.4: Top: Sample payoffs in the Generalized LSG (darker colors represent higher values). Bottom: Expected payoffs for reasoning Level 1, where darker colors represent higher values. The circle denotes the best available action against two random players (L0).

responding to the average degree of sophistication in the population, and not necessarily the highest possible strategy calculation with the most reasoning.

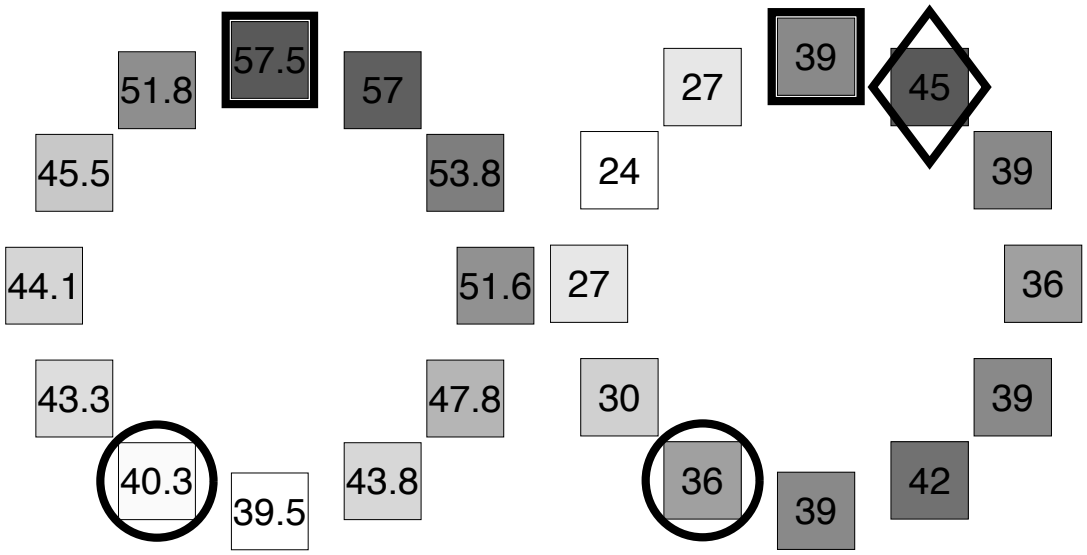


Figure 5.5: Expected payoffs for reasoning Level 2 (left) and 3 (right): darker colors represent higher values. The □ denotes the best reply to one L1 player ◯ and one uniform random player (L0). The ◊ denotes the best available action against one L1 player ◯ and one L2 player □.

Table 5.4: 2009 LSG Tournament results including two agents inspired by the γ-Model hierarchy (italicized). The winners are in bold. Level 0.83 would correspond to a player that **Sticks** with probability of 0.83, but is random the rest of the time. An agent that would qualify as Level 1.63 would mean that a player **Sticks** when in an advantageous starting position. When its initial spot is less beneficial than it is constant with probability equal to 0.63, and the rest of the time moves Across from another player, preferring the more constant one. In cases where it is already **Across** from a player, it remains in place by choosing the same action.

Rank	Strategy (Affiliation)	Score	Error	Est. Level
1.	*Meta-reasoning Model (new, adaptive)*	8.62	± 0.0069	1.63
2.	**EA^2 (Southampton/Imperial)**	8.58	± 0.0069	1.63
3.	CoOpp (Rutgers)	8.55	± 0.0055	1.38
4.	ModifiedConstant (Pujara, Yahoo!)	8.52	± 0.0076	1.93
5.	MyStrategy (Waugh, CMU)	7.97	± 0.0087	0.96
6.	ACT-R (Reiter et. al., CMU)	7.95	± 0.0086	0.96
7.	GreedyExpectedLaplace (Princeton)	7.57	± 0.0086	0.83
8.	FrozenPontiac (U Michigan)	7.44	± 0.0075	0.63
9.	GregStrategy (U Texas Austin)	6.82	± 0.0054	0.13

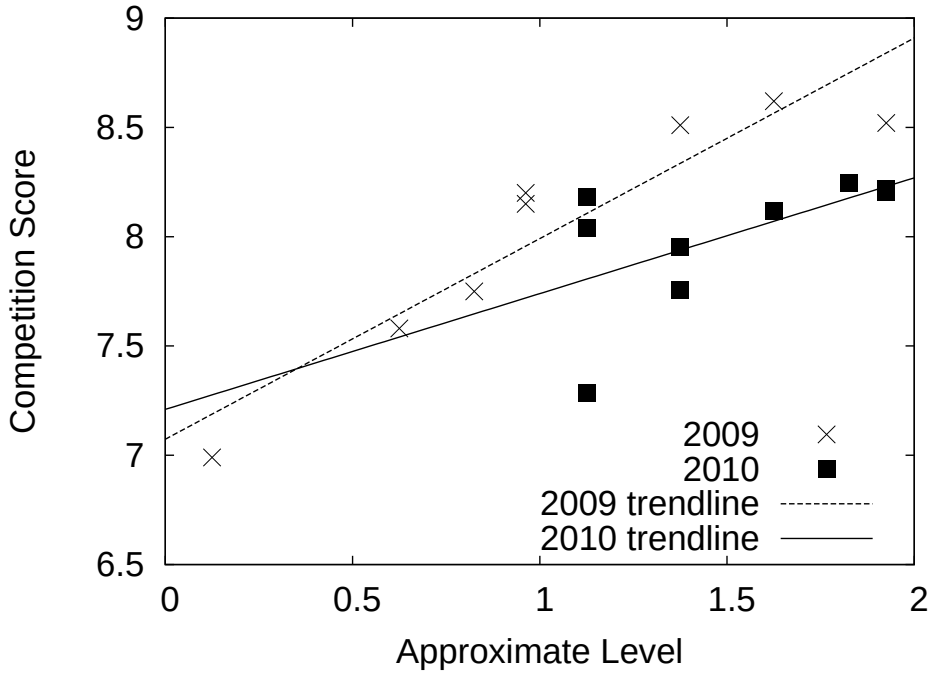


Figure 5.6: Estimated levels of competitors in two Lemonade Stand Game tournaments. Both sets of agents show positive correlation between reasoning and performance. R^2 values are 0.77 for 2009 and 0.34 for 2010. The more recent agents show a shift to higher reasoning levels, as well as a compression of scores.

5.5.1 Experiments using 2009 and 2010 agents

The levels of LSG, while useful, are theoretical constructs. Nonetheless, the basic elements of this account arose in a group of agents developed independently. This section shows the viability of the level-based analysis by applying it to the two rounds of open LSG competitions, one in Dec. 2009 and the other in Dec. 2010. The submitted strategies were a diverse collection. No two were alike and ranged from complete uniform action to near constant, to Across-seeking and initiating, and many in between.

Table 5.5: 2010 LSG Tournament results including an agent inspired by the γ-Model hierarchy.

Rank	Strategy (Affiliation)	Score	Error	Est. Level
1.	*Meta-reasoning Model (new, adaptive)*	8.37	± 0.0099	1.98
2.	Waugh (Carnegie Mellon)	8.27	± 0.0094	1.93
3.	Matchmate (GA Tech)	8.22	±0.0095	1.13
4.	ModifiedConstant (Pujara, Yahoo!)	8.17	±0.0097	1.93
5.	Shamooshak (Alberta)	8.13	±0.0094	1.25
6.	**TeamUP (Southampton/Imperial)**	8.12	± 0.0099	1.83
7.	Collaborator (Rutgers)	8.10	±0.0105	1.38
8.	GoffBot (Brown)	8.00	±0.0108	1.13
9.	Meta (Carnegie Mellon)	7.72	±0.0102	1.38
10.	Cactusade (Arizona)	7.21	±0.0085	1.13

Tournaments run on both sets of agents demonstrate the power of an adaptive strategy. Tables 5.4 and 5.5 show the results including a new adaptive agent with a discounted level-based meta-reasoning model of its opponent population. In both cases, it outperforms the others in the test suite.

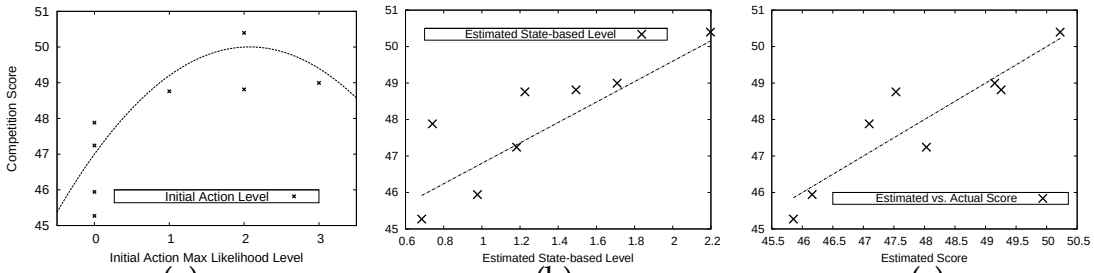
To apply the model to real agents, we would like to classify each strategy by level or as a hybrid between levels. If our level-k γ-model is a good fit for LSG, populations consisting of agents that correspond to a similar mix of levels should behave, and score, in roughly the same way as their idealized counterparts. Since each level has its unique strengths and weaknesses, performance depends on the makeup of the population and specifically the relative frequency of each level. For the purposes of this paper, we classify a strategy by inspecting how it scores against idealized strategies from each of the levels we identified. See Figure 5.6 and Tables 5.4 and 5.5 (right hand side) for these estimated levels. We ran the submitted agents against strategies over various values for the relevant parameters, such as $\phi_0, \phi_1 \in [0, 0.5, 0.75, 0.9, 0.95, 1.0]$. Again, the decimal values for L1 correspond to the probability of players repeating their action when they receive less than the security value (remaining

constant otherwise), while the probabilities are not conditioned on this state. Using the derived strategies as benchmarks to compare to, we take the squared difference between unknown agent and level representative and find the smallest difference between two adjacent scorings, say Level 1.95 and 1.975.

The rankings of the players in both tournaments provide a rough correlation to the amount of reasoning. The bottom half of the 2009 performers act like the base assumption strategies. The top half behave like those derived in the higher levels of the γ-model model. From the 2010 data set, we find that—on average—reasoning has shifted up a level. Players identify the Across position as a goal state, but the top performers are more patient to get there, which implies more reasoning (or forward-thinking) according to the model.

5.5.2 Experiments using 2011 and 2012 agents

The experimental data presented next is derived from the submitted set of agents from the July 2011 and 2012 Generalized LSG tournaments, which implemented the LSG with varying customer distributions. We have two goals in the following analysis. The first is to show that the features resulting from the derived level-based model accurately predict performance in the tournaments. This claim is confirmed by the more advanced strategies under this definition being among the winners. Second, we would like to demonstrate how this level data can be utilized to efficiently determine the distribution over types that are present in an ongoing series of games, and subsequently outflank them. This goal was of interest for the 2012 competitions, where agents had the chance to observe opponents in action over many games, albeit anonymously, and respond as they wished. As a side point, it has been observed that in LSG's current form, there is not much opportunity for modeling and responding to



(a) (b) (c)

Figure 5.7: A comparison of the final 2011 competition score with the maximum likelihood level ($R^2 = 0.73$), estimated state-based level ($R^2 = 0.72$) and estimated score ($R^2 = 0.83$). (a) The best fit over the data points is a quadratic function because most of the agents are L1 or less, which means that L3 does worse than an L2 strategy in this population. (b) Average likelihoods of each level are computed and normalized, and then each level is weighted by the corresponding likelihood. The Spearman's rank coefficient is 0.93. (c) The estimated score is output by the final model as a function of initial level, state-based level, and discount factor. The Spearman's rank coefficient is 0.9. The trendline shown is X=Y.

Table 5.6: Results of an internal experimental match-up including the top four agents of the 2011 Tournament and an agent constructed using the meta-reasoning modeler. Average score over 100 repetitions of each match shown.

Ranking	Agent	Score	Error
1	*Meta-reasoning Model (new, adaptive)*	51.61	$\pm$ 0.39
2	Rutgers	49.36	$\pm$ 0.37
3	Alberta	47.63	$\pm$ 0.39
4	Harvard	47.60	$\pm$ 0.32
5	Brown	43.10	$\pm$ 0.34

opponents within games, as matches are often settled within a small number of moves. Therefore, any agent that possesses an accurate model of behaviors will have a great advantage when it comes to planning a strategy in matches with previously unseen density functions.

In Table 5.7, we show the performance of our agent, the Full-Model agent, against the top four challengers in the 2011 tournament. The best submissions are used since other strategies have been observed to emulate them in time for the next competition [Zinkevich et al., 2011], and we would like our agent to

Table 5.7: Results of an internal experimental match-up including all agents of the 2011 Tournament and an agent constructed using the meta-reasoning modeler. Average score over 100 repetitions of each match shown.

Ranking	Agent	Score	Error
1	*Meta-reasoning Model (new, adaptive)*	50.48	$\pm$ 0.42
2	Rutgers	50.06	$\pm$ 0.33
3	Harvard	49.21	$\pm$ 0.35
4	Alberta	48.83	$\pm$ 0.30
5	Brown	48.40	$\pm$ 0.36
6	Pujara	47.27	$\pm$ 0.39
7	BMJoe	46.95	$\pm$ 0.37
8	Chapman	45.46	$\pm$ 0.35
9	GATech	45.03	$\pm$ 0.31

Table 5.8: Results of the final 2012 Tournament, which includes an agent constructed using the full state-based γ hierarchy. Average score over 100 lifetimes of 100 matches shown.

Ranking	Agent	Score	Error
1	Rutgers (*Meta-reasoning Model*)	52.33	0.14
2	Waterloo	48.92	0.17
3	Alberta	48.13	0.16
4	Harvard	46.30	0.12
5	Sydney	45.33	0.10

perform well against likely future competitors. This agent runs the model to estimate the game-length values of the best starting points, and selects the best of these accordingly. Once the game has begun, it switches into state-based mode and makes regret-informed decisions using an internal value of γ that may be adjusted based on observations from the population.

Table 5.8 shows the results of the fourth and final tournament in 2012, where the γ-Model agent detailed above was submitted by the Rutgers team and won a convincing victory. The power of this modeling/planning technique is demonstrated by the fact that the margin of victory of about 7% was over three times as large as in previous competitions. Clearly, if the predictions of opponent play in each succeeding game were inaccurate, then the method

would recommend actions that would not pay off as well as the results indicate.

5.6 Conclusion

As the experiments demonstrate, a meta-reasoning model that includes reasoning ability and future discounting in a population of strategies is a superior method for predicting and responding in a highly strategic scenario. A base set of action primitives including randomness, repetition, and imitation provides a strong starting point for constructing the hierarchy. The results from 2011 and 2012 show that this type of meta-reasoning model performs well at transferring experience to new, previously unseen payoffs. In particular, the modeler can apply its experience to a new setting through a process of forecasting future behaviors of the model it has learned. This generalized algorithm uses the trained parameters to generate new dynamic strategies in response to a payoff function and the likely states generated by the population model.

Chapter 6

Empirical Agent Based Models of Cooperation in Public Goods Games

In previous chapters, I have focused on predicting the behavior of artificial agents for the purpose of building a modeler agent that can do well against them. Here, I use prediction for another purpose; namely, to evaluate a suite of models to select the best one for a different set of experiments. Additionally, the historical data is not artificially generated, but gathered from a set of online public goods games experiments with human participants. I then use the models trained on this data set to explore the parameter space in a range of scenarios that the original data does not cover.

First, I present how agent-based models have traditionally been used in the literature and why it is important to empirically validate such models, and in the process defining Empirical Agent-based Models (EABMs). Section 6.2 contains selected related work in the field of ABMs and public goods experiments. In Section 6.3, I detail the experimental setup and data set. The next two parts, Sections 6.4 and 6.5, show how to apply the meta-reasoning model from Chapter 2 to this domain, both deterministically (a point prediction) and stochastically (distribution over predictions). These sections also provide evaluations of the established models, first on an individual prediction basis, and then at the level of population distributions. Section 6.6 presents the results of simulations of the trained EABMs in new domains like larger networks with

complex structure, and Section 6.7 concludes.

6.1 Introduction to Empirical Agent-based Modeling

Agent-based models (ABMs), also sometimes called "individual-based models" or "artificial adaptive agents" [Holland and Miller, 1991], constitute a relatively recent approach to modeling complex systems—one that stakes out a middle ground between the highly formal but also highly abstracted approach of traditional mathematical models—emphasizing analytical solutions of algebraic or differential equations—and the richly descriptive but also ambiguous and imprecise approach of intuitive reasoning [Bonabeau, 2002]. ABMs typically assume the existence of discrete agents, whose behavior is specified by rules that depend on the states of other agents, as well as some arrangement of interactions between the agents, where both the agent rules and the interaction patterns can vary from very simple and abstract—as in cellular automata—to highly complex and realistic. On the strength of their flexibility and realism, ABMs have been extensively deployed over the past thirty years to model a wide range of problems of interest to social scientists, including neighborhood segregation [Schelling, 1978], organizational problem solving [Lazer and Friedman, 2007], cooperation and conflict [Axelrod, 1984], opinion change [Deffuant et al., 2000], cultural evolution [Axtell and Epstein, 1996; Axelrod, 1997], and political state formation [Cederman, 1997].

The generally greater complexity of ABMs, however, also requires the modeler to make potentially many assumptions regarding (a) the amount and type of information possessed by the agents, (b) the manner in which that information is processed, and (c) the rules governing the interactions between agents.

Traditionally, these modeling choices have been made on the grounds of intuitive plausibility, rather than on empirical accuracy. This choice reflects, in part, the philosophical position of ABMs researchers who have viewed ABMs as thought experiments intended to explicate theories and explore causal mechanisms, not as forecasting engines [Axelrod, 1997; Macy and Willer, 2002]. In recent years, however, the idea of grounding modeling assumptions on empirical observations of human behavior has begun to attract attention [Janssen and Ostrom, 2006; Heckbert et al., 2010]. There are two main reasons for this new emphasis, one technological and one philosophical. The advent of large online data sets has enabled the construction of behavioral models of individuals at a level of detail never before possible. The deeper issue is that even plausible and apparently innocuous assumptions about agent behavior can turn out not only to be mistaken but also critical to the emergent behavior of interest. Even if the goal of agent-based modeling is theory explication not empirical accuracy per-se, a certain amount of empirical accuracy may be necessary in order to avoid spurious conclusions.

To illustrate the potential problem, consider two provocative papers from the organizational learning literature that used ABMs to explore the impact of "network efficiency" on collective problem solving [Lazer and Friedman, 2007; Fang et al., 2010]. Both papers reached the same surprising and counterintuitive conclusion: that under certain general conditions, inefficient networks—those with long path lengths and slow diffusion times—should outperform efficient networks when nodes were searching for a global optimum in a fitness landscape. The explanation was that the models assumed that when an agent's neighbor discovers a superior solution, the agent would copy their neighbor, hence efficient information diffusion led to herding onto local optima,

whereas inefficient diffusion allowed for greater exploration, hence higher likelihood of discovering the global optimum. Subsequent human subjects experiments [Mason and Watts, 2012], however, have found that human agents do not copy with a fixed probability, and in general do much less copying than the agents in the models. Moreover, the greater exploratory tendencies of human subjects vis-a-vis the artificial agents, turns out to eliminate the putative benefits of inefficiency, while the costs (slower diffusion of information about successful solutions) remain. The result is that in experiments, efficient networks always outperform inefficient networks, even under conditions chosen specifically to favor exploration [Mason and Watts, 2012].

This chapter articulates an approach labeled "empirical agent based modeling" (EABM), in which candidate models are first trained and evaluated on data from human-subjects experiments, and then deployed in the same way as traditional ABMs to explore regions of the parameter space outside of those in which the original experiments were conducted.[1] Adopting this data-oriented approach means motivating and evaluating our models almost exclusively in terms of how well they predict observable player actions[2] , ignoring obvious criteria such as psychological interpretability or theoretical plausibility. As a consequence, the models do not map in a straightforward fashion to conventional agent-based models, which are often motivated by strategic or psychological arguments; however, as we will indicate, a number of these models are in fact behaviorally equivalent, and therefore are effectively included in the following analysis. Finally, although the idea of empirical validation of

[1] Note that agent models could also be evaluated on data from non-experimental sources such as role-playing games, participant observation, or surveys [Janssen and Ahn, 2006].

[2] Where predictive performance of competing models is close we can also place some weight on parsimony.

ABMs is general, here we examine the approach in the specific context of cooperation in public goods games, an important problem in social science, and to agent-based modeling in particular [Axelrod, 1984, 1997; Macy and Willer, 2002], Critically, it is an area in which recent large-scale human subjects experiments [Suri and Watts, 2011; Wang et al., 2012] have made the appropriate data available for training and testing EABMs.

6.2 Related Work

Although empirical evaluation of ABMs is a topic that has received relatively little attention, a handful of attempts have been made, also in the context of games of cooperation. The earliest, by Deadman 1999, attempted to fit data from previously conducted common pool resource experiments with a reinforcement-learning model. According to Deadman, the resulting aggregate behavior was "similar" to the empirical data, but no quantitative evaluation was performed and no alternative models were considered. Subsequently, Castillo and Saysel 2005 developed a system-dynamics model of player behavior also in common pool resource games, and compared its behavior with data from field experiments involving fisherman and crab hunters from the Providence Island of Columbian Caribbean Sea. The authors assessed their model's validity predominantly in terms of its ability to display behavior that is consistent with theoretical expectations (e.g. its sensitivity to key parameters), not empirical data. Nevertheless, they showed that it was possible to find parameters for which the model could approximately replicate observed aggregate contributions, where again no quantitative evaluation was performed and no alternative models considered.

There are other well-known examples where modelers have aimed to replicate historical phenomena, such as the Artificial Anasazi Project [Axtell et al., 2002]. Again, because only a small amount of aggregate data is available, such as the number of farms in the valley, it is difficult to evaluate the results of these experiments by how closely the model captures the agent behavior. In other research studying more recent economic data, it is notable that many of the proposed ABMs also focus on reproducing aggregate measures such as GDP [Fagiolo et al., 2007a]. The problem with training the various parameters of ABMs to capture these aggregate observations is that it is very susceptible to overfitting those parameters and could produce wildly divergent results in a situation where the conditions are slightly different. Finally, and most similar to the current work, Janssen and Ahn 2006 fitted an experience-weighted attraction (EWA) model of learning [Camerer and Ho, 1999] to data from two earlier experiments. Fitting separate models to individual players, they identified 8 player "types," defined in terms of their best-fit parameter values, that accounted for the vast majority of the sample population.

The contribution of the following analysis differs from, and builds upon, this previous work in three key respects:

1. *Prediction.* Whereas previous attempts have emphasized plausibility and interpretability of the candidate models over predictive accuracy, this work takes a machine-learning approach, similar to that adopted by Wright et al. 2012, in that it introduces a basket of models and compares their predictive performance on out-of-sample test data. Note that this approach does not rule out cognitively plausible models—indeed, as indicated below, a number of conventional models of cooperation, including well-known strategies like Tit-for-Tat, are behaviorally consistent

with the ones proposed here. Because the primary goal is predicting behavior, however, we can be less concerned with the underlying cognitive model than with the behavior itself. Even so, the structure of the best models seem to imply the existence of certain psychological biases, such as inertia and following the crowd.

2. *Evaluation.* The model performance is evaluated more rigorously than previous work, first on average contributions over time, and second on the full round-by-round distribution of contributions—a far more challenging requirement.

3. *Application.* Finally, going beyond simply fitting a model to the experimental data, a selected model is then deployed to explore parameter regimes beyond those covered by the experimental design. In other words, this approach preserves the "ABM as thought experiment" tradition of agent-based modeling, but attempts to ground it in agent rules that are calibrated to real human behavior within some domain.

With these elements, the Multiagent Cycle described in the first chapter is in full effect. The experimental data, detailed in the next section, was gathered to investigate the relationship between behavior in public goods games on social networks. This domain is interesting in part because observations have long contradicted the predictions of game theory, which is that people shall decline to contribute to the common good. Another reason is that there is still no consensus about how to model behavior in this realm, with possible explanations ranging from reciprocation to altruism to delayed best response.

Although the original hypothesis laid out by Suri and Watts 2011 was not confirmed, the data holds deeper value when it comes to this latter question,

because it contains multiple observations of the same people in this game under various circumstances. Finding a model that is maximally predictive is imperative when it comes to using it for future applications. For this reason, the model-building step in the cycle is key because it connects what is known to what we would like to know.

The later sections will attempt to complete the cycle by using the learned models in simulated domains that are too big to gather reliable human data for a number of reasons such as coordination, technology, and cost. However, once we can trust the underlying generative model due to exhaustive testing, the results of such explorative simulations will have a firmer basis for hypotheses that emerge out of them.

6.3 Background on Experimental Setup and Data

Before defining and analyzing the models, let us first briefly describe the experiments used to gather the data, which were conducted using Amazon Mechanical Turk[3] (AMT), and were originally reported by Suri and Watts 2011 (hereafter referred to as SW). The experiments were a variant of a linear public goods game [Ledyard, 1995], a game of cooperation that is widely studied in laboratory settings. Each game comprised 10 rounds, where in each round each participant i was allocated an endowment of $e = 10$ points, and was required to contribute $0 \leq c_i \leq e$ points to a common pool. In standard public goods games, participants' contributions are shared equally among members of the same group. SW, however, studied a variant in which participants were

[3] `http://www.mturk.com`

arranged in a network, so they shared their contributions with their neighbors. To reflect this change, players' payoffs were given by the payoff function $\pi_i = e_i - c_i + \frac{a}{k+1} \sum_{j \in \Gamma(i)} c_j$, where in place of the summation over the entire group of n players, payoffs are instead summed over $\Gamma(i)$, the network neighborhood of i (defined to include i itself), and k is the vertex degree (all nodes in all networks have the same degree). Therefore, i's contributions were, in effect, divided equally among the edges of the graph that are incident on i, where payoffs are correspondingly summed over i's edges. From this payoff function it is easy to show that when $1 < a < n$, players face a social dilemma in that all players contributing the maximum amount maximizes social welfare, but individually players are best off if they contribute nothing, thereby free-riding on the contributions of others.

SW chose networks that spanned a wide range of possible structures between a collection of four disconnected cliques at one extreme, and a regular random graph at the other, where all networks comprised $n = 24$ players, each with constant vertex degree $k = 5$. SW conducted a total of 73 networked experiments on AMT over a period of 6 months, including the following treatments, which are analyzed in this work[4] :

1. *All Human*, 23 *games*. All 24 players were human subjects.

2. *Altruistic Dummies*, 13 *games*. Four positions were played by computer, which contributed the full endowment each round. The dummies were arranged so that each human player was adjacent to precisely one dummy (*i.e.*, the dummies constituted a covering set for the graph).

[4] Section 6.5.3 makes use of an additional set of 15 related experiments conducted after the publication of SW. Because they were not described in SW, however, they are not included in the main results.

3. *Free Riding Dummies*, 17 *games*. Same as for altruistic dummies, but the dummies contributed zero in each round.

4. *Neighboring Altruistic Dummies*, 20 *games*. Same as for altruistic dummies, but the four dummies were arranged in two pairs, such that some human players were adjacent to two dummies, while others were adjacent to zero.

Surprisingly, SW found that network topology had no significant effect on contributions in any of the experimental treatments. From the Altruistic and Free Riding Dummy conditions, they established that players were behaving as conditional cooperators (in the generalized TFT sense), hence contributions in neighborhoods with high local clustering were more correlated than those with low clustering; however, the symmetrical nature of conditional cooperation effectively led positive and negative effects to cancel out. Moreover, from the concentrated dummies (Neighboring Altruistic) condition, they also established the absence of multi-step contagion of positive effects, although they did not rule out negative contagion.

6.4 Deterministic Models

This section defines and then evaluates a collection of models that can be considered *deterministic*, meaning that the output of a model is the expected contribution for the next round. All the deterministic models presented here suffer from a major shortcoming in predicting the full distribution of contributions, because they cannot in fact be expressed as distributions. Nevertheless,

we begin with them for three reasons: first, they are relatively simple and intuitive; second, they perform reasonably well at predicting average contributions; and third, they are frequently invoked both in agent-based models of cooperation [Axelrod, 1984] and also in previous attempts to replicate empirical data [Deadman, 1999; Castillo and Saysel, 2005; Janssen and Ahn, 2006]. Furthermore, the components of these models represent the base strategies from the meta-reasoning framework. Specifically, the actions of repeat and imitate feature prominently, although they function as a weighted combination that outputs a new action on the spectrum between full cooperation and full free-riding. Certain games have a particular structure wherein the payoffs can be considered as a continuous function over the action space. As a result, one can view strategies in this space as a simple linear function mapping actions from one round to the next. This property may not hold for games in general, but, in this class of games, it allows us to examine heuristics of the subsequent form.

While the predicted behaviors resulting from the following behavioral rules do not take payoffs directly into account, they do react to the observed behavior of others in a way that mimics intelligent thinking. For instance, the action in the next round may be influenced by others through a reciprocal relationship, or it may change towards selfish behavior, or some mix of the two. The best response option is another core component in this framework, and some of the following models also include a temporal aspect that allows for some measure of forward thinking. In later sections, the repeat/respond dynamic will serve as an anchor for the probabilistic strategy that appears to match the population better.

6.4.1 Model Definitions

Here are several representative deterministic models explored in the following analysis.

Linear Self-factor Model. Perhaps the simplest model one might imagine captures the commonly observed empirical regularity that players who contribute a lot (respectively, a little) in the previous round are more likely to contribute a lot (respectively, a little) in the current round [Wang et al., 2012]. Formally, the model predicts $c_{i,t}$, player i's contribution on round t, to be a linear function of player i's contribution in the previous round $c_{i,t-1}$:

$$\hat{c}_{i,t} = \beta_1 c_{i,t-1}$$

In essence, this β_1 coefficient is the repetition factor that determines how stable a player is. If $\beta_1 = 1$, then the same action would be repeated ad infinitum.

Linear Neighbor-factor Model. A second simple model is motivated by the notion of conditional cooperation [Fischbacher et al., 2001]—the more player i's neighbors contribute, the more player i is likely to contribute. Specifically, $c_{i,t}$ is predicted by the weighted average of player i's neighbors' contribution in the previous round, $\bar{c}_{i,t-1}$:

$$\hat{c}_{i,t} = \beta_2 \bar{c}_{i,t-1}$$

The β_2 incorporates how dependent the behavior is on others' actions, reflecting the imitation base component. Thus, a high β_2 would cause a player to converge to the same contributions as her neighbors over time.

Linear Two-factor Model. Next, we combine these two single-parameter models in a two-factor model that predicts $c_{i,t}$, player i's contribution on round t, as a weighted linear combination of (a) player i's contribution in the previous

round $c_{i,t-1}$, and (b) the average contribution in round $t-1$ of the local neighbors of player i, $\bar{c}_{i,t-1}$:

$$\hat{c}_{i,t} = \beta_1 c_{i,t-1} + \beta_2 \bar{c}_{i,t-1}$$

The coefficients β_1 and β_2 therefore capture the relative importance of player's previous actions versus his neighbors' previous actions, where we note that models of this general form ("place some weight on my own intrinsic inclination to contribute and some weight on my neighbors' contributions") generate behavior that is consistent with conditionally-cooperative models such as Tit-for-Tat [Axelrod, 1984], and even more complicated strategic models such as that proposed by Kreps et al. 1982. By combining these factors into a linear equation, we retain the spirit of a base non-reasoning strategy, while also achieving some robustness in terms of predictions.

Triangle-shaped Model. Motivated by Fischbacher et al. 2001, who observed that some players contribute proportional to their neighbors up to about 50% of the total endowment, after which their contributions decline in proportion to their opponents, let us examine the following "triangle" model:

$$\hat{c}_{i,t} = \beta_1 c_{i,t-1} + \beta_2 \bar{c}_{i,t-1} + \beta_3 (5 - |5 - \bar{c}_{i,t-1}|)$$

Threshold Model. Previous theoretical models [Glance and Huberman, 1993; Lopez-Pintado and Watts, 2008] have posited that players will contribute to a public good only when the average neighborhood contribution is above a certain threshold. We capture the essence of these "threshold models" using a logistic function, which maps a continuous variable onto the $[0,1]$ range and does so with a gradual probabilistic change between binary options. This function can represent rapid changes in behavior as a threshold and is written as:

$$\sigma(\bar{c}_{i,t-1}) = \frac{1}{1 + e^{-\lambda(\bar{c}_{i,t-1} - \theta)}}$$

Note this function has two parameters: θ, which is the midpoint where an average neighbor contribution of $\bar{c}_{i,t-1} = \theta$ leads to a probability equal to 0.5; and λ, which indicates how rapidly the function changes around the midpoint (*i.e.* as λ increases, the threshold approaches a step function). The resulting model is as follows:

$$\hat{c}_{i,t} = \beta_1 c_{i,t-1} + \beta_2 \sigma(\bar{c}_{i,t-1}) = \beta_1 c_{i,t-1} + \frac{\beta_2}{1 + e^{-\lambda(\bar{c}_{i,t-1} - \theta)}}$$

Let us note that the Tit-for-Tat strategy ("cooperate when others cooperate") translates roughly [5] to a threshold model with $\beta_1 \approx 0$, $\beta_2 \approx 10$, $\lambda \gg 1$, and $\theta = \theta^*$, where θ^* determines the position of threshold separating cooperation from defection. While the previous models are linear functions, the nonlinear threshold creates an all-or-none dynamic, pushing the estimated action towards the extremes. Still, the link between repetition/imitiation and next action is maintained, just with a different montonic mapping.

Time Discounted Models. The models above are all strictly reactive. Although backward-looking behavior is consistent with previous ABM models of cooperation [Macy and Willer, 2002], evidence from experiments on reward, punishment [Fischbacher et al., 2001] and partner updating [Wang et al., 2012] indicate that players are forward-looking, in the sense that they choose their current action in part in anticipation of how they expect other players to behave. In this context, forward-thinking involves an agent reasoning about how players built from one of the simpler models will react to its own behavior. Conveniently, the basic model reduces to itself when performing a step of optimization. A player who believes that the opponents are going to partially reciprocate will respond with some amount of cooperation, *based on how much*

[5] Technically Tit-for-Tat is defined for a two-player repeated prisoner's dilemma, so the translation to a multiplayer public goods game is necessarily imperfect.

the others are already contributing. This crucial point links a player's beliefs about others directly to the amount that could be lost if the wrong action is played. Additionally, the payoff at risk is proportional to the amount of time remaining in a repeated game; there is little sense in considering someone's reaction if the game will be over next round.

The relevant concept, as seen previously, is "future discounting:" the idea that people prefer payoffs today to larger payoffs tomorrow [Williams, 1938]. The relative size of this preference can be captured by a discounting parameter. It is here that recursive modeling can enter the picture of an n-player game. Assuming that the others implement a two-factor model, one of these weights, β_2, is the mechanism for participants to condition their play on others' actions.

If we take the view that i's contribution in the present round serves as an investment in keeping i's neighbors in a generous state, and setting $0 \leq \delta \leq 1$ as the discount rate, we can derive time-discounted versions of the two-factor linear and threshold models as follows:

$$\hat{c}_{i,t} = \beta_1 c_{i,t-1} + \beta_2 \sum_{\tau=t}^{T} \delta^{\tau-t} \bar{c}_{i,\tau}$$

where T is the total number of rounds in the game and δ is the discount rate. A player may have realized from prior play that his neighbors' contributions levels decline with time, but respond positively to high contributions. So, we can model the above equation as:

$$\hat{c}_{i,t} = \beta_1 c_{i,t-1} + \beta_2 \sum_{\tau=t}^{T} \delta^{\tau-t} \theta^{\tau-t} \bar{c}_{i,t-1}$$

where $0 \leq \theta \leq 1$. Setting $\gamma = \delta\theta$ and simplifying the geometric series gives:

$$\hat{c}_{i,t} \quad = \quad \beta_1 c_{i,t-1} + \beta_2 \bar{c}_{i,t-1} \left(\frac{1 - \gamma^{T-t+1}}{1 - \gamma} \right)$$

Thus, we obtain the following models.

$$\text{Discounted Two-Factor Model:} \qquad \hat{c}_{i,t} = \beta_1 c_{i,t-1} + \beta_2 \frac{1 - \gamma^{T-t+1}}{1 - \gamma} \bar{c}_{i,t-1}$$

$$\text{Discounted Threshold Model:} \qquad \hat{c}_{i,t} = \beta_1 c_{i,t-1} + \beta_2 \sigma \left(\frac{1 - \gamma^{T-t+1}}{1 - \gamma} \bar{c}_{i,t-1} \right)$$

where σ is the logistic function described in the Threshold Model subsection.

6.4.2 Predicting Average Contributions

Having defined a basket of models derived from the meta-reasoning framework, we can now proceed to evaluate them on their ability to predict the next action in the game for player i ($c_{i,t}$, the contribution at time t) given the current level of personal ($c_{i,t-1}$) and average neighbor contributions ($\bar{c}_{i,t-1}$). Consistent with previous work [Janssen and Ahn, 2006], there are two different types of evaluation based on predicting individual contributions: a *homogenous* population evaluation, which assumes that all players act the same way; and a *heterogeneous* population evaluation, in which each player is allowed to behave differently—sometimes very differently. Previous studies of public goods experiments [Fischbacher et al., 2001; Janssen and Ahn, 2006] have observed that behavioral data is better explained by allowing for heterogeneous types; however, homogenous strategies allow us to use more data to fit and evaluate each model, so let us consider both.

Homogenous Population Evaluation

As just noted, beginning with the assumption of a homogenous population, all players are described by the same set of model parameters. Each model is then fit using regression or parameter search where appropriate. For evaluation, the leave-one-out method is applied; that is, for a total of g games train on $g-1$ games, and test on the gth game, where every game is the test set once. The evaluation metric for each model's performance is root mean squared error (RMSE), a simple, intuitive measure of predictive accuracy.[6]

Table 6.1 shows the results of this evaluation. The two single factor models do the worst, where the self-factor model beats the neighbor-factor model, indicating that the contribution of player i, $c_{i,t-1}$, has more predictive power than the average contribution of player i's neighbors, $\bar{c}_{i,t-1}$. The linear 2-factor model, which uses both player $c_{i,t-1}$ and $\bar{c}_{i,t-1}$, has better predictive accuracy than either single factor model alone; thus there is predictive power in using both $c_{i,t-1}$ and $\bar{c}_{i,t-1}$. In general, the linear 2-factor, discounted 2-factor, triangle, and threshold models are comparable in performance. Because the simple linear 2-factor model has an error close to models with more parameters, it is a good tradeoff between parsimony and predictive accuracy.

Heterogenous Population Evaluation

Analogous to the homogeneous case, each model is trained on the majority of a player's games, keeping a hold-out set of 20% of the total or a single game,

[6] Note that using log-likelihood and max-likelihood to fit the models is a common technique in these domains. However, the risk of over-fitting individual behavior (see next section) was significant due to sparsity of available data in many cases. Regression appears to robustly fit the models.

Table 6.1: Homogenous model evaluation leaving one game out. The errors are the average RMSE for predicting individual contributions. Standard errors are all less than ± 0.02.

Model name	# of Params	All Human	Altruistic Dummies	Free-Riding Dummies	Neighbor Dummies	Mean
Self-factor	1	2.37	2.19	2.09	2.36	2.25
Neighbor-factor	1	3.16	3.40	2.72	3.37	3.16
Linear 2-Factor	2	2.27	2.14	2.02	2.31	2.18
Disc. 2-factor	3	2.26	2.12	2.02	2.30	2.18
Triangle-shaped	3	2.26	2.11	2.00	**2.27**	2.18
Threshold	4	2.25	2.12	**1.99**	2.29	2.16
Disc. Threshold	5	**2.23**	**2.07**	2.00	2.26	**2.14**

Table 6.2: Heterogenous model evaluation leaving one game out. RMSE results are shown for several behavioral models, based on learning a custom model for each player.

Model name	# of Params	All Human	Altruistic Dummies	Free-Riding Dummies	Neighbor Dummies	Mean
Self-factor	1	2.05	1.87	1.74	1.96	1.91
Group-factor	1	2.36	2.24	1.81	2.37	2.20
Linear 2-Factor	2	**1.97**	1.87	**1.57**	1.89	1.83
Disc. 2-factor	3	1.98	**1.80**	**1.57**	1.92	1.82
Triangle-shaped	3	2.11	1.93	1.67	**1.75**	1.87
Threshold	4	1.98	1.87	1.58	1.76	**1.80**
Disc. Threshold	5	2.02	1.86	1.59	1.87	1.83

whichever is larger[7] . We then evaluate the model on the hold-out set, repeating this procedure with a rotating hold-out set until all games are tested. We compute the RMSE on the test set and average those across all players weighted by their experience. The results of this analysis are shown in Table 6.2.

Although each model is now fit with much less data than in the homogeneous case, in general errors are reduced by learning individually customized

[7] Any player with fewer than three games is excluded on the basis that there is not enough training data for that individual.

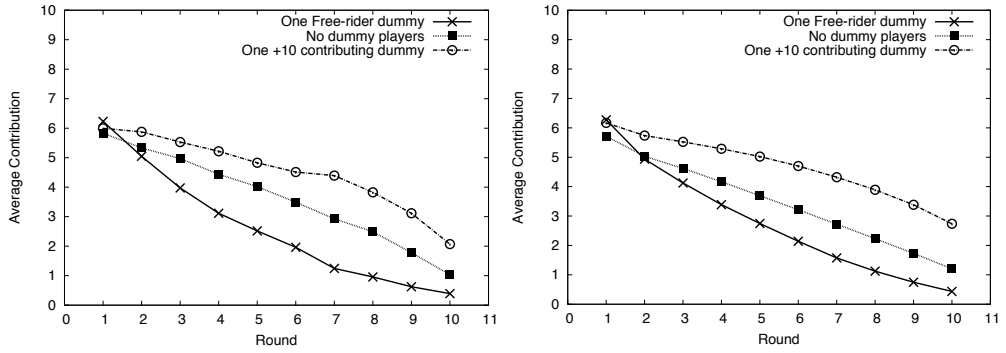


Figure 6.1: Average contributions per round for (a) experimental results from Watts and Suri 2011 and (b) simulated results using the time-discounted two-factor model.

models. We also see varying performance in the different treatments. For example, including a triangle strategy hurts performance when predicting trials with free riders, but helps when there are multiple high contributors present. Finally, Fig. 6.1 shows graphically, for the special case of the discounted two-factor model, how the predicted average contributions (right panel) compare with the empirically observed contributions from Suri and Watts 2011, for the three main treatments: all human, altruistic dummies, and free-riding dummies. Visually, the curves, which are generated via the method described in section 6.4.3, are hard to distinguish, indicating that the quantitative performance measures in Table 6.2 correspond to qualitatively meaningful agreement.

Analysis of Types

The superior performance of the heterogenous models in spite of their more limited data suggests that players use a variety of strategies that are not being captured by the homogeneity assumption. It can be instructive to ask what the population looks like, in terms of its distribution over these parameters. The

following results derive the first-round equivalent weights for players with $\gamma > 0$, because by necessity the relevant models assign lower weightings to the β terms to make up for the extra discounted term. As a result, the coefficients of some players will add to more than one, but we would expect them to decrease over the course of the game. Fig. 6.2 confirms the intuition that each player concocts his or her own strategy, showing that the distributions of the three parameters in the discounted two-factor linear model, β_1, β_2, and γ, all have broad support. Interestingly, Fig. 6.2 also shows that the distributions of β_1 and γ are effectively tri-modal, while the distribution of β_2 is close to uniform.

Motivated by this observation, the population is partitioned into "types" as follows: for β_1 allocate players to "low" $(\beta_1 < 0.25)$, "medium" $(0.25 \leq \beta_1 < 0.75)$, and "high" $(0.75 \leq \beta_1)$; for β_2, we have "low" $(\beta_2 < 0.5)$ and "high" $(0.5 \leq \beta_2)$; and γ, low, medium, and high as per β_1. As Table 6.3 shows, this partition corresponds to 18 cells, or "types", of which 9 have between 7% and 14% of the population. Together, these 9 types account for nearly 90% of the population[8] . In addition, note that 90% of the population lies in the medium ranges of γ and β_1, which constitutes only half of the parameter space, while there is a near even split between highly reciprocating players with high β_2 (46% are above 0.5) and those with low β_2 (54% are below 0.5). We might describe players with high γ as forward thinkers, and those with high β_2 as conditional cooperators.

As a side note, by individually customizing these models we see that certain models fit some individuals better than others, in terms of the best testing performance. The best model for an individual varies across the population,

[8] Interestingly, Janssen and Ahn 2006 presented a similar result using a different methodology, finding that a similar majority of players were accounted for by 8 out of 16 possible types.

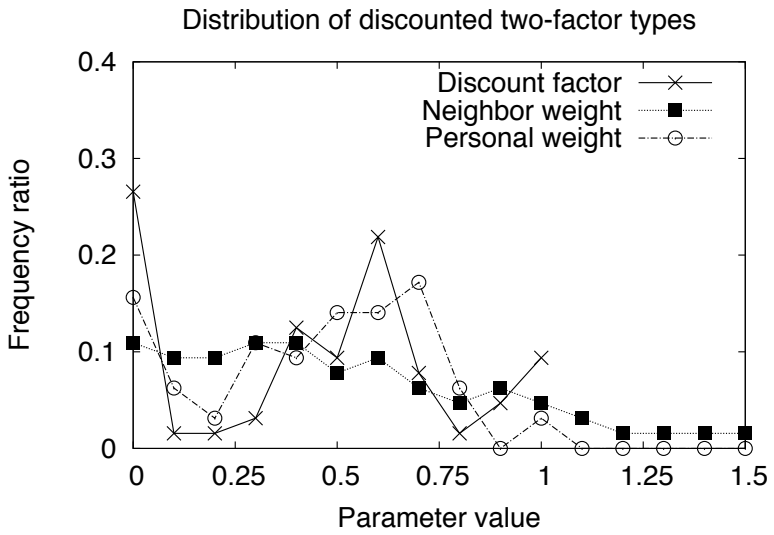


Figure 6.2: The distribution over types in the heterogeneous discounted two-factor model.

Table 6.3: Frequency of type by player, low, medium, or high discount, low, medium, or high personal weight, low or high neighbor weight. One could easily ascribe cognitive motivations to these values.

Neighbor: β_1 (Reciprocation)	Low $\beta_2 < 0.5$			High $\beta_2 \geq 0.5$			
Personal: β_2	Low $\beta_1 < .25$	Medium $.25 \leq \beta_1 < .75$	High $.75 \leq \beta_1$	Low	Medium	High	Sum for γ
Discount γ							
Low ($\gamma < .25$)	0.07	0.07	0.02	0.07	0.10	0.00	0.33
Medium ($.25 \leq \gamma < .75$)	0.02	0.14	0.11	0.10	0.12	0.00	0.49
High ($\gamma \geq .75$)	0.00	0.09	0.02	0.02	0.05	0.00	0.18
Sum for β_1	0.09	0.30	0.15	0.19	0.27	0.00	
Sum for β_2		0.54			0.46		

suggesting that not only is there a distribution across parameter values, but also across the decision-making process itself. We can attempt to use the most general model when possible to allow for a single model to capture these variations, but still face a tradeoff between the number of parameters and small amounts of data.

6.4.3 Predicting Full Distribution of Contributions

The model evaluation of the previous section seems promising and is also consistent with previous attempts to validate models empirically, which have also focused on average contributions over time [Deadman, 1999; Castillo and Saysel, 2005; Janssen and Ahn, 2006]. In light of this history, however, it is important to realize that the average contribution is potentially an extremely poor proxy for the full distribution of contributions. The reason is that contributions in public goods games are strikingly bimodal, with extreme actions of zero and ten appearing as the two modes, and a minority playing the actions between one and nine [Ledyard, 1995; Suri and Watts, 2011]. Over time the number of players at the maximum contribution declines while those who contribute zero increases significantly, but the bimodality persists. Clearly it is possible to accurately predict the average of a bimodal distribution while completely misrepresenting the underlying distribution. Yet, also clearly, it would be desirable for any agent-based model to replicate the full distribution as well as the average.

Thus motivated, we can now evaluate the same models in terms of their ability to predict the full empirical distributions, training one instance of each model per player on half of the data in each treatment, and testing against the distribution of the other half. Specifically, we first construct a simulated population of agents in the following manner: if player i is in the test set and the training set, put the model for player i in the simulated population in proportion to its experience in the test set; and if player i is in the test set but not the training set, we select at random from the training set chosen weighted by that player's experience in the training set. For each simulated population, we can then run a simulated game by sampling 24 players from the population and

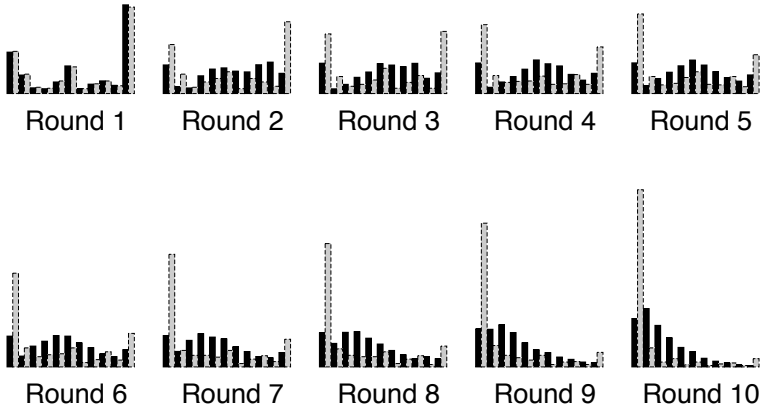


Figure 6.3: Actual population behavior compared to the deterministic discounted 2-factor model.

running their models using first round contributions chosen from the distribution of actual first round contributions in the test set. By repeating this process 20 times to get 20 simulated games, we arrive at roughly the number of actual games in each experimental treatment.

The result for the discounted 2-factor model is illustrated graphically in Figure 6.3, from which it is evident that the distribution of the model's predictions is clearly distinguishable from the bimodal distribution of the empirical data. That means that the average value is not the best indicator for what is going on at the micro-level, even though it is useful to fit model parameters. We can express this qualitative observation quantitatively through 100 repetitions this process by finding the Kullback–Leibler (KL) divergence, a standard measure for the extra information needed for a model to represent some original distribution, between the simulated and empirical distributions of rounds 2-10. For example, as shown in Table 6.4, the linear 2-factor model with low RMSE has a relatively high KL divergence value above 1, meaning that, on average, the log-odds ratio of the two distributions is off by a factor of 3 or greater. In general, Table 6.4 shows relatively poor performance for all the deterministic

Table 6.4: Evaluation of the distribution of the population's contribution for the deterministic models trained on half the experiments and tested on the other half. KL divergence measures (non-symmetrically) the difference between the true data and model output. Lower KL divergence represents higher accuracy.

Model name	# of Params	All Human	Altruistic Dummies	Free-Riding Dummies	Neighbor Dummies	Mean
Linear 2-factor	2	1.32	2.40	1.15	4.82	2.42
Disc. 2-factor	3	**0.84**	1.36	**0.93**	4.17	1.83
Threshold	4	0.96	**1.10**	1.87	**2.12**	**1.52**
Disc. Threshold	5	3.29	2.17	1.07	6.16	3.17

models. The reason is that in spite of their differences, all the deterministic models predict that high contributing agents will reduce their contributions steadily over time—a tendency that leads the initially bimodal distribution to become increasingly unimodal—whereas empirically, human agents tend to jump from very high to very low contributions almost discontinuously, spending very little time in the middle of the distribution and thereby preserving the bi-modality of the distribution even as the mean decreases. Predicting average contributions, in other words, is no guarantee of having captured the underlying behavioral dynamics.

6.5 Stochastic Models

Motivated by the observation that individual contributions are not well represented by the expectation, it makes sense to introduce a method for constructing stochastic models that builds on the successful aspects of the deterministic models, but more accurately captures the bi-modality of the empirical distribution. The general approach is that for each of the deterministic models in

the previous section, we can define a corresponding stochastic model that invokes the deterministic model as a subroutine. Rather than predicting an expected contribution, however, the stochastic model instead makes use of the deterministic model to predict that player i will make the same contribution they did in the last round with probability ϕ, and change to 0 with probability $1 - \phi$. This model essentially limits the choices to cooperate (contribute the same) and defect (contribute nothing). In addition, the stochastic model also predicts that a player will make a strategy uniformly distributed in the range $[1, 10]$ with probability ϵ, which is estimated directly from the data and reflects the empirical observation that some agents play the non-extremal actions or just do something completely unpredicted. To make the probabilities sum to 1, the probability of free-riding is adjusted to $1 - \phi - \epsilon$.

To illustrate our method for generating a stochastic model from a deterministic one, let us use as an example the linear 2-factor model described above. From Section 6.4.1 we see that the two factor model predicts the next round's contribution via

$$\hat{c}_{i,t} \quad = \quad \beta_1 c_{i,t-1} + \beta_2 \bar{c}_{i,t-1}$$

Conditioned on $c_{i,t-1} > 0$, we can rewrite this equation as

$$E[\hat{c}_{i,t} \mid c_{i,t-1} > 0] \quad = \quad c_{i,t-1} \left(\beta_1 + \beta_2 \frac{\bar{c}_{i,t-1}}{c_{i,t-1}} \right)$$

Notice that one can interpret this expectation as a value times the probability of a player contributing that value. Since contributions generally decrease, it is most often the case that $c_{i,t} \leq c_{i,t-1}$. In addition, contributions are always at least 0. Thus, we can interpret $\phi(c_{i,t-1}, \bar{c}_{i,t-1}) = \beta_1 + \beta_2 \frac{\bar{c}_{i,t-1}}{c_{i,t-1}}$ as a probability of playing $c_{i,t-1}$ again during round t. Players may, of course, choose not to contribute the same as they did last round. It is possible that

players increase their contributions or contribute some amount between 1 and 10. To capture these cases, we let ϵ be the probability of contributing a random amount $\Pr[c_{i,t} = \mathcal{U}[1,10]] = \epsilon$. Figure 6.3 shows that players often contribute 0. So we let the remaining probability, $1 - \epsilon - \phi(c_{i,t-1}, \bar{c}_{i,t-1})$ be the probability of contributing 0. Combining these quantities gives

$$E[\hat{c}_{i,t} | c_{i,t-1} > 0] = c_{i,t-1}(1-x)\phi(c_{i,t-1}, \bar{c}_{i,t-1}) + 5.5\epsilon,$$

where $x = \frac{5.5\epsilon}{\phi c_{i,t-1}}$ corrects the upward bias in the expectation caused by the uniform random variable in $[1,10]$. Observe that if we plug x into the above equation we get:

$$E[\hat{c}_{i,t} | c_{i,t-1} > 0] = c_{i,t-1}\phi(c_{i,t-1}, \bar{c}_{i,t-1}),$$

which shows that the stochastic model will output the same prediction, in expectation, as the deterministic model. However, we shall see that the actual distribution of predictions is much closer to the experimental data. The above equation describes the model for when $c_{i,t-1} > 0$. When $c_{i,t-1} = 0$, players most often play 0 for the rest of the game, but occasionally they do increase their contributions. To capture this effect we say that a player might contribute an amount uniformly distributed in $[1,10]$ with probability ϵ_0, giving

$$E[\hat{c}_{i,t} \mid c_{i,t-1} = 0] = 5.5\epsilon_0$$

In this case we can fit ϵ_0 to the data so that we can ensure that the expected prediction of the stochastic model is the same as the prediction of the probabilistic model.

Recall that ϕ was defined in terms of the linear 2-factor model. The other

parameters, ϵ and ϵ_0, were fit to the data. Thus, this stochastic model is determined by

$$\Pr[c_{i,t} = c_{i,t-1} \mid c_{i,t-1} > 0] = \phi(c_{i,t-1}, \bar{c}_{i,t-1}) = \beta_1 + \beta_2 \frac{\bar{c}_{i,t-1}}{c_{i,t-1}}$$

The general technique described here can similarly be applied to each of the models defined in Section 6.4.1.

6.5.1 Baseline Stochastic Models

Although this recipe for generating a stochastic version of each of the previously defined deterministic models yields a corresponding collection of stochastic models, it is clearly not the only way of generating a plausible stochastic model. To check that the deterministic component of our stochastic models is contributing to their performance in a meaningful way, therefore, let us add two unrelated baseline models that are also stochastic in nature but derive their probabilities in different ways.

Simple Stochastic Model

The first baseline model is extremely simple. Again, let ϕ be the probability of a player contributing the same in round t as in $t-1$. But, this model estimates ϕ directly from the training data and does not use a deterministic model to do so. Let ϵ be the probability of contributing some amount uniformly distributed in the range $[1, 10]$. Again, ϵ is estimated from the training data. Finally, let $1 - \epsilon - \phi$ be the probability of contributing 0. Thus, this model is given by

$$E[c_{i,t}] = c_{i,t-1}\phi + 5.5\epsilon$$

Since this model estimates ϕ directly from the data, comparing the stochastic models that estimate ϕ using a deterministic algorithm to it will allow us to

understand how much predictive accuracy a trained deterministic model adds.

Experience-Weighted Attraction

A different type of stochastic model, a version of which has been used previously to model agent behavior in public goods games [Janssen and Ahn, 2006], is motivated by the notion of Experience-Weighted Attraction (EWA), proposed by Camerer and Ho [1999], as a way to represent gradual learning in response to payoffs. The EWA model keeps track of two variables for every player: the number of observations N_t, and A_{jt}, attraction of action j after period t. These attraction values represent the current tendency of playing the corresponding actions, and can therefore be converted directly into a probabilistic strategy.

Updating has two steps. In step one, the experience is updated as $N_t = \rho N_{t-1} + 1$, where ρ is an experience decay parameter. In step two, the attractions are changed:

$$A_{j,t} = \frac{1}{N_t}(\phi N_{t-1} A_{j,t-1} + [\delta + (1-\delta) I(s_i, s_j)] U_i(c_{i,t}, \bar{c}_t))$$

where U is the utility function over the actions of the players in the neighborhood and I indicates whether the strategy was used at time t. The values ϕ and δ are parameters of the model respectively representing the decay of previous attractions and a calibration of actual versus imagined effects.

To convert the attraction values to a strategy, a logit function is typically used, which has its basis in the quantal response function and uses a temperature parameter λ:

$$P_{j,t+1} = \frac{e^{\lambda A_{j,t}}}{\sum_{k=1}^{M} e^{\lambda A_{k,t}}}.$$

Along with the experience decay ρ, this model contains four parameters

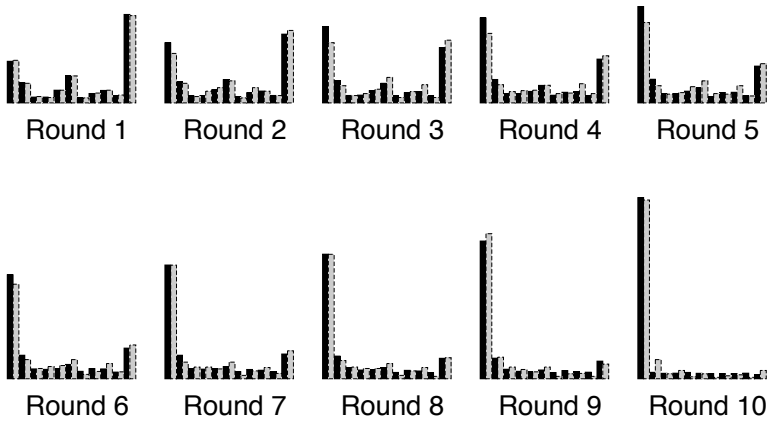


Figure 6.4: Average action frequencies of the actual population (gray bars) versus the heterogeneous population, 2-factor discounted stochastic model (black bars).

that must be set by an exhaustive brute-force search. Extra parameters are sometimes added to represent temporal decay or modify the utility function that might be shifted towards considering other players' utilities. Unfortunately, the entire parameter space must be searched simultaneously because of the non-linear ways that each parameter interacts with and depends on the others. As a result, fitting this model takes time exponential in the number of parameters. An alternative is to use the self-tuning EWA model, which relies on a single parameter and the other parameters are adjusted on the fly. However, this option does not necessarily solve the problem, as different sequences or conditions can result in divergent outcomes for test data.

6.5.2 Predicting the Full Distribution of Contributions

The ability of the stochastic models to predict the distribution of the population's contributions can be tested using the same method described in Section 6.4.3; that is, the same models are trained on the no-dummy treatment and compared to the human behavior data across each treatment to test for

Table 6.5: Evaluation of the simulated stochastic models' output distribution of contributions where individual models are trained on half the experiments and tested on the other half. KL divergence measures (non-symmetrically) the difference between the true data and model output. Lower KL divergence represents higher accuracy.

Model name	# of Params	All Human	Altruistic Dummies	Free-Riding Dummies	Neighbor Dummies	Mean
Simple Stochastic	3	0.44	**0.47**	0.61	0.83	0.59
Stochastic 2-factor	4	0.34	0.68	0.53	0.81	0.59
Stochastic Disc. 2-factor	5	**0.20**	0.53	0.47	0.72	**0.48**
Stochastic Threshold	7	**0.20**	0.65	**0.43**	**0.71**	0.50
Stochastic Disc. Threshold	8	0.24	0.63	0.64	1.11	0.66
EWA	5	0.70	1.22	1.21	1.34	1.12

transferability across experiments. Figure 6.4 shows the results for the stochastic version of the discounted two-factor model over 20 independently generated populations, each playing one game with 24 players. Visually, the match is much better than for the deterministic case, an impression that is confirmed quantitatively in Table 6.5, which shows the KL divergence between the true population behavior in the all-human experiments and the actions output by the simulated model. Clearly, the performance of the stochastic models is strikingly better than their deterministic counterparts. Moreover, the stochastic models using the deterministic subroutines outperform both the simple stochastic baseline model and also the EWA model, which performs relatively poorly. These results, in other words, justify the approach to constructing meta-reasoning stochastic models: clearly the information contained in the deterministic predictions is useful. However, converting them to stochastic generative processes dramatically improves their ability to replicate the full distribution while slightly decreasing RMSE performance.

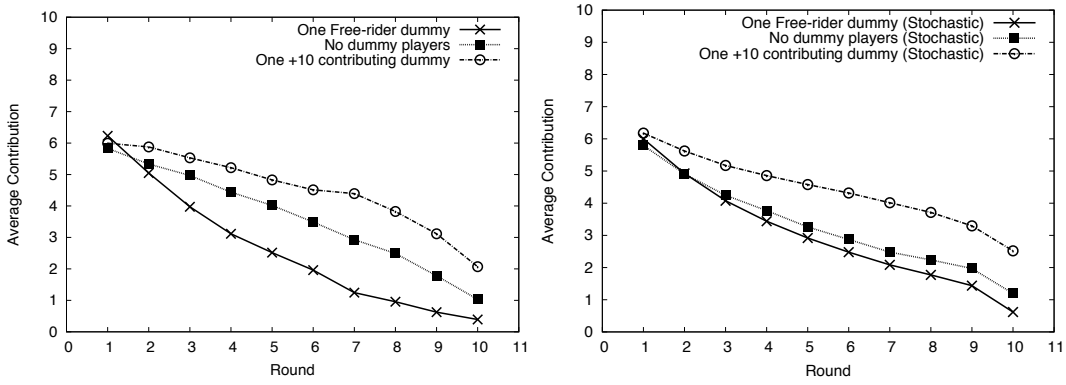


Figure 6.5: Average contributions per round for the stochastic discounted two-factor model (right) compared with empirical data (left).

6.5.3 Selecting a Model

Table 6.5 shows that the stochastic discounted two-factor model exhibits the best overall performance with respect to the KL divergence. In addition, Figure 6.5 shows that this simulated model generates average aggregate contributions over time that are again visually similar to those from Suri and Watts 2011 and comparable to those generated by the deterministic version of the same model.[9] Finally, Table 6.6 shows the transfer learning performance of each model; i.e. where each model is trained on the all-human data and then evaluated on a distinct experimental treatment. To maintain a fair comparison between all treatments, the test set for the all-human treatment that is used here is a second set of all-human experiments conducted by Suri and Watts several months after the experiments reported by SW 2011. This set of experiments

[9] Because the stochastic model makes predictions about the probability of a move, not the actual contribution, it is unclear how to evaluate its performance using the RMSE tests from the previous section. On the one hand, evaluating the expected contribution yields performance very close to the deterministic models, where the only effective difference lies in the additional noise term. On the other hand, first generating the full distribution of simulated moves and then scoring each move results in much higher RMSE. This is because the stochastic models predict extreme values and RSME penalizes heavily when one of these predictions is wrong. Since we are interested primarily in replicating the distribution of moves, and because the average of this distribution is also close the empirical average, the RMSE tests are omitted.

Table 6.6: Evaluation of the simulated stochastic models' ability to transfer experience across different experimental setups. Actual behavioral data is compared using KL divergence to simulated output distribution of the population's contribution where individual models trained on the all human treatment and tested on the other treatments, including previously left-out all human experiments. Lower KL divergence represents higher accuracy.

Model name	# of Params	All Human	Altruistic Dummies	Free-Riding Dummies	Neighbor Dummies	Mean
Simple Stochastic	3	0.19	0.21	0.32	0.22	0.24
Stochastic 2-factor	4	0.13	0.24	0.19	0.17	0.18
Stochastic Disc. 2-factor	5	**0.08**	**0.16**	0.28	**0.15**	**0.17**
Stochastic Threshold	7	0.13	0.32	**0.17**	0.18	0.20
Stochastic Disc. Threshold	8	0.14	0.32	0.38	0.19	0.26
EWA	5	0.44	0.40	1.09	1.13	0.77

differed from the original all-human experiments in two respects: first, given the lapse in time relative to the churn rate of workers on AMT, the subject pool was largely distinct; and second, subjects were informed not only of the contributions and payoffs of their immediate network neighbors (the original treatment), but also those of their neighbors' neighbors, along with the connections between them. For both reasons, this set of all-human experiments can be considered to be a true out-of-sample test set, hence the all-human results in Table 6.6 can be compared naturally with those of other treatments. Based on both within treatment (Table 6.5) and between treatment (Table 6.6) performance, therefore, we should select the stochastic discounted two-factor model as the preferred model for conducting the agent-based simulations, to which we turn next.

6.6 Simulating Empirical Agent-Based Models

Having selected the stochastic discounted two-factor model (SD2F) model as our candidate empirical agent-based model, we now return to our original motivation of deploying this model in the traditional manner of ABMs, namely as thought-experiments designed to generate new theoretical insights. Specifically, the first step is to fit a customized model for all players, which allows us to construct a model population from which agents are drawn to participate in a series of games. In this explorative setting, other parameters of the situation, such as the network size or structure, or the arrangement of player types to nodes in the network, can be varied systematically. In this way, we can explore a much broader range of the parameter space than would be possible with human subjects experiments.

6.6.1 Network Size

Recall that the main result of SW was their surprising finding that network topology had no significant impact on contributions. Because, however, the networks in question were relatively small ($N = 24$) it is possible that the lack of effect was due simply to insufficient variation in the path lengths, which for the connected networks varied only between 2 and 2.5. If true, then running the same experiments on much larger networks would allow for greater variation in the underlying structural features, and hence greater impact of structure on contribution. Testing this hypothesis requires us to simulate the model populations on networks of increasing size, ranging from $N = 24$ to $N = 2400$. Interestingly, Figure 6.6A shows no dependency on size for the three fully connected topologies studied by SW: the connected cliques, the

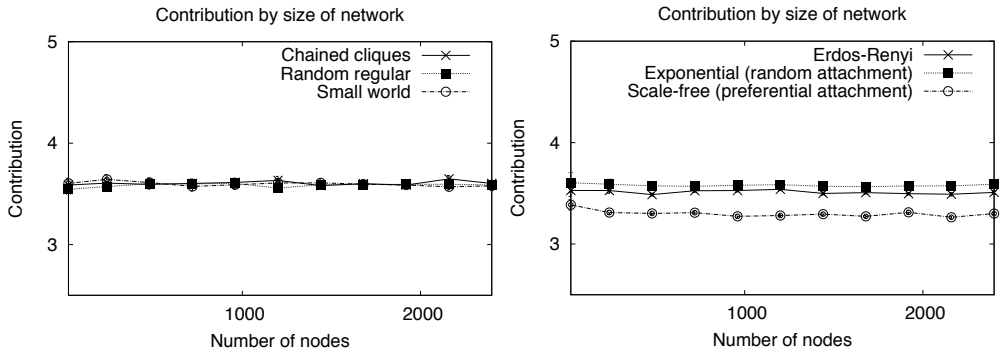


Figure 6.6: Average game contributions vs. N for (a) the connected clique, small-world and random regular topologies studied by Suri and Watts [2011], and (b) Erdös-Renyi, exponential, and scale-free random graphs.

small-world network, and the random regular graphs. Figure 6.6B shows similar findings for three other natural topologies—an Erdös-Renyi random graph, a random graph with an exponential degree distribution, and a scale-free random graph[10] —suggesting that the conclusion of SW is robust with respect to network size but not across other graph distributions. We should expect this result because the decisions faced by individual agents is insensitive to the total number of players in the network, but can depend heavily on the number of neighbors.

6.6.2 Network Density and Variance of Degree

Another possible explanation for the absence of dependence on network structure in the SW experiments is that all players had equally sized neighborhoods, thus overlooking two additional sources of variation in network structure: the

[10] The exponential and scale-free random graphs were constructed using the configuration method [Newman, 2003].

average degree d of the network; and the variance $var(d)$ (across players). Testing these dependencies, Figure 6.7 shows that although varying d has no impact (Figure 6.7A), increasing the variance of degree leads to lower contributions (Figure 6.7B, solid squares), consistent with the scale-free results from Figure 6.6B. On reflection, these results make sense. As explained in Section 3, the network version of the public goods game effectively splits a player's contribution equally among its edges, hence all else equal nodes with many partners contribute less per partner than nodes with few partners. As long as all players have the same number of partners, the dependency on degree is symmetrical, hence the average density has no effect in the case of zero variance. Increasing the variance, however, breaks this symmetry, creating winners (high degree nodes) and losers (low degree nodes), where the latter are thereby more inclined to lower their contributions. Following this reasoning, we should expect networks with high variance to yield somewhat lower average contributions, as indeed the simulations suggest.

For similar reasons, we might also expect that contributions should depend on "degree assortativity" α, the tendency of high (low) degree nodes to be adjacent with other high (low) degree nodes [Newman, 2003]. Indeed, Figure 6.7B shows that as α changes from negative (crosses) to positive (open circles), the dependency on variance decreases. Most of this effect, however, is due to the positive assortativity: that is, when high-degree players are more likely to be neighbors with each other (likewise for low-degree nodes), contributions increase, mitigating the effects of degree variance.

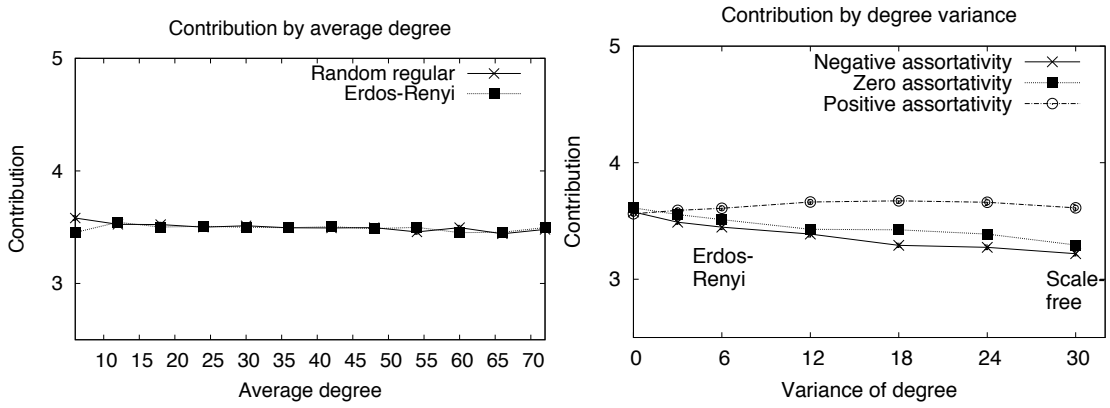


Figure 6.7: Average game contributions for random graphs of size $N = 240$ vs (a) average degree k and (b) variance $var(k)$ of the degree distribution.

6.6.3 Correlations between Player Type and Node Degree

The dependency of contributions both on degree variance and also assortativity raises an additional possible source of dependency—namely that assigning more (or less) generous players to nodes with higher (or lower) degrees might mediate or alternatively exacerbate the effects of breaking the degree symmetry. To check this hypothesis, define a new parameter $\rho = corr(\frac{\beta_1+\beta_2}{1-\gamma}, d)$, which quantifies the correlation between the overall generosity of agents in the SD2F model (as measured by their respective parameters) and the degree of a node in the network. As ρ is varied, that is, high degree nodes become either more $(\rho > 0)$ or less $(\rho < 0)$ likely to be generous. Figure 6.8 shows the same results as Figure 6.7B except where ρ is now strongly negative (left panel) and strongly positive (right panel), respectively. Interestingly, in networks with negative or no assortativity (α), a negative ρ lowers contributions further, while a positive ρ can by and large reverse the effects of negative assortativity. Positive assortativity, moreover, appears to compensate for increasing variance regardless of ρ. Overall, we conclude that both positive ρ and positive α can reverse the negative contributory effects of an unequal network, while negative values cause

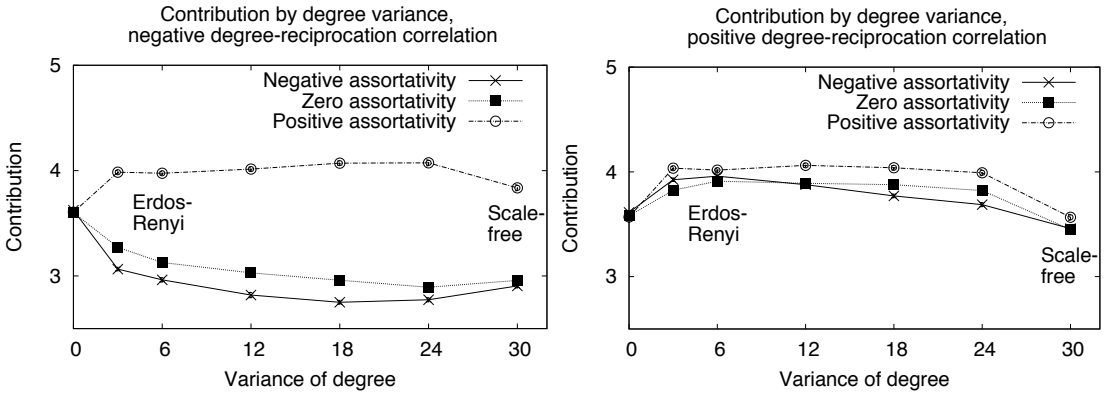


Figure 6.8: Average game contributions vs. degree variance where player generosity and node degree are (a) negatively correlated, and (b) positively correlated.

low contributions in an unequal network to drop still further.

6.7 Conclusions and Future Work

The experiments of this chapter demonstrate the overall success of an approach of basing stochastic agent models on the fundamental components of repetition, imitation, and randomness. Because imitation is ill-defined in a context with many agents, it makes sense to use the average action as an attraction point that can slow or speed the tendency towards a best response. The added resilience with the decaying time parameter is one way to incorporate extra reasoning or forward-thinking when it becomes computationally infeasible to recursively model everyone at once. In sum, this type of meta-reasoning model has proven its utility in these complex settings where people make decisions in a myriad of ways.

A possible limitation of this approach, however, relates this emphasis

on empirically accurate models of agent behavior over the traditional emphasis among ABM researchers on cognitive plausibility. Aside from interpretability, cognitively plausible models would seem to have the advantage of generalizability—that is, one might expect the same model to work not only in the exact conditions tested in a given experiment, but across a broad range of conditions. By contrast, a cognitively implausible or otherwise uninterpretable model seems less likely to apply to novel conditions, even if it performs well on the training data. For example, the finding in the previous section that contribution levels do not change with network density seems highly dependent on the assumption—implicit in the behavioral model—that the marginal per-capita return (MPCR) defined in the payoff function does not depend on degree. How would player behavior change if that assumption were violated? Because we have no model of how the agent is thinking about the game, or evaluating its utility, we cannot say. All the same, it is perhaps the lack of data that prevents us from finding this type of dependency, and presumably the meta-reasoning model could be adapted accordingly. In reality, even models with a deeper basis in utility theory, such as experience-weighted attraction, suffer from a lack of an easy interpretation of how the parameters work together, notwithstanding loose cognitive connections. One big reason for this weakness in previous models is that the change in one parameter can affect how the others are expressed in the model, in somewhat unpredictable ways.

Although the issue of generalizability is an important one, note that even cognitively plausible models can fail in exactly the same way. Most obviously, it can happen when the circumstances are varied in a way not imagined by the modeler, but as noted in Section 1, models can fail even under precisely

the conditions imagined simply because humans agents violate the model assumptions in subtle but consequential ways. Furthermore, it is possible that players will act differently in a game when the rules or structure of a game is changed, leading to the possibility that a model trained in one scenario will fail to adequately predict in another. This phenomenon would hold true if there is some cost to reasoning that alters the tradeoff between high utility and the problem-solving required to achieve it, such that a player could do more or less reasoning or learn in a different way. Currently, models do not have the capacity to account for this type of dependence on the difficulty of reasoning, and thus all of them would fail to transfer between games that are different enough. Thus, while interpretability seems a desirable feature for ABMs, all else equal, empirical calibration has advantages when it comes to robustness and blindness to modeler bias. In any case, the challenge of generalizability can be reframed as one of conducting the appropriate range of experiments.

This last point therefore motivates a need a for tighter integration between agent-based modeling and behavioral experiments (as well as data collection more broadly). In the current work, that is, data from behavioral experiments was used to identify an empirically accurate ABM. This empirical agent-based model was applied to an exploration of the behavior of hypothetical human agents across a much broader parameter space than was possible in the experiments. A natural next step is to view these results as new hypotheses—about the effect of assortativity, for example, or lack of effect of density—to be tested in future experiments. These experiments, in turn, would no doubt lead to more accurate and generalizable EABMs, which could then be used to perform still more general simulations, followed again by more hypotheses and more experiments. In short, the demonstrated results advocate that future

work should attempt to close the "hypothesis-testing loop" of the Multiagent Cycle, thereby allowing behavioral experiments and EABMs to complement and reinforce one another over time.

Chapter 7
Conclusion

This dissertation has explored the use of an agent and population modeling framework that justifies the hypothesis that simple heuristics plus some learning, reasoning, and planning are the basis of decision-making in repeated multiagent interactions. The rest of this chapter summarizes my contributions to the field of empirical multiagent studies and the framework's application to selected strategic domains.

To review, Chapter 2 introduced the formal meta-reasoning model used in the following chapters. Chapters 3 and 4 explored how the model works with simple learning algorithms in repeated games. In Chapter 5, the focus was on reasoning and planning against an anonymous population in the Lemonade Stand Game, where an accurate opponent model is more important than within-match learning. The final experiments in Chapter 6 applied the framework to humans in public goods games, which have been a popular setting that economists have used to make discoveries about non-equilibrium behavior.

7.1 A Meta-reasoning Framework for Repeated Games

The meta-reasoning model presented above is built upon certain foundational assumptions. First, players who are making strategic decisions apply some

amount of reasoning in the form of cognitive effort to succeed in a game. Because thinking carries a computational cost, perfectly optimal play becomes harder to carry out as the difficulty of the task increases. In addition, the widespread use of bounded rationality is *public knowledge*, which further decreases the likelihood that players will enter into, much less begin with, an equilibrium state in complex social situations. The form that reasoning will take in practice will depend on the circumstance. In p-beauty games, a reasoning step takes the form of a multiplication operation. Other games require a more advanced optimization procedure. When learning is a feasible option, then teaching could be the proper response.

The next assumption is that non-reasoning behavior takes the form of a relatively small set of base strategies, which depend on the details of the game. These strategies are based on fact that the temporal aspect of sequential game decisions focuses behavior on actions that have been previously and recently played, either by the individual in question or the other participants in the game. These psychological tendencies, defined as repetition (ϕ) and imitation (μ), provide two main pillars for the set of base strategies. Possible explanations for the widespread observation of repetitive strategies include comfort, reinforcement, or simple inertia. Imitation is also viable when people expect that others have superior experience or knowledge. A third supporting heuristic is simple randomness (ϵ), which has been the anchor of single-round iterated reasoning models since their inception and covers the rest of the base-level strategy space, thereby providing an outlet for noise. These base components are the foundation for further reasoning in a repeated scenario, and a non-reasoning agent can be represented as a stochastic mixture of them: $[\epsilon, \phi, \mu]$, corresponding to the probabilities of randomness, repetition, and imitation.

In addition, the iterated best response to maximize future discounted reward yields additional features, with their corresponding parameters $\omega_{k,\gamma}$.

For complex or asymmetric games, the mapping of these action primitives to specific actions can itself be an challenging question. For example, how to imitate another player is not always clear or possible, although the answer is self-evident in other cases. Imitation could simply mean a complementary action, as when coordination is possible only if players choose opposite actions. Routing games are an instance with this property. The point is that these abstracted features can provide more information about future actions than the original inputs. I have added in the extra insights about how to define and extract these primitives in an automated way.

The final assumption is that players engage in some amount of forward-thinking, ranging from zero to near-infinite. The choice of horizon for planning the consequences of the next action can have a large impact on the decisions of agents. Indeed, the conflict between short-run and long-run effects of choices has been studied in psychology and related disciplines for a long time and remains a significant challenge in understanding and shaping group decisions in a globalized world. One way of representing the time horizon of an agent is the discount rate, which weights the values in future time steps according to a decreasing exponential function. This parameter is not the only way to account for short-term thinking (a hyperbolic function provides an even steeper drop off), but it has some mathematical convenience.

Whether these assumptions are more well-founded than the traditional game theory assumptions of perfect reasoning and common knowledge is a valid question. This dissertation takes the view that there are empirical reasons for using a framework that is more resilient to a variety of outcomes. In a

world of big data, it would seem that modelers should prefer this approach to the alternative.

I have addressed the algorithmic challenges of training predictive models built from a meta-reasoning framework in part by carefully defining the actions output by each strategic component and partially by constraining previous models. By representing each agent's strategy with a small number of parameters, a modeler using this technique can efficiently compute the set of strategies and find the optimal parameter setting. This method reduces a dynamic modeling problem into a machine-learning problem with inputs (the underlying strategy types) and targets (the next action in the sequence). This dissertation has used regression and gradient methods to fit the models, but future analysis is not limited to these tools.

7.2 Experiments in Repeated Games

The games and sources of data explored above have the property that different types of behavior emerge based on the decision-making process employed by the participating agents. Models built for the purpose of transferable prediction will be most robust when they can capture the relative frequencies of these types. What unites these three experimental settings is the perception of agents that the joint action serves as a state of the world, which then affects the resulting decisions. One common theme is the conflict between cooperation and private interest (or at a higher level, between conditionality and non-responsiveness), which prisoner's dilemma and public goods have long been used as basic tools to investigate. The Lemonade Stand Game shares the cooperative element, although the decision about the characteristics of an ideal partner requires a little more thought. The tension between long-term

and short-term results is another key element here. In long interactions, it can often benefit a player to make short-term sacrifices for the sake of ultimate reward. However, thinking about future possibilities is also computationally intensive, and players must once again try to identify agents with this capacity and respond to them differently than a myopic opponent. These games are interesting to people because there is more than one way to play, and the choice of strategy is influenced by repeated experience with other social agents, who bring a unique perspective to the interaction.

7.2.1 Learning Algorithms in Simple Games

In the single-round version of social dilemma games, the basic conflict is between cooperation and defection. In the case of repeated games, the more important distinction is between reciprocation and unconditionality (the observation that a player will not encourage or respond to good behavior). A modeler that can distinguish between players of these two types will have a sure advantage when it comes to constructing a strategy and can exhibit superior performance as a result.

Chapter 3 described how a variety of learning algorithms behave in 2-player, 2-action games, with a special focus on memoryless ϵ-greedy Q-learning. This algorithm has the unique property that it can cooperate conditioned on the other player's conditionality, without knowing the state of the previous round. This algorithm presents a challenge for a modeler because, in the empirical setting, it does not converge to any particular behavior, but instead adapts in response to the relative cooperation it sees.

Chapter 4 used the framework described above to build models of historical data of learners and teachers. For a pure teaching strategy, the base

mappings are sufficient in these cases. A learner attempting to maximize the discounted sum of rewards will incorporate a mix of the discounted best responses, along with the base strategies. Furthermore, a learning agent will increase its probability of certain discounted responses as its reward changes. This observation led to the development of the δ-model, a dynamic model that shifts the probabilities as a linear function of the deviation of reward from a midpoint. Using regression over data gathered from a series of trial runs, the parameters of the δ-model can then be used to craft an optimal teaching response that would not be available without these observations. The modeler outperforms the other algorithms provided history against a set of strategies covering the space.

The issue of adaptive algorithms arises whenever players have the opportunity to respond to their environment over long temporal periods. Chapter 4 lays out one way to deal with learning agents, by demonstrating a link between accumulated rewards and the change in behavior. The lessons from these simple games show how some of this behavior can be captured by a general modeling framework, which can then be applied to more complex domains.

7.2.2 Lemonade Stand Game

Chapter 5 investigated the behavior of a population of software agents in the Lemonade Stand Game (LSG), a special case of the class of games known as location games, or canonically as Hotelling Games. These types of games are typical of the decisions faced by retail establishments when deciding upon locations for stores, given that others are doing the same. More generally, the strategic choices in this game have a similar structure to any where players

need to find a balance of meeting demand for some item in the face of competing supply. As such, they are a microcosm of the challenges faced in the realm of auctions and markets.

The significant effect of others' behavior, along with the confluence of competition and partnership that emerge in the LSG, means that modeling—understanding the behavior of competitors—takes on paramount importance. This dissertation shows how to successfully apply a meta-reasoning framework to this class of games, and how models trained on historical data can be transferred across different payoff regimes. In particular, the $[\epsilon, \phi, \mu]$ basis is a helpful place to start the iterated reasoning process, where the degree of repetition as well as imitation (in this context, the mirroring or across action) provide the groundwork for how likely players are to remain in their location or move to a new one. The time horizon or discount factor is also a useful way to capture how these probabilities vary from one state to another.

In the 2011 and particularly the 2012 competition, the ability to transfer experience across the large payoff space resulted in a convincing victory and validation for this modeling approach. The large gap in performance between the Rutgers agent and the next-best player demonstrates clearly that even anonymous population models have tremendous potential for predicting the likely course of play in these rich and noisy domains. The strength of the model shines through as a technique for forecasting the behavior of the population for a previously unobserved payoff function.

7.2.3 Public Goods Behavioral Experiments

Much of the focus on models in economics is the link between utility and response, and how this dynamic leads to some equilibrium over time.

Economists have proposed learning rules such as experience-weighted attraction to explain non-equilibrium behavior. This adaptive mechanism is a non-linear strategy function that contains parameters for update speed, noise, other-regarding preferences, etc. However, this dissertation takes the perspective that a repeated base strategy (again, the parameters $[\epsilon, \phi, \mu]$) combined with some extra forward-thinking is a more robust model for training with little data because the linear nature of this type of strategy function is easier to fit.

The experimental results in Chapter 6 demonstrate the strength of this model, especially when converted into a stochastic generative strategy. This type of strategic model, when customized to individual players, reproduces the distribution over the actions of the population well. Predicting a group's behavior from individual models is a different objective from training accurate models, but is sometimes the more important goal.

7.3 Implications for Future Research

This dissertation outlines a general framework for modeling agent behavior in strategic settings. In a world of proliferating data sources in social domains, there will be increasing need for efficient models that transfer to new situations. As explained throughout the document, a pure statistical analysis can fail without some connection to the reasoning process that self-directed agents execute when interacting with others. The consequence of this phenomenon is that model builders will be forced to rely on more sophisticated approaches whenever people are making social decisions. The conclusion from the case studies investigated above is that a simple yet general agent modeling framework is both feasible and valuable.

Bibliography

Robert Aumann and Michael Maschler. *Repeated Games with Incomplete Information*. MIT Press, 1995. 6, 28, 53

Robert Axelrod. *The Evolution of Cooperation*. Basic Books, 1984. 129, 168, 171, 177, 179

Robert Axelrod. *The complexity of cooperation: Agent-based models of competition and collaboration*. Princeton University Press, 1997. 3, 168, 169, 171

Robert L. Axtell and Joshua M. Epstein. *Growing artificial societies: social science from the bottom up*. MIT press, 1996. 168

Robert L. Axtell, Joshua M. Epstein, Jeffrey S. Dean, George J. Gumerman, Alan C. Swedlund, Jason Harburger, Shubha Chakravarty, Ross Hammond, Jon Parker, and Miles Parker. Population growth and collapse in a multi-agent model of the Kayenta Anasazi in Long House Valley. *Proceedings of the National Academy of Science*, 99(3):7275–7279, 2002. 172

Monica Babes, Enrique Munoz de Cote, and Michael L. Littman. Social reward shaping in the prisoner's dilemma. In *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS-08)*. 2008. 114

Avrim Blum, Mohammad Taghi Hajiaghayi, Katrina Ligett, and Aaron Roth.

Regret minimization and the price of total anarchy. *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing (STOC)*, 2008. 142

Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 3):7280–7287, 2002. 168

Tilman Borgers and Rajiv Sarin. Learning through reinforcement and replicator dynamics. *Journal of Economic Theory*, 77(1), 1997. 66, 70

Michael Bowling and Manuela Veloso. Rational and convergent learning in stochastic games. 2001. 65, 67, 82, 122

George W. Brown. Iterative solution of games by fictitious play. *Yale*, 1951. 48, 51, 63

Colin F. Camerer. *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press, 2003. 47, 55

Colin F. Camerer and Teck-Hua Ho. Experience-weighted attraction learning in normal form games. *Econometrica*, 67:827–874, 1999. 172, 194

Daniel Castillo and Ali Kerem Saysel. Simulation of common pool resource field experiments: a behavioral model of collective action. *Ecological economics*, 55(3):420–436, 2005. 171, 177, 188

Lars-Erik Cederman. *Emergent actors in world politics: how states and nations develop and dissolve*. Princeton University Press, 1997. 168

Miguel Costa-Gomes, Vincent Crawford, and Bruno Broseta. Cognition and behavior in normal-form games: An experimental study. *Econometrica*, 69(5):1193–1235, 2001. 3, 47, 55

Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a nash equilibrium. *ACM symposium on Theory of computing*, pages 71–78, 2006. 28

Peter Dayan and Yael Niv. Reinforcement learning and the brain: The good, the bad and the ugly. *Current Opinion in Neurobiology*, 18(2):185–196, 2008. 66

Enrique Munoz de Côte, Archie C. Chapman, Adam M. Sykulski, and Nicholas R. Jennings. Automated planning in adversarial repeated games. *Proceedings of the Twenty-sixth Conference on Uncertainty in Artificial Intelligence (UAI-10)*, 2010. 146

P.J. Deadman. Modelling individual behaviour and group performance in an intelligent agent-based simulation of the tragedy of the commons. *Journal of Environmental Management*, 56(3):159–172, 1999. 171, 177, 188

Guillaume Deffuant, David Neau, Frederic Amblard, and Grard Weisbuch. Mixing beliefs among interacting agents. *Advances in Complex Systems*, 3(01n04):87–98, 2000. 168

Mario Di Bernardo, Chris Budd, and Alan R. Champneys. *Piecewise-smooth Dynamical systems: Theory and Applications*. Springer-Verlag, 2008. 72, 73, 74, 76

Daniel Egnor. Iocaine powder. *ICGA Journal*, 23, 2000. 52, 53, 140, 147

Giorgio Fagiolo, Christopher Birchenhall, , and Paul Windrum. Special issue on empirical validation in agent-based models. *Computational Economics*, 30(3), 2007a. 172

Giorgio Fagiolo, Alessio Moneta, and Paul Windrum. A critical guide to empirical validation of agent-based models in economics: Methodologies, procedures, and open problems. *Computational Economics*, 30, 2007b. 5

Eugene F. Fama. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25, 1970. 4

Christina Fang, Jeho Lee, and Melissa A. Schilling. Balancing exploration and exploitation through structural design: The isolation of subgroups and organizational learning. *Organization Science*, 21(3):625–642, 2010. 4, 169

Urs Fischbacher, Simon Gachter, and Ernst Fehr. Are people conditionally cooperative? evidence from a public goods experiment. *Economics Letters*, 71(3):397–404, 2001. 21, 178, 179, 180, 182

Jean J. Gabszewicz and Jacques-Francois Thisse. Location. *Handbook of Game Theory with Economic Applications*, 1992. 142

Ya'akov Gal and Avi Pfeffer. Networks of influence diagrams: Reasoning about agents' beliefs and decision-making processes. *Journal of Artificial Intelligence Research (JAIR)*, 33:109–147, 2008. 54

Natalie S Glance and Bernardo A Huberman. The outbreak of cooperation. *Journal of Mathematical sociology*, 17(4):281–302, 1993. 179

Piotr Gmytrasiewicz and Prashant Doshi. A framework for sequential planning in multiagent settings. *Journal of Artificial Intelligence Research (JAIR)*, 24:49–79, 2005. 54, 56

Piotr J. Gmytrasiewicz and Edmund H. Durfee. A rigorous, operational formalization of recursive modeling. *Proceedings of the First International Conference on Multi-Agent Systems*, 1995. 56

Eduardo Rodrigues Gomes and Ryszard Kowalczyk. Dynamic analysis of multiagent Q-learning with e-greedy exploration. *Proceedings of the Twenty-sixth International Conference on Machine Learning (ICML-09)*, 2009. 71, 73, 109

Scott Heckbert, Tim Baynes, and Andrew Reeson. Agent-based modeling in ecological economics. *Annals of the New York Academy of Sciences*, 1185(1):39–53, 2010. 169

Teck-Hua Ho, Colin F. Camerer, and Keith Weigelt. Iterated dominance and iterated best response in experimental p-beauty contests. *American Economic Review*, 88:947–969, 1998. 3, 5

John H. Holland and John H. Miller. Artificial adaptive agents in economic theory. *The American Economic Review*, pages 365–370, 1991. 168

Harold Hotelling. Stability in competition. *The Economic Journal*, 39:41–57, 1929. 142, 146

Junling Hu and Michael P. Wellman. Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, 4:1039–1069, 2003. 68

Marco A. Janssen and T.K. Ahn. Learning, signaling, and social preferences in public-good games. *Ecology and Society*, 11(2):21, 2006. 170, 172, 177, 182, 186, 188, 194

Marco A. Janssen and Elinor Ostrom. Empirically based, agent-based models. *Ecology and Society*, 11(2):37, 2006. 169

Patrick R. Jordan, Michael P. Wellman, and Guha Balakrishnan. Strategy and mechanism lessons from the first ad auctions trading agent competition. *Proceedings of the Eleventh ACM Conference on Electronic Commerce (EC-10)*, 2010. 32, 147

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998. 56

Michael Kaisers and Karl Tuyls. Faq-learning in matrix games: Demonstrating convergence near nash equilibria, and bifurcation of attractors in the battle of sexes. *In Workshop on Interactive Decision Theory and Game Theory (IDTGT 2011). Assoc. for the Advancement of Artif. Intel. (AAAI)*, 2011. 70

Maureen Kilkenny and Jacques-Francois Thisse. Economics of location: A selective survey. *Computers and Operations Research*, 14(26):13691394, 1999. 142

Daphne Koller and Brian Milch. Multi-agent influence diagrams for representing and solving games. *Games and Economic Behavior*, 2003. 54

David M Kreps, Paul Milgrom, John Roberts, and Robert Wilson. Rational cooperation in the finitely repeated prisoners' dilemma. *Journal of Economic Theory*, 27(2):245–252, 1982. 179

Paul Krugman. How did economists get it so wrong? *The New York Times*, 2009. URL `http://www.nytimes.com/2009/09/06/magazine/06Economic-t.html`. 1

Thomas S. Kuhn. *The Structure of Scientific Revolutions*. University of Chicago Press, 1970. 3

David Lazer and Allan Friedman. The network structure of exploration and exploitation. *Administrative Science Quarterly*, 52(4):667–694, 2007. 3, 168, 169

John Ledyard. Public goods: A survey of experimental research. In John H. Hagel and Alvin E. Roth, editors, *Handbook of Experimental Economics*, pages 111–194. Princeton University Press, Princeton, NJ, 1995. 174, 188

Kevin Leyton-Brown and Yoav Shoham. *Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations*. Cambridge University Press, 2009. 25, 112

M. Littman and M. Zinkevich. The 2006 AAAI computer-poker competition. *International Computer Games Association Journal*, 29, 2007. 8

Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on Machine Learning (ICML-94)*. 1994. 68

Michael L. Littman. Friend-or-foe Q-learning in general-sum games. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML-01)*. 2001. 69

Michael L. Littman and Peter Stone. Implicit negotiation in repeated games. In *Eighth International ATAL Workshop (ATAL-2001)*. 2001. 67

Michael L. Littman and Peter Stone. A polynomial-time Nash equilibrium algorithm for repeated games. *Proceedings of the Fourth ACM Conference on Electronic Commerce (EC-03)*, 2003. 30, 113

M. Dolores Lopez, Javier Rodrigo, and Isabel Lillo. Two-party political competition: A geometric study of the nash equilibrium in a weighted case. *Applied Mathematical Sciences*, 1(55), 2007. 142

Dunia Lopez-Pintado and Duncan J. Watts. Social influence, binary decisions and collective dynamics. *Rationality and Society*, 20(4):399–443, 2008. 179

Aleksandr Mikhailovich Lyapunov. The general problem of the stability of motion. *International Journal of Control*, 55(3):531–534, 1992. 83

Michael W. Macy and Robert Willer. From factors to actors: Computational sociology and agent-based modeling. *Annual review of sociology*, pages 143–166, 2002. 3, 169, 171, 180

Alfred Marshall. *Principles of Economics*. Macmillan, 1890. 31

Winter Mason and Duncan J. Watts. Collaborative learning in networks. *Proceedings of the National Academy of Sciences*, 109(3):764–769, 2012. 4, 170

Richard McKelvey and Thomas Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 10:6–38, 1995. 39

Walter Mischel, Ebbe Ebbesen, and Antonette Raskoff Zeiss. Cognitive and attentional mechanisms in delay of gratification. *Journal of Personality and Social Psychology*, 21(2):204–218, 1972. 8

John Nash. Non-cooperative games. *The Annals of Mathematics*, 1951. 26

John Von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944. 25

Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003. 200, 201

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. 139

Rob Powers and Yoav Shoham. New criteria and a new algorithm for learning in multi-agent systems. *Advances in Neural Information Processing Systems*, 2005. 113

William H. Press and Freeman J. Dyson. Iterated prisoners dilemma contains strategies that dominate any evolutionary opponent. *Proceedings of the National Academy of Sciences*, 2012. 30, 115, 126

Debajyoti Ray, P. Read Montague, Brooks King-casas, and Peter Dayan. Bayesian model of behaviour in economic games. *Advances in Neural Information Processing Systems (NIPS)*, 2008. 57

David Rumelhart and James McClelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge: MIT Press, 1986. 64

Tuomas W. Sandholm and Robert H. Crites. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems*, 37:144–166, 1995. 71

Thomas C. Schelling. *Micromotives and Macrobehavior*. WW Norton, 1978. 168

Yoav Shoham, Rob Powers, and Trond Grenager. If multi-agent learning is the answer, what is the question? *Artificial Intelligence, special issue on Foundations of Multi-Agent Learning*, 171(7):365–377, 2007. 22

Herbert Simon. *Models of Bounded Rationality*. MIT Press, 1982. 31

Satinder Singh, Michael Kearns, and Yishay Mansour. Nash convergence of gradient dynamics in general-sum games. 2000. 64, 67, 78

Dale O. Stahl and Paul W. Wilson. On players' models of other players: Theory and experimental evidence. *Games and Economic Behavior*, pages 218–254, 1995. 53, 55

Peter Stone and Amy Greenwald. The first international trading agent competition: Autonomous bidding agents. *Electronic Commerce Research*, pages 229–265, 2005. 140, 147

Siddharth Suri and Duncan J. Watts. Cooperation and contagion in web-based, networked public goods experiments. *PLoS One*, 6(3), 2011. 171, 173, 174, 185, 188, 197

Richard Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. *Proceedings of the 7th International Conference on Machine Learning*, 1990. 41

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998. 61, 78

Matthew E. Taylor and Peter Stone. Cross-domain transfer for reinforcement learning. *Proceedings of the Twenty-fourth International Conference on Machine Learning (ICML-07)*, 2007. 139

John N. Tsitsiklis. Asynchronous stochastic approximation and Q-learning. *Machine Learning*, 16(3):185–202, 1994. 66

Karl Tuyls, K. Verbeeck, and T. Lenaerts. A selection-mutation model for Q-learning in multi-agent systems. *Proceedings of Second International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2003)*, 2003. 67, 69

Amos Tversky and Daniel Kahneman. Judgment under uncertainty: Heuristics and biases. *Science*, (185):1124–1130, 1974. 34

Jing Wang, Siddharth Suri, and Duncan Watts. Cooperation and assortativity with dynamic partner updating. *Proceedings of the National Academy of Sciences*, 109(36):14363–14368, 2012. 171, 178, 180

Christopher Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3):279–292, 1992. 66

John Burr Williams. *The Theory of Investment Value*. Harvard University Press, 1938. 41, 181

James R. Wright and Kevin Leyton-Brown. Beyond equilibrium: Predicting human behavior in normal form games. *Proceedings of the Twenty-fourth Conference on Artificial Intelligence (AAAI-10)*, 2010. 55, 155

James R. Wright and Kevin Leyton-Brown. Behavioral game-theoretic models: A bayesian framework for parameter analysis. *Proceedings of the Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS-12)*, 2012. 56, 172

Michael Wunder, Monica Babes, and Michael Littman. Classes of multiagent q-learning dynamics with epsilon-greedy exploration. *Proceedings of the Twenty-seventh International Conference on Machine Learning (ICML-10)*, 2010. iv, 63, 67, 71

Michael Wunder, Michael Kaisers, Michael Littman, and John Robert Yaros. Using iterated reasoning to predict opponent strategies. *Proceedings of the Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-11)*, 2011. iv, 15

Michael Wunder, Michael Kaisers, Michael Littman, and John Robert Yaros. A framework for modeling population strategies by depth of reasoning. *Proceedings of the Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS-12)*, 2012. iv, 15

Michael Wunder, Siddharth Suri, and Duncan Watts. Emprical agent based models of cooperation in public goods games. *Proceedings of the Fourteenth ACM Conference on Electronic Commerce (EC-13)*, 2013. iv, 19

E. Zawadzki, A. Lipson, and K. Leyton-Brown. Empirically evaluating multi-agent learning algorithms. *Working Paper*, November 2008. 71

Martin Zinkevich. The lemonade game competition. http://tech.groups.yahoo.com/group/lemonadegame/, December 2009. 140, 145

Martin Zinkevich, Michael Bowling, and Michael Wunder. The lemonade stand game competition: Solving unsolvable games. *ACM SIGecom Exchanges*, 10, 2011. 164