LEARNING, EVOLUTION, AND BAYESIAN ESTIMATION IN GAMES AND DYNAMIC CHOICE MODELS

by

ALEXANDER MONTE CALVO

A DISSERTATION

Presented to the Department of Economics
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

June 2014

DISSERTATION APPROVAL PAGE

Student: Alexander Monte Calvo

Title: Learning, Evolution, and Bayesian Estimation in Games and Dynamic Choice Models

This dissertation has been accepted and approved in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Economics by:

Van Kolpin	Chair
Jeremy Piger	Core Member
Ralph Mastromonaco	Core Member
Shawn Lockery	Institutional Representative

and

Kimberly Andrews Espy	Vice President for Research & Innovation/ Dean of the Graduate School

Original approval signatures are on file with the University of Oregon Graduate School.

Degree awarded June 2014

DISSERTATION ABSTRACT

Alexander Monte Calvo

Doctor of Philosophy

Department of Economics

June 2014

Title: Learning, Evolution, and Bayesian Estimation in Games and Dynamic Choice Models

This dissertation explores the modeling and estimation of learning in strategic and individual choice settings. While learning has been extensively used in economics, I introduce the concept into standard models in unorthodox ways. In each case, changing the perspective of what learning is drastically changes standard models. Estimation proceeds using advanced Bayesian techniques which perform very well in simulated data.

The first chapter proposes a framework called Experienced-Based Ability (EBA) in which players increase the payoffs of a particular strategy in the future through using the strategy today. This framework is then introduced into a model of differentiated duopoly in which firms can utilize price or quantity contracts, and I explore how the resulting equilibrium is affected by changes in model parameters.

The second chapter extends the EBA model into an evolutionary setting. This new model offers a simple and intuitive way to theoretically explain complicated dynamics. Moreover, this chapter demonstrates how to estimate posterior distributions of the model's parameters using a particle filter and Metropolis-Hastings

algorithm, a technique that can also be used in estimating standard evolutionary models. This allows researchers to recover estimates of unobserved fitness and skill across time while only observing population share data.

The third chapter investigates individual learning in a dynamic discrete choice setting. This chapter relaxes the assumption that individuals base decisions off an optimal policy and investigates the importance of policy learning. Q-learning is proposed as a model of individual choice when optimal policies are unknown, and I demonstrate how it can be used in the estimation of dynamic discrete choice (DDC) models. Using Bayesian Markov chain Monte Carlo techniques on simulated data, I show that the Q-learning model performs well at recovering true parameter values and thus functions as an alternative structural DDC model for researchers who want to move away from the rationality assumption. In addition, the simulated data are used to illustrate possible issues with standard structural estimation if the rationality assumption is incorrect. Lastly, using marginal likelihood analysis, I demonstrate that the Q-learning model can be used to test for the significance of learning effects if this is a concern.

CURRICULUM VITAE

NAME OF AUTHOR: Alexander Monte Calvo

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:
University of Oregon, Eugene, OR
Pacific Lutheran University, Tacoma, WA

DEGREES AWARDED:
Doctor of Philosophy Economics, 2014, University of Oregon
Bachelor of Arts Mathematics/Economics, 2008, Pacific Lutheran University

AREAS OF SPECIAL INTEREST:
Game Theory, Bayesian Econometrics, Behavioral Economics

PROFESSIONAL EXPERIENCE:

Graduate Teaching Fellow, University of Oregon, 2009-2014

Analyst, Bank of New York Mellon, 2008-2009

GRANTS, AWARDS AND HONORS:

Best GTF Instructor, University of Oregon Economics Dept., 2012/2013

Best Field Paper, University of Oregon Economics Dept., 2012

ACKNOWLEDGEMENTS

I would like to thank professors Van Kolpin, Jeremy Piger, and Ralph Mastromonaco for always listening and giving suggestions or feedback no matter how disjointed the idea or flustered my presentation of it.

For Taylor, without whom none of this would have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

Figure Page

LIST OF TABLES

Table Page

Table Page

CHAPTER I

INTRODUCTION

The overarching theme of this dissertation is learning. Learning is not a new concept in economics; in fact it has permeated almost every major topic within the field. However, as you will find, I introduce the notion of learning in new and compelling ways.

Typically, learning in economics focuses on individuals gathering information over time and using it to inform them as to what action to take. This information may be about each option's performance directly via past experiences, or it may be about unknown parameters that in turn affect the performance of other options. The first two chapters take a much different perspective on what learning means. Specifically, I model how individuals get better at utilizing available options through previous experience.

The first chapter introduces one formal way of modeling this idea of learning in a game theoretic context, which I call Experienced-Based Ability. In this model, players gain experience in a particular strategy when they utilize it. This increase in experience translates to an increase in payoffs from using that strategy in the future. In addition to some basic examples, this framework is introduced into the Singh and Vives (1984) model of differentiated duopoly. Singh and Vives presented a model where firms could choose to utilize price or quantity contracts, and found that it was a dominant strategy to choose quantity contracts in the case of substitutes if firms could commit to a contract type. When EBA is introduced into this framework, I explore how the resulting Markov equilibrium changes as the model parameters change. It is found that the equilibrium can feature both firms always using quantity

contracts as before, but changes in parameter values generate equilibrium where firms find it optimal to choose opposite contract types, and where both choose to utilize price contracts.

The second chapter extends this model into an evolutionary setting. Specifically, I assign each type within a population ability levels at dealing with other types. These ability levels affect contemporary fitness outcomes and are passed down to offspring. The more prevalent a type is within the population, the more skilled others will become at dealing with it. This new model offers a simple and intuitive way to theoretically explain complicated dynamics, even in the case of a 2 strategy population. That is, the standard evolutionary model only allows for monotonic adjustment to a steady state in the 2 strategy case, whereas the proposed model allows for much more complicated dynamics, e.g. limit cycles.

The third chapter moves back to the traditional notion of learning, but introduces it into a discrete dynamic choice model. Learning is something that has been absent from discrete dynamic choice models in economics. Most structural estimation models assume that individuals have solved for an optimal policy function, but in reality this solution is extremely hard to find, even with modern computing power. While the machine learning literature has extensively modeled learning optimal policy functions in dynamic choice environments, economics lacks an estimable model that accounts for individuals learning policy functions over time. In the third chapter, I demonstrate that the Q-learning model is a simple and flexible model of policy learning, and show that this model can be easily used for estimation of dynamic discrete choice (DDC) models. Using Bayesian MCMC techniques on simulated data, I show that the Q-learning model performs well at recovering true parameter values. In addition, the simulated data are used to illustrate possible issues with standard

structural estimation if the rationality assumption is incorrect. Lastly, using marginal likelihood analysis, I demonstrate that the Q-learning model can be used to test for the significance of learning effects.

In addition to contributing to the current theoretical literature on learning, this dissertation also demonstrates and develops several advanced computational techniques. The first chapter utilizes value function iteration to approximate Markov equilibria. The second chapter estimates parameters of a non-linear, non-Gaussian state space model using a particle filter and an MH-algorithm. Lastly, the third chapter demonstrates how to estimate the proposed Q-Learning model of discrete dynamic choice, and compares it to other leading dynamic discrete choice estimation models.

CHAPTER II

PRACTICE MAKES PERFECT

Introduction

Learning is a fundamental element of human nature and civilization. Indeed, people around the world spend the better part of their first two decades of life in school, and others up to a quarter or more of their entire life. As such, it is only appropriate that notions of learning have been incorporated into economic models. In fact, learning now plays an important role in both micro and macro economics. The current idea behind most learning models is that individuals analyze the past, forecast the future, and observe other players' actions in an attempt to learn which strategy should be chosen. But this is only a part of what real learning involves.

As Section II lays out in more detail, current models capture a very important element of what real life learning is all about. Sports teams record and watch themselves and their opponents in order to develop a particular action plan for upcoming games. Students watch diligently as professors demonstrate solution techniques. Musicians listen to their favorite artists for inspiration. But what do all these people do afterward? They practice, relentlessly perfecting their carefully chosen plan of action. Indeed, without practice and the accumulation of experience, simply knowing *what* to do would not yield perfect results without also knowing *how* to execute these plans.

The main aim of this paper is to develop a framework for including this practice and experience element into games. The basic idea is that playing a particular strategy today will increase the payoffs of that strategy in the future. In a repeated game form,

players are assigned ability levels for each of their available strategies in the stage game. Payoffs for each player are functions of players' ability levels, and their ability levels increase/decrease depending on the history of chosen actions. I call these types of games "Games with Experienced-Based Ability (EBA)." This setup creates games with rich dynamics where the structure of the stage game changes endogenously, and I focus on finding Markov equilibria. In addition to some generic examples, I also introduce the EBA framework into a model of differentiated duopoly. Singh and Vives (1984) presented a model where firms could choose to utilize price or quantity contracts, and found that it was a dominant strategy to choose quantity contracts in the case of substitutes if firms could commit to a contract type. When EBA is introduced into this framework, I explore how the resulting Markov equilibrium changes as the model parameters change. It is found that the equilibrium can feature both firms always using quantity contracts as before, but changes in parameter values generate equilibrium where firms find it optimal to choose opposite contract types, and where both choose to utilize price contracts.

The remainder of the chapter is laid out as follows: Section II reviews the current state of learning models and discusses what features will be desirable when incorporating "practice." Section III presents the basic framework of Experienced-Based Ability. Section IV develops my preferred equilibrium focus for these games, and Section V provides various numerical examples. Section VI extends the EBA framework to a model of differentiated duopoly. Section VII discusses possible directions for future research and Section VII concludes.

Learning and Practice - A Discussion

Current Learning

Learning has become a very important element of both micro and macro economics. Within the field of game theory, learning models are often used to justify Nash Equilibrium outcomes. In these models, agents look at the past performance of their actions and their opponents' actions to inform which action to take in the current period. It is then examined whether or not play will converge to that predicted by the Nash Equilibria. In macro economics, learning has also gained a prominent position. For example, the models such as those used in Brock and Hommes (1998) have heterogeneous agents choose different belief systems or forecasting techniques. The prevalence of each type used in the population is directly linked to the type's past performance.

The vast majority of learning models in economics are focused on individuals learning *what* strategy is best to play. While this is a very important element of learning to model, it is not the only or necessarily most important one on which to focus. As I have been arguing, it may also be of equal importance to focus on individuals learning *how* to implement their chosen actions. While there are very few economic models which have this focus, there are some that do; one such example being the Learning By Doing literature.

The main example of the practice or experience idea being incorporated into current economic models is that of Learning By Doing (LBD). LBD began to arise in the middle of the last century, motivated by the observation that many firms experienced decreasing marginal costs over time. Empirically, a great deal of work has been done examining learning by doing within the semiconductor industry. Most

studies incorporate learning by doing by assuming that unit costs, usually captured by price observations, depend on factors such as cumulative output, time, or other proxies for experience such as engineering time devoted to a specific process. Gruber (1992) finds that erasable programmable read only memory (EPROM) production is characterized by significant learning effects, mostly driven by cumulative output. Hatch and Reichelstein (1995) use yield data, instead of prices, to infer unit costs and find persistent learning effects which are driven by cumulative output and engineering time.

Another focus within this literature is whether or not learning spillovers exist. For example, Irwin and Klenow (1994) examine the production of dynamic random access memory (DRAM) chips and find evidence that spillover learning is present between firms and countries, but that internal learning is more important. However, the authors only find weak evidence of learning spillovers between generations of DRAM chips. Similarly, Gruber (1998) examines EPROM production and finds that spillover learning is present between firms, but internal learning still dominates. Following the significant amount of evidence that learning by doing exists within firms, many theoretical models were developed in order to analyze the implications of learning by doing for firm behavior, market structure, and policy.

Examining the implications of learning by doing for firm behavior, Spence (1981) models the typical assumption that marginal costs are a function of cumulative output. Focusing on open-loop equilibrium (wherein firms select best responses given the entire output paths of competitors), Spence finds that learning can create significant barriers to entry, and thus has can have an impact on market shares. Fudenberg and Tirole (1983) use a similar model to Spence, but focus on perfect equilibrium, wherein firms' optimal strategies dictate a best response to any possible

course of action by competitors. They find that for a monopolist (or social planner) output will increase over time, while it may decrease over time in the strategic setting of duopoly. An important element of both models is that a firm that recognizes the existence of LBD will find it optimal to produce where current unit costs are greater than price. This is because the true marginal costs of production account for the future decreases in unit costs. While most theoretical models focus on the relationship between unit costs and cumulative production, some have offered alternative models of learning by doing. For example, Jovanovic and Nyarko (1996) present a one-agent Bayesian model of learning by doing. In their model, an agent chooses a production technology and learns about its parameters through continued use. Switching to a "better" technology is costly in that the agent must start learning about this new technolog's parameters. This feature makes overtaking possible. That is, an agent may gain so much experience they will choose not to switch to a higher technology, while a less experienced agent may choose to continue switching to ever and ever more productive technologies. LBD was later utilized in macroeconomics to develop models of endogenous growth. In these models, LBD is present in that technology (productivity) is a function of the current capital stock (Thompson 2010).

The theoretical models began to show that LBD had broad implications for social and trade policy. Dick (1991) shows that because firms experiencing LBD may produce at a loss in current periods, they may be incorrectly found guilty of dumping in foreign markets. Dasgupta and Stiglitz (1988) find that LBD gives support to protecting infant industries, depending on relative learning effects in foreign and domestic industries. The authors also find that there is a tendancy for a dominant firm to emerge in industries with significant learning effects and that a monopoloy may be socially preferred to other market structures such as Duopoly.

Clearly, LBD models exhibit the ideas of "practice;" i.e. producing more today will make it easier to produce in the future. Unfortunately, these models are designed to be very specific. The main restriction, and distinction between the proposed model and LBD, is that LBD is one dimensional. For example, the LBD models of firm cost have no inclusion of other factors, such as quality. That is, a firm who chooses to mass produce may indeed see lower unit costs, but may also see a decrease in quality. Relating this to the proposed framework, this could be imagined as a game wherein a firm had ability levels in quantity and quality, and increasing one necessarily decreases the other. Thus, it is difficult to look to LBD as representing a thorough framework for incorporating "practice."

In the EBA framework I later present in Section III, the choices agents make today affect their payoffs in later stages of the game. This type of framework bares similarities to others currently used in several areas of game theory and economics. One closely related field is that of Common-Pool Resource (CPR) games and Bioeconomics. Initially, the economic theory behind these games was static in nature,[1] but as the field matured it became dominated by much more dynamic models. Indeed, most CPR and Bioeconomic models currently use a dynamic or stochastic game framework. The basic idea behind these models is that agents choose a harvesting plan of a renewable resource. The resource(s) being extracted have base population dynamics that are then affected by the agents' harvesting plans. A common focus is whether or not the harvesting plans implemented by the different agents are sustainable or not (i.e. will they allow for the continued existence of the resource, or will they eventually drive it into extinction.) These models can become very complex, allowing for multiple agents, and multiple resources which interact with

[1] e.g. see Haveman (1973)

each other as well as the agents' harvesting; e.g. predatory-prey populations being harvested (Clark 2010, Conrad and Clark 1987).

The most closely related model in game theory to the EBA framework is that of Frequency-Dependent (FD) payoff games. Introduced by Brenner and Witt (2003), and expanded upon in Joosten, Brenner, and Witt (2003), FD games are similar in that the payoffs in stage games are functions of the relative frequency of actions chosen throughout the history of the game. Interestingly, FD games can also be used to model certain CPR games.[2] However, FD games fall short of EBA games in several aspects. As I demonstrate in the next section, FD games are actually a specific subclass of EBA games, wherein ability is simply measured as the relative frequency of strategies throughout a game's history. Because of this, FD games are extremely limited in terms of being able to incorporate the idea of practice. Their equilibrium focus is typically a situation where the relative frequencies reach a steady state, and thus the structure of the stage game reaches a steady state as well. As is demonstrated later on, when EBA games reach a steady state in terms of relative frequency in equilbrium, they can still exhibit changing abilities and thus changes in the stage game even after relative frequencies settle down. Lastly, the focus of FD games is often times on how playing a particular strategy changes the environment in which the game is being played, and not always on how playing a strategy changes the *player* using it (although this *could* be an interpretation). Thus, while possibly applicable, FD games are not specifically interested in learning.

[2]e.g. a common game used in the FD literature is the "Pollution" game

Practice and Ability

The basic idea behind games involving EBA is that your strategy choice today affects your expected payoffs in the future, and that, more specifically, use of a particular strategy today will increase the expected payoff from using this same strategy tomorrow. This approach seems to be one of the most flexible ways of incorporating notions of "practice" into a game. It is also a reduced form of several other approaches for modeling "practice." For example, an intuitive way to model the idea of practice would be to allow for failure in the stage game. That is, for each strategy available, a player can succeed or fail with a certain probability. As your experience with the strategy increases, the probability of failure decreases, and thus the expected payoff increases. Thus, simply making payoffs functions of ability could be interpreted as modeling this situation. Another way to interpret or use EBA is for modeling extremely complicated games. For example, in many real-world games there are classes of strategies (e.g. running or passing in football) which, once chosen, then involve a mind-boggling amount of exact timing and execution. Each timing and particular execution is an individual strategy, but there is one perfect mix. These better options may not be available to players until they have used the strategy an appropriate number of times. Thus, again, in this situation expected payoffs would be increasing in player ability.[3]

So, what features of the real world should be allowed for as possibilities in a general framework? One important element of real-world practice is that while practice may make perfect, it is also costly. Many musical instructors will tell you

[3]Another possible way to model the idea of practice would be a setup where players could choose to incur a cost in order to practice basic skills which would later enhance payoffs of other strategies. While this is a reasonable type of model to look at, EBA is a much more general and tractable way to incorporate experience and practice.

that practice should sound bad. The point is to get better and learn from mistakes, so of course you will not and should not sound perfect while practicing. This represents one of the most fundamental tradeoffs that experienced-based ability models allow for; playing a strategy to gain experience versus playing a strategy for immediate gain. This tradeoff is very similar to others in economics. For example, in a macroeconomic setting, individuals must often tradeoff consumption today for investment in capital to produce more tomorrow. Indeed, it is often this very tradeoff that distinguishes those who succeed in a particular arena and those that do not flourish. Only those with enough concern about the future are willing to sacrifice enough immediate gain in order to be better later on. Thus, any general framework for incorporating "practice" should allow for the prescence of this tradeoff.

Another feature of the real world is that there are often interconnections between practicing one strategy, and the ability of another. For instance, there are often circumstances where specializing in one strategy may lower your ability in another. This could arise directly, or it could arise because of a "use it or lose it" situation; i.e. the longer you go without using a strategy, the lower your ability becomes. However, there are certainly situations where becoming better at a strategy also increases your ability in other, similar strategies. One such example of this, learning spillovers between semiconductor generations, has already been discussed. Furthermore, there are most certainly relationships between the abilities of *opponents* and your own payoffs. Thus, payoffs should be allowed to be functions not only of the players' own abilities, but also of their opponents' abilities. These are all elements that should be available in a general framework.

Framework

Model Setup

With the previous discussion in mind, I will now set up the basic framework of EBA models. The setting is in an infinitely repeated game form, wherein it is assumed that, for simplicity, each stage is a simultaneous move game (although this is not required). For each player, let $s_i \in S_i$ denote a strategy for player i *in a particular stage*. This paper focuses on the case of finite stage-game strategy spaces; i.e. S_i is finite for all players. Furthermore, let $\mathbf{s} \in S$ represent a strategy profile in a particular stage, where $S = \Pi_{i=1}^{N} S_i$. The majority of this paper focuses on the case of two players; i.e. $S = S_1 \times S_2$. In the case of a finite strategy space, a player's current ability is simply a vector describing an ability level for each available strategy in the stage game. That is, if $|S_i| = K$, then that player's ability will be represented by $\mathbf{a}_i \in \mathbb{R}^K$ For now, let a_i^s denote the ability of player i in playing strategy $s \in S_i$. A player's payoffs are represented by $\pi_i(s, s_{-i}, \mathbf{a}_i, \mathbf{a}_{-i})$, or more simply $\pi_i(\mathbf{s}, \mathbf{a})$ where $\mathbf{a} \in \mathbb{R}^{K_1} \times \mathbb{R}^{K_2} \times \cdots \times \mathbb{R}^{K_N}$.

The last, and perhaps most crucial, element of the EBA framework is an ability evolution specification. That is, exactly how ability changes from one period to the next needs to be specified. Throughout the remainder of the paper, let $\alpha(\cdot)$ denote the functional relationship describing tomorrow's ability in a particular strategy as a function of time, current ability levels of all players, and the particular strategy profile chosen in the current period. That is, it is generally be the case that:

$$a_{it+1}^s = \alpha_i^s(\mathbf{a}_t, \mathbf{s}_t, t) + \epsilon_{t+1} \tag{2.1}$$

Where ϵ_{t+1} represents a stochastic shock, typically with $E[\epsilon_{t+1}] = 0$. The above general specification allows for a great deal of possibilities as far as ability evolution is concerned. While many of these possibilities are discussed in Appendix 2, the majority of the paper focuses on a more specific form of ability evolution, wherein a player's future ability in a particular strategy only depends on the player's own action choices:

$$a^s_{it+1} = \alpha^s_i(a^s_{it}, I_t(s_t = s)) + \epsilon_{t+1} \tag{2.2}$$

Where $I_t(s)$ represents an indicator function for whether strategy s was chosen by player i in period t. Again, note the drastic difference between the learning modeled here, and that of most previous learning models. The model is not interested in the process of an agent learning which strategy to choose. It is instead focused on modeling how agents improve their ability to use available strategies. While Appendix A presents some detailed examples of ability specifications, the remainder of the paper focuses on a specification in which ability is simply a function of last period's ability and strategy choice. For example, consider the below deterministic ability function:

$$\begin{aligned} a^s_{it+1} &= a^s_{it} + [I_t(s_t = s)(\mu(1 - a^s_{it})) - (1 - I_t(s_t = s))\mu a^s_{it}] \\ &= a^s_{it} + \mu(I_t(s_t = s) - a^s_{it}) \end{aligned} \tag{2.3}$$

Where $\mu \in [0, 1]$ is a parameter determining the size of adjustment. In this specification, ability is bound between 0 and 1. Note that only the current value of a^s_{it} and the chosen action need to be known in order to determine what the next period ability level will be.

A simplification to ability evolution that is utilized later is letting the ability in one strategy to be directly tied to the ability in the other. For example, suppose

$S_1 = \{A, B\}$ in the stage game for player 1. Instead of creating two ability evolution specifications for each strategy, it could simply be that:

$$(a_{1t}^A) = 1 - (a_{1t}^B) \tag{2.4}$$

This specifications such as these exhibit a feature wherein increasing your ability in A necessarily decreases your ability in B. This feature is similar to those exhibited by the models of Jovanovich and Nyarko (1996) and Klenow (1998) which exhibited decreases in productivity immediately following a switch to a different technology choice. After choosing a specification for either a_1^A or a_1^B, calculating the other is very simple. A full specification could indeed be worked out for the other, but it would be unnecessary. As is discussed later, simplifications such as this will buy a smaller state space for any problem in which the state space is the players' ability levels because there won't be a need to track all ability levels as some are linked. This also makes any numerical analysis much easier and quicker. As it turns out, the cost of this assumption may be very little in some circumstances.

Consider the below deterministic ability specification:

$$a_{it+1}^s = a_{it}^s + [I_t(s_t = s)(\mu(G(a_{it}^s))) - (1 - I_t(s_t = s))\mu L(a_{it}^s)] \tag{2.5}$$

Assume there are only two strategies, A and B, and both follow the above specification. If the gain and loss functions, $G(\cdot)$ and $L(\cdot)$, satisfy $G(1-x) = L(x)$ then the system has $a_t^A = 1 - a_t^B$ as a steady state.[4] That is, if you define $x_t = a_t^A + a_t^B$, it can be shown that $x_t = 1$ is a steady state. Thus, in many situations, the assumption $(a_{1t}^A) = 1 - (a_{1t}^B)$ costs very little.

[4]See appendix for proof

Allowing for specifications such as those above implies that a_{it} may never be constant for a player. Instead, it will usually be fluctuating. However, depending on the parameters, players could conceivably reach a "steady state", depending on their chosen strategies, such that $Pr(a_{it+j} \notin R) < \delta \; \forall j > 0$. That is, while their ability levels are never constant, their strategy choices are such that ability is bound within some range after a certain point in the game. Indeed, Pakes and McGuire (2001) utilize such a situation in their algorithm to compute Markov Perfect equilibrium. Note that this does not imply that players necessarily specialize in one strategy. Rather, players could be able to continue mixing appropriately over several strategies, keeping their ability levels relatively constant in each one. In fact, this may be optimal. Often times, if a person can only do one action well, they may be predictable and easy to exploit. To keep yourself from being easily exploitable, it may be worth it to you to invest time developing ability, or at least competence, in multiple strategies and maintaining this level of ability.

To summarize, the EBA framework simply adds two new elements to repeated games. First, payoffs are functions of abilities (in addition to the chosen strategy profile). Second, an appropriate ability evolution function is added to the game, such as those presented above. Thus, as players choose a particular strategy, their ability in it increases, and can expect higher payoffs from playing it next period. At this point, it is useful to note that, just like FD and many CPR games, games using the EBA framework are actually a very specific type of dynamic game.[5] This implies that analyzing EBA games with standard notions of subgame perfection or simple Nash Equilibrium may allow for an unlimited set of possible outcomes (Dutta 1995).

[5]They may also be classified as stochastic games, especially if the ability evolution function itself had a stochastic element to it.

Throughout the remainder of this chapter, I focus on Markov perfect equilibria in which all players find it optimal to utilize policy functions.[6]

Solving for a Markov Perfect Equilibrium

This section discusses Markov perfect equilibria involving policy functions for a specific class of EBA games.In what follows, the use of the phrasing policy functions, rather than markov strategies, is intended to keep the focus on the solution technique; i.e. policy/value function iteration. The class of EBA games examined here are those in infinitely repeated game form with finite strategy spaces for all players. Furthermore, I assume that each player has an assigned ability-evolution function that only depends on current ability and personal action choice. That is, for each player, future ability can be described by the following functional relationship:

$$a^s_{it+1} = \alpha_i(a^s_{it}, I_t(s)) + \epsilon_{it+1} \tag{2.6}$$

In this case, tomorrow's ability in strategy s is determined by today's ability and whether or not s is chosen today. Lastly, I assume that all players are attempting to maximize the expected sum of discounted payoffs.

[6]This is commonly referred to as an equilibrium consisting of stationary strategies in the stochastic game literature.

Policy Functions

To begin discussing what a Markov perfect equilibrium consists of, an appropriate state variable or space must be defined. I define the state space for the problem as the current levels of ability for all players. Let A_i denote the ability space for player i; where $A_i = \mathbb{R}^{K_i}$ with finite strategy spaces. The state space can be defined as $X = A_1 \times A_2 \times \cdots \times A_N$, and let $\mathbf{a}_t \in X$ denote the current state in period t. Then, a policy function for a player dictates what to do for each possible state.

Given the appropriate state space, X, a player could then choose to utilize a policy function as a strategy for the entire game. I assume that a player's policy function describes what stage-game strategy (action)[7] to use in the current period depending on the ability levels. That is, a policy function is defined by:

$$\boldsymbol{\sigma}_i : X \to S_i \tag{2.7}$$

Equilibria

Recall that the players will attempt to maximize their total expected discounted payoffs subject to the other players' chosen strategies. Assume now that player i discounts future payoffs at rate β and faces a situation wherein all other players have chosen to use a policy function. Given the policy functions of the other players, player i can find a best response that maximizes the sum of expected discounted payoffs. In doing so, the below Bellman Equation represents a necessary condition for optimality:

$$V(\mathbf{a}_t|\boldsymbol{\sigma}_{-i}) =_{s \in S_i} (\pi_i(s, \boldsymbol{\sigma}_{-i}, \mathbf{a}_t) + \beta E[V(\mathbf{a}_{t+1}|\boldsymbol{\sigma}_{-i})]) \tag{2.8}$$

[7]You could also allow the player's policy to indicate what mixed strategy to use in the current period, in which case $\boldsymbol{\sigma}_i : X \to \Delta S_i$

	A	B
A	$(1 + a_1^A + a_2^A), (1 + a_1^A + a_2^A)$	$(a_1^A + a_2^A - 1), (2 + a_2^B - a_1^A)$
B	$(2 + a_1^B - a_2^A), (a_2^A + a_1^A - 1)$	$(1 + 0.5a_1^B + 0.5a_2^B), (1 + 0.5a_1^B + 0.5a_2^B)$

TABLE 1. Numerical Example Stage Game

The solution to the above Bellman equation yeilds a policy function for player i, and thus it is a best response for player i to use a policy function when all other players use a policy function themselves. Now denote the policy found from the above Bellman equation as $\boldsymbol{\sigma}_i^*(\cdot|\boldsymbol{\sigma}_{-i})$. A Markov perfect equilibrium would then be a set of best responses, $\boldsymbol{\sigma}_i^*(\cdot|\boldsymbol{\sigma}_{-i}^*)$, for each player.

In this chapter, I utilize the brute force method for approximating these equilibria. This consists of using value function iteration to find best response policies, and then iterating these policies until a fixed point is reached. In order to illustrate this process, and to further clarify the above discussion, I work through an example of finding such an equilibrium in policies in Section V. As the number of players grows, this process will grow less feasible. Pakes and McGuire (2001) propose a stochastic algorithm which approximates symmetric policy functions in the space of recurrent points for that equilibrium. Weintraub et.al. (2005) propose a method for finding "oblivious" equilibrium for games with large numbers of players, wherein each player's policy depends only on their own state.

Numerical Example

In this section, I present a simple example of an EBA game. The game consists of a simple 2x2 stage game (with strategies A and B), listed below. Ability is being measured on a scale of [0,1]. Note that this game exhibits payoffs which are increasing functions of the chosen strategy ability level. The game is symmetric in the sense that

the payoff functional forms are the same for both players. However, depending on the current state, any particular stage game may not be symmetric. In this particular game, the state space is easily divided into three regions which each exhibit distinct forms of stage games:

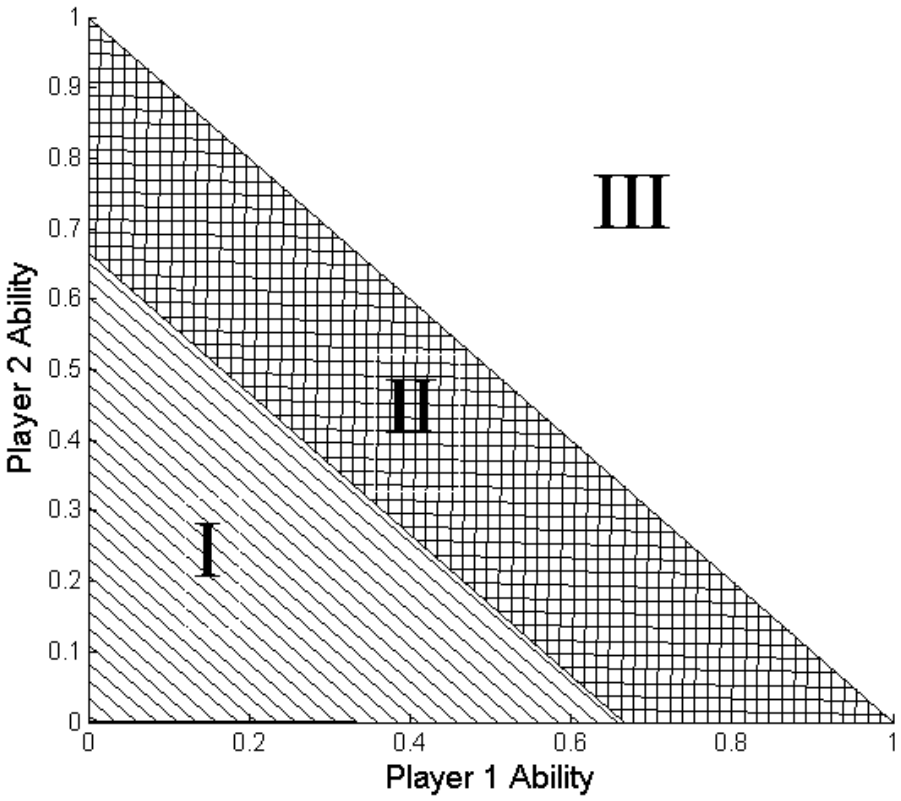


FIGURE 1. Example Game - Regions

In Region III, the stage game will be a coordination game. That is, the stage game has the below form, where best responses for each player are indicated with an asterisk: In Region II, $D_i > H_i > S_i > L_i$ and thus the stage game will be a Prisoner's Dillema game: Finally, in Region I, the stage game will still have (B,B) as a NE, but will not be a Prisoner's Dilemma because $H_i < S_i$.

	A	B
A	(H_1^*, H_2^*)	(L_1, D_2)
B	(D_1, L_2)	(S_1^*, S_2^*)

TABLE 2. Region I Stage Game

	A	B
A	(H_1, H_2)	(L_1, D_2^*)
B	(D_1^*, L_2)	(S_1^*, S_2^*)

TABLE 3. Region II Stage Game

Equilibrium Policies

Before the equilibrium policy functions are explored, it is necessary to specificy an ability evolution function for the game (as it would not be a complete EBA game without this). Both games utilize the ability evolution function described in Equation (**??**), except now ability evolution is stochastic:

$$a_{it+1}^s = a_{it}^s + \mu(I_t(s_t = s) - a_{it}^s) + \epsilon_{it+1} \tag{2.9}$$

Where ϵ_t is distributed truncated multivariate normal with mean $[0, 0]'$ and covariance matrix Σ. I also utilize the assumption that $(a_i^A) = 1 - (a_i^B)$. This is an example of ability evolution where not playing a strategy in a period means you lose some ability in that strategy. This type of ability evolution simplifies the problem greatly because the state space can simply be represented by $X = [0, 1] \times [0, 1]$. That is, players only need to know a_1^A and a_2^A to fully characterize any stage game. In light of this simplification, I utilize a slight departure in notation and let the current state be

denoted by $\mathbf{a}_t \equiv (a_{1t}^A, a_{2t}^A)$. Lastly, assume that each player has a common discount rate, β.

For a given opponent policy function, σ_j, Player i can solve for an optimal policy, $\boldsymbol{\sigma_i}^*$. I approximate this optimal policy using grid-based value function iteration, closely following Rust (1997). That is, given a grid of N points in the state-space, assign initial values V_0^n to each point, $\mathbf{a}^n = (a_1^n, a_2^n)$. For each grid point, n, at each iteration, g, calculate the value of each action, $\hat{V}_{A,g}^n, \hat{V}_{B,g}^n$ as:

$$\hat{V}_{s,g+1}^n = \pi_i(s, \sigma_j(\mathbf{a}^n), \mathbf{a}^n) + \beta * \sum_{k=1}^{N} \frac{\phi(\mathbf{a}^k|\mathbf{a}^n, s, \sigma_j)V_g^k}{\sum \phi(\mathbf{a}^k|\mathbf{a}^n, s, \sigma_j)} \tag{2.10}$$

Where $\phi(\mathbf{a}^k|\mathbf{a}^n, s, \sigma_j)$ represents the pdf value of state $\mathbf{a}^k$ based on the distribution of $\mathbf{a}_{t+1}$ given previous state $\mathbf{a}_t = \mathbf{a}^n$ and choices s and σ_j. Finally, for each grid point, update the associated value as the larger of the two found in Equation 2.10:

$$V_{g+1}^n = max(\{\hat{V}_{A,g+1}^n, \hat{V}_{B,g+1}^n\}) \tag{2.11}$$

This process was applied iteratively until the optimal policies stopped changing in response to one another.

Results

The equilibrium policy functions were identical for each player, and is shown below for the case where $\beta = 0.9, \mu = 0.02, \Sigma_{11} = \Sigma_{22} = 0.05,$ and $\Sigma_{12} = 0$. Black indicates the player will choose B, and White represents the use of action A. Note

that this is simply one equilibrium. In general, there may exist others, and it may be the case that obtained equilibria depend on initial value or policy functions.[8]

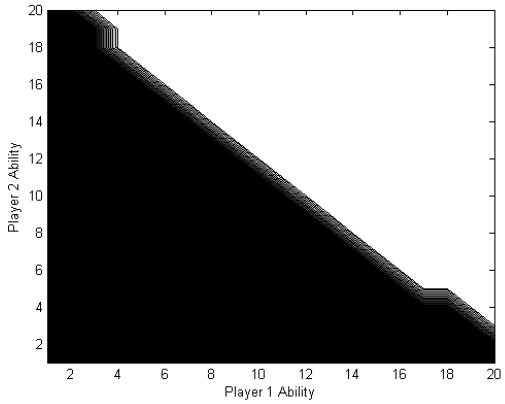


FIGURE 2. Example Game Equilibrium Policy

In this particular example, the policy function for each player was exactly the same. This symmetry might have been expected because the stage game is symmetric in so far as the payoff functions are the same.

Perhaps a better way to illustrate the implications of the equilibrium policies is to construct a direction field of the one step ahead conditional expectation of ability. That is, for any current set of abilities, what is the expected value of ability next period based on the policy functions. Such a graph is shown in the below figure:

[8]Various initial conditions were used to generate similar results for both games, indicating that the results may not be very sensitive. However, I cannot say conclusively that these equilibrium results are unique.

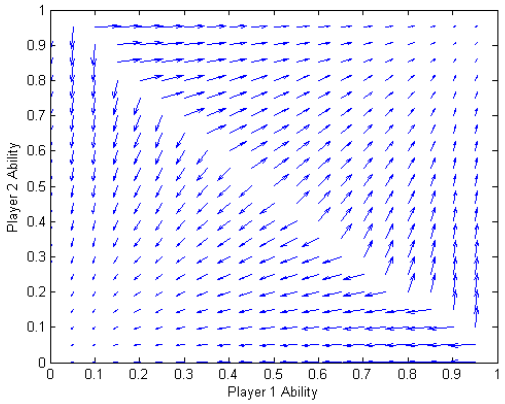


FIGURE 3. One-Step Ahead Expected Ability - Game 1

The implications of the policy set are now much clearer. From the above graph, it appears that there are two attracting states. Either both players end up specializing in A or both end up specializing in B. However, because ability is stochastic, there is the possibility that the actual path of ability might not be restricted to either of the attracting states . Thus, the equilibrium might be better described by simulating play following the equilibrium policy, and then looking at the distribution of ability.

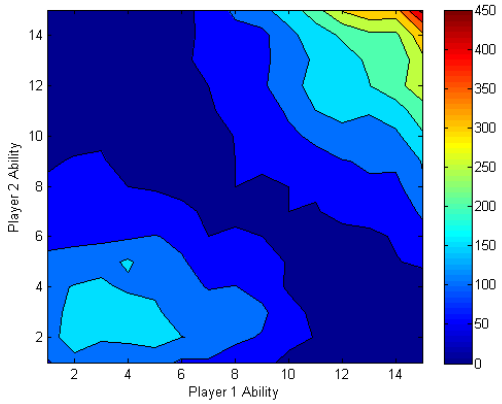


FIGURE 4. State Distribution in Equilibrium

The above distribution was generated from an initial state of $[0.1, 0.1]$, and shows the frequencies of states for 20,000 periods. This distribution shows what we might

have expected; that the system will spend most it's time near the all A or all B corners. While both these states have large attracting power, a series of large enough shocks can switch between attracting states.

Repeated Differentiated Duopoly with EBA

A Model of Differentiated Duopoly

In order to explore the choice of competition, Singh and Vives (1984) used the following model. Two firms compete with each other, each producing a differentiated good. The representative consumer maximizes $U(q_1, q_2) - \sum_{i=1}^{2} p_i q_i$, where $U(q_1, q_2) = \alpha_1 q_1 + \alpha_2 q_2 - (b_1 q_1^2 + b_2 q_2^2 + 2\gamma q_1 q_2)/2$. The first order conditions of the maximization problem lead to the following linear demand system:

$$\begin{aligned} p_1 &= \alpha_1 - b_1 q_1 - \gamma q_2 \\ p_2 &= \alpha_2 - b_2 q_2 - \gamma q_1 \end{aligned} \tag{2.12}$$

In a two-stage game, each firm first chooses whether to utilize price-contracts or quantity-contracts. After choosing contract types, the firms compete with each other, the outcome of which is contingent on the type of contracts chosen. If each firm chooses quantity-contracts, the standard Cournot equilibrium will result and each firm will gain the Cournot Profit, π^{QQ}, and if each chooses price-contract, the standard Bertrand equilibrium results and each firm gains Bertrand Profit π^{PP}. The only issue, then, is what happens if firms choose different types of contracts. Suppose firm i chooses price, and firm j chooses quantity. Then firm i chooses a price, p_i, to maximize profits taking q_j as given. Likewise, firm j would pick q_j to maximize its

profit taking p_i as given. In this case, firm i will earn a profit of π^{PQ}, and firm j will earn π^{QP}.

One of the main results presented in Singh and Vives (1984) is that these four profits can be ranked in both the substitution and complements cases. Specifically, in the case of substitutes $\pi^{QQ} > \pi^{QP} > \pi^{PP} > \pi^{PQ}$, and $\pi^{PP} > \pi^{PQ} > \pi^{QQ} > \pi^{QP}$ if the goods are complements. Thus, if firms can commit to a contract type, then it is a dominant strategy to choose the quantity (price) contract if the goods are substitutes (complements). Later research extended the differentiated duopoly model to include features such as asymmetric costs (Zanchettin 2006) and demand uncertainty (Klemperer and Meyer 1986) in a one shot game. The remainder of this section examines the implications of extending the differentiated duopoly model to incorporate the concepts of EBA which were discussed earlier.

The Infinitely Repeated Game with Price and Quantity Ability

Now suppose that the two firms repeatedly play this game, and the goods are substiutes. Each stage, firms have two available strategies, Price or Quantity contracts. After each stage, firms will gain ability in their chosen strategy, and lose ability in the strategy that was not chosen. The increases in ability are not impacted by the quantities produced or prices that are set, only by the choices of strategies themselves. Let $a_{it}^{Q(P)}$ denote player i's ability in utilizing quantity (price) contracts, and let $I(Q)$ be an indicator for player i choosing quantity contracts in period t. Ability will then accumulate as follows:

$$\begin{aligned} a_{it+1}^{Q} &= [a_{it}^{Q} + \mu(1 - a_{it}^{Q})]I(Q) + [a_{it}^{Q} - \mu(a_{it}^{Q}](1 - I(Q)) + \epsilon_{it+1} \\ a_{it}^{P} &= 1 - a_{it}^{Q} \end{aligned} \tag{2.13}$$

	Q	P
Q	$(\pi^{QQ}(a_{1t}, a_{2t}), \pi^{QQ}(a_{1t}, a_{2t}))$	$(\pi^{QP}(a_{1t}, a_{2t}), \pi^{PQ}(a_{1t}, a_{2t}))$
P	$(\pi^{PQ}(a_{1t}, a_{2t}), \pi^{QP}(a_{1t}, a_{2t}))$	$(\pi^{PP}(a_{1t}, a_{2t}), \pi^{PP}(a_{1t}, a_{2t}))$

TABLE 4. Duopoly Stage Game

Because I am again using the assumption that $a_{it}^P = 1 - a_{it}^Q$, I will define $a_{it} \equiv a_{it}^Q$ for ease of notation. Abilities affect firms' payoffs as follows. Marginal costs are constant each period, but depend on the current level of Quantity ability. This can be seen as a type of Learning by Doing effect, where the more a firm chooses the quantity contract, the lower their marginal costs become. Specifically, let marginal costs be represented by $c = k(1 - a_{it+1})^M$. Price ability doesn't affect production costs, but instead affects the demands which each firm faces:

$$\begin{aligned} p_1 &= \alpha_1 - b_1 q_1 - \gamma q_2 \\ p_2 &= \alpha_2 - b_2 q_2 - \gamma q_1 \end{aligned} \tag{2.14}$$

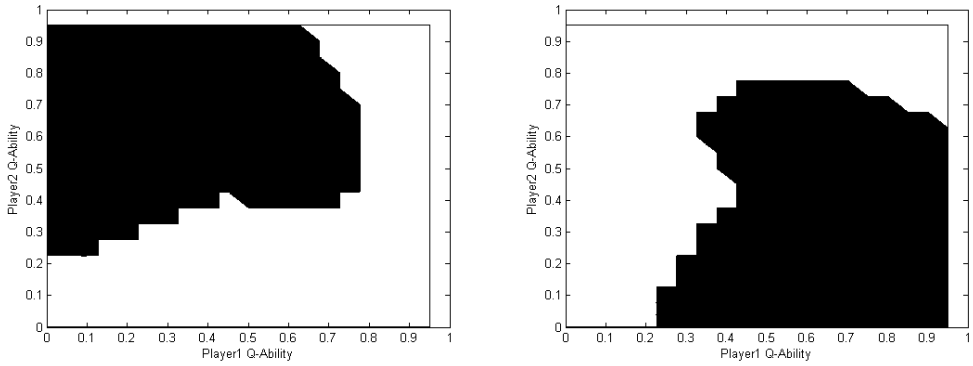
Where $\gamma = K(a_{1t+1})(a_{2t+1}) > 0$.

It is assumed that, because the actual quantities produced and prices set have no impact on future ability levels, after contracts are chosen, each firm chooses quantities and sets prices to maximize it's profit for that period. The problem for the firm is what contracts to choose over time in order to maximize its expected sum of discounted profits. Stated in the notation used previously, firms repeatedly play the following stage game: Notice that this game payoffs are nonlinear fuctions of ability levels, unlike those presented in Section V. Each firm then chooses a policy function, $\sigma_i(a_{1t}, a_{2t})$, which indicates the probability that firm i chooses the quantity contract in period t, in order to maximize the expected sum of discounted payoffs.

Results

The same techniques described in Section V were utilized to find the optimal policies for several values of the parameters K, k, and μ. Demand was always specified with $\alpha_1 = \alpha_2 = 15$ and $b_1 = b_2 = 4$. In order to make sure the results were consistent with the findings in Singh and Vives (1984), optimal policies were found for the case of constant marginal costs and constant γ. That is, it should be the case that if firms don't gain or lose ability, the optimal policies should lead both firms to always choose the quantity contracts when the goods are substitutes. Indeed, this is exactly what the optimal policies dictate. The optimal policies in this case dictated that firms always pick quantity.

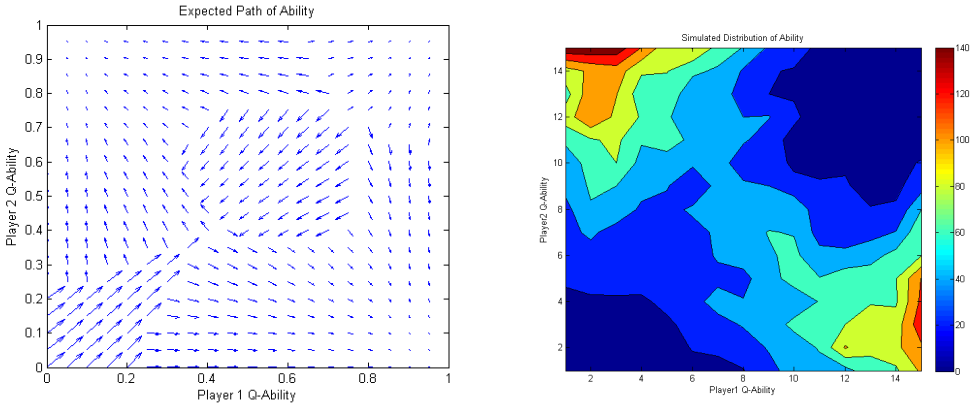
Next, I solved for equilibrium policies when firms could change their ability levels. In this first example, the parameters were set as follows: $K = 3, k = 1.25, M = 1, \mu = 0.02,$ and $\beta = .9$; that is $\gamma = 3(1 - a_{1t}^P)(1 - a_{2t}^P)$, and marginal costs for each firm are given by $c = 1.25(1 - a_{it}^Q)$. For Σ, the covariance of shocks was set at $\Sigma_{12} = 0$, and the variance was set as $\Sigma_{11} = \Sigma_{22} = .005$. The optimal policies for each firm were symmetric, and both are shown below. Again, White indicates the firms will choose Quantity, and Black indicates firms choosing Price:



(a) Player 1 Policy (b) Player 2 Policy

FIGURE 5. Policy Functions

In understanding the implications of these policies, it will be usefull to look at the plot of expected motion and the simulated distribution of ability induced by the equilibrium policies:



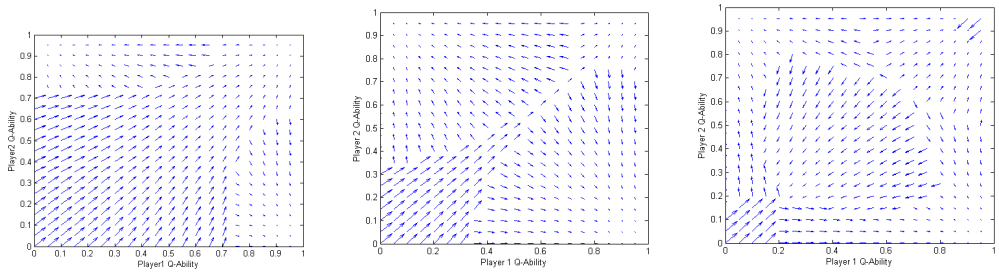
(a) Simulated Choices 1 (b) Simulated Choices 2

FIGURE 6. Ability Expected Path and Distribution

The above figures demonstrate why it is important to look at the distribution of abilities and not just the expected path of abilities. The graph of the expected path indicates that given an appropriate initial state, the two firms might end up both specializing in Quantity contracts. However, this apparent area of convergence

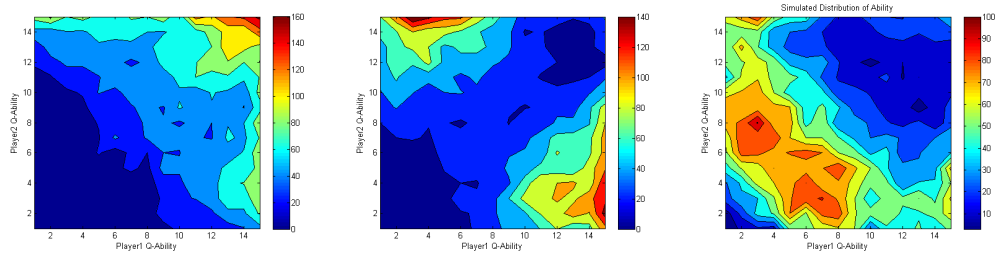
is quite small, and the stochastic nature of ability is such that the players actually spend most of their time in the upper left or bottom right portion of the state space. In these areas, one player will use price contracts, while the other uses quantity. Usually, a transition will occur because a large enough shock will push the players into the zone where they both utilize price contracts, which will eventually lead them both back into one of the two attracting areas.

Because this equilibrium behavior is sensititve to the parameter values, changes to equilibrium policies due to changes in parameter values were explored. First, I investigated what happens as the value K (which impacts the demand parameter γ) changes. The below figures show the expected path and distribution of ability induced by equilibrium policies as the parameter K changes from 1 to 5:



(a) $K = 1$ (b) $K = 2$ (c) $K = 5$

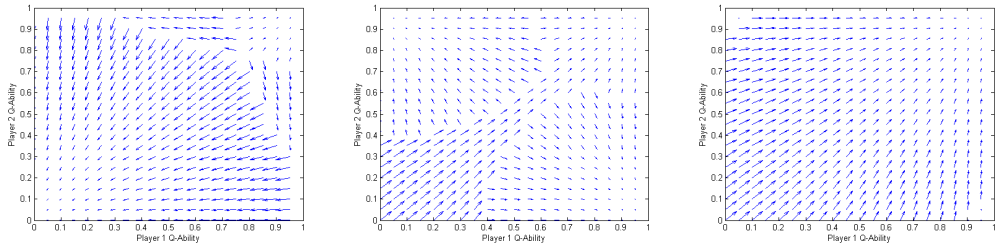
FIGURE 7. Ability Expected Path - Varying K



(a) $K = 1$ (b) $K = 2$ (c) $K = 5$

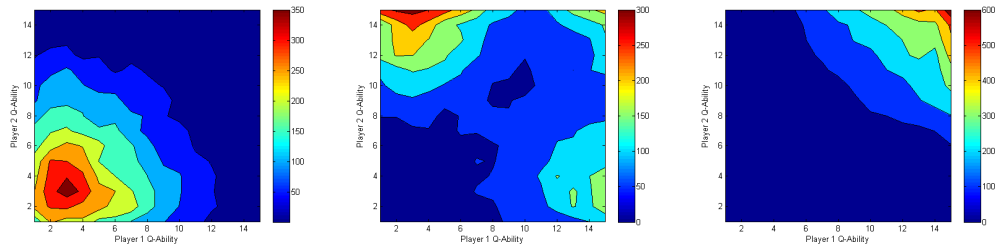
FIGURE 8. Ability Distribution - Varying K

Recall that the parameter K dictates the maximum that the demand link parameter γ can be. That is, $\gamma \in [0, K]$, depending on the ability levels of both players. Clearly, as K increases, the equilibrium policies of both firms have more and more area allocated to using the Price strategy, as might be expected. Next, consider how equilibrium policies change as the marginal cost parameter changes:



(a) $k = 0.5$ (b) $k = 2$ (c) $k = 4$

FIGURE 9. Ability Expected Path- Varying k

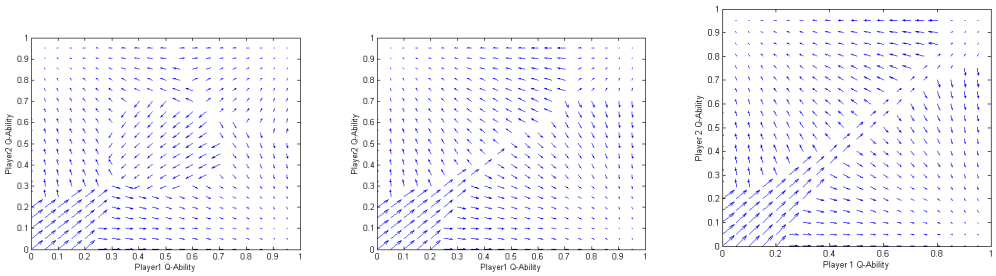


(a) $k = 0.5$ (b) $k = 2$ (c) $k = 4$

FIGURE 10. Ability Distribution - Varying k

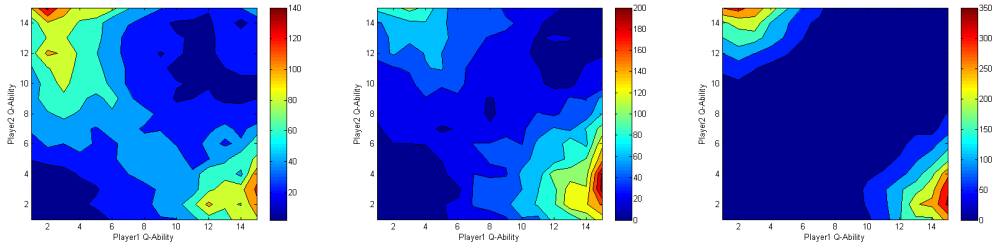
As is shown above, when marginal costs are small, the equilibrium policies have firms using price contracts throughout a majority of the state space. However, as the parameter increases, the area in which firms choose to use quantity contracts increases. When $k = 4$ the equilibrium policies have firms always choosing quantity; which was exactly the same as the original results of Singh and Vives with no learning.

We can also investigate what happens as the speed of learning parameter, μ, changes. The below figures show what happens as it changes from 0.01 to 0.06:



(a) $\mu = 0.01$ (b) $\mu = 0.03$ (c) $\mu = 0.06$

FIGURE 11. Ability Expected Path - Varying μ

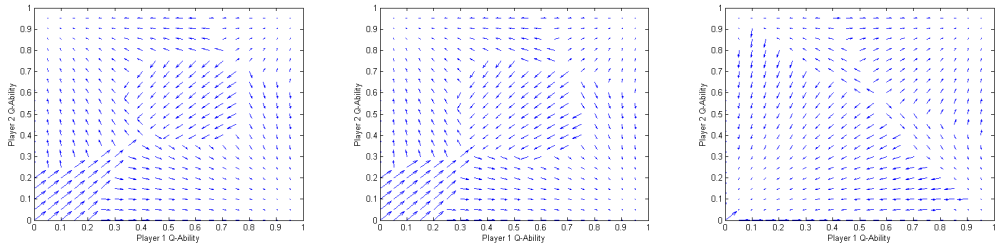


(a) $\mu = 0.01$ (b) $\mu = 0.03$ (c) $\mu = 0.06$

FIGURE 12. Ability Distribution - Varying μ

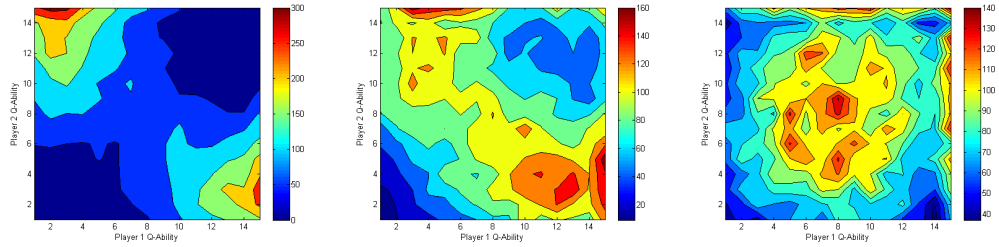
As the speed of learning increases, the area where players both choose to utilize price contracts simultaneously vanishes, and this increases the attraction size of the upper left and lower right corners. Thus, as μ reaches 0.06, most of the observations will display players specializing in opposite strategies.

Lastly, we can look at what happens as the variance of the ability shocks increases.



(a) Σ_{11}= 0.005 (b) Σ_{11} = 0.01 (c) Σ_{11} = 0.05

FIGURE 13. Ability Expected Path - Varying Σ_{11}



(a) Σ_{11} = 0.005 (b) Σ_{11} = 0.01 (c) Σ_{11} = 0.05

FIGURE 14. Ability Distribution - Varying Σ_{11}

As the variance increases, it changes the equilibrium policies substantially and eliminates the upper left and lower right corners as attracting states. It appears to instead push towards the both price and both quantity corners. However, since the variance has increased, the actual distribution of states is much more disperse, and the implications of the policy harder to see by looking at the distribution.

In summary, the parameters governing the demand link and marginal costs, K and k, can shift the equilibrium distribution from all quantity, to all price, as they increase or decrease appropriately. The learning parameter, μ, does not seem to change the location of the equilibrium distribution, but instead decreases the transition time between attracting states and tightens the distribution. The is the opposite effect of increasing the variance, which increases the dispersion of equilibrium

distributions. One concern to be drawn from this exercise is that any econometric application of this particular model might be plagued with identification issues. That is, because increases in the demand link parameter have similar effects as decreasing the marginal cost parameter, it might be hard to separate those effects.

Discussion and Extensions

While the previous sections presented a thorough introduction on what EBA is and how it can be used to introduce the concept of "practice" into current learning and game theory models, there is clearly an abundance of future work that needs to be done in this area. From the presented framework to the equilibrium focus discussed earlier, further exploration is needed in all aspects of EBA. This section briefly discusses the possibilities for refining and extending EBA in future research.

First of all, the proposed mechanisms for ability evolution are not entirely complete in terms of fully representing the real world processes being modeled. For example, while allowed for in the general specification of ability evolution in Equation (**??**), opponent ability did not play any role in the examples of ability evolution. In the real world, however, it is often the case that opponent ability matters a great deal to your own ability development. That is, you may develop ability much quicker playing a more skilled opponent as opposed to playing a very unskilled opponent. In a related vein, it may also be possible to allow ability to be action profile specific; indicating how skilled you are at playing strategy s specifically when your opponent plays strategy s'.

Within the proposed framework, the equilibrium concepts and outcomes also need to be further analyzed. Will Markov equilibrium be unique or will it depend on the initial conditions used (i.e. the intitial policies)? How sensitive are results

to different ability evolution specifications? Is this equilibrium concept the most compelling? All of these questions should be addressed, and this paper has not examined them in any real depth. While there do exist various equilibrium existence results emanating from the stochastic game literature (Chakrabarti 1999, Curat 1996, Horst 2005, Dutta and Sundaram 1992), the specific nature and focus of EBA necessitates the development of EBA specific equilibrium existence results just as how specific existence results have been developed for CPR and bioeconomic games (Martín-Herrán and Rincón-Zapatero 2005, Sorger 1996).

While the model presented in Section VI was insightful, it is most definitely not the only possible extension to the standard differentiated duopoly model. In Section VI, quantity ability was interpretated as a kind of reduced-form learning by doing effect, and price ability could be interpreted as a type of advertising skill. One extension to this would be to explicitly model these effects. That is, allow marginal costs to be a function of *cumulative* output, and perhaps allow the history of chosen prices to affect market demand. In general, EBA represents a possible tool for modeling endogenous rivalry and this should also be considered as an area for future research.

Lastly, there exists vast possibilities for extensions of the EBA framework into other established fields within economics. One possible extension would be to incorporate EBA into an evolutionary framework. For example, using a profile-specific ability specification, one could model how agents become more and more experienced in their interactions with the dominating population type; of course this might also lead to a deterioration in the agents' ability to deal with the minority population type, offering that population a chance to emerge again. This particular extension is carried out in Chapter III.

Conclusions

Learning is an important feature of who we are as human beings. And while learning in economics has taken great strides in recent years, it has also been doing so mainly on one foot. Current learning models fall short in depicting one of the most critical parts of the learning process: Practice. Thus, it is only prudent that a formal, basic framework be developed to handle this learning process in order for economic models to fully incorporate learning.

The framework presented in this paper for modeling Experienced-Based Ability is both simple in terms of what is being added to standard repeated games, and rich in terms of its possibilities. By simply adding an ability evolution function and allowing payoffs to depend on abilities, repeated games become much more dynamic. These models generate an endogenously changing stage game, including both the payoff amounts and the very *structure* of the stage game itself. These games become increasingly difficult to analyze, but result in very rich player behavior dynamics.

By assuming players solve a dynamic programming problem subject to policies of other players, I demonstrated how a Markov equilibrium could be approximated. The equilibrium policies in turn generate very interesting player behavior dynamics which can be simulated and analyzed. Depending on the game studied, players may end up specializing in a particular strategy or continually rotating between being skilled at one, then at the other. These outcomes may be dependent on initial ability levels, and they may be independent of such initial conditions.

It should now be clear that no discussion of learning in games would be complete without a framework like EBA to address practice and experience. These concepts seem so natural, and it is difficult to see why they are currently absent from the game theory literature. Perhaps this innate naturalness is responsible for the abundance

of extensions and applications which flow out of the EBA framework; an abundance which, hopefully, will not go unexplored.

CHAPTER III

EXPERIENCED EVOLUTION

Introduction

Since its conception several decades ago, evolutionary game theory has become an increasingly important theoretical tool within economics; helping to refine equilibrium selection and promising great application possibilities both within and outside the field of economics. For example, not surprisingly, the field of biology has adopted evolutionary game theory as one of its most important tools in explaining the existence of various behaviors within populations. However, despite the use of ever more complicated models, most empirical work based on evolutionary game models, both in the field of biology and elsewhere, fails to go beyond qualitative matching of models and data. This does not have to be the case, and the current paper represents one solution for overcoming this problem.

Evolution is concerned with the relationship between individuals and their environment; both the effect of environment on individuals and the effect of individuals on the environment. Standard evolutionary models typically define the environment as the current make up of the population; i.e. how many of each type current exist within the population. Thus, while the evolution of the population is examined, little attention is payed to how the individual types within the population might evolve. That is, they leave out the possibility that, over time, these individual populations of different types might adapt and get better at dealing with prominent types within the population; conversely it also ignores the possibility that more prominent types might forget how to deal with less prominent types.

This paper proposes extending the model of Experienced-Based Ability (EBA) into an evolutionary framework in order to tell the above story. EBA games are a specific type of dynamic game wherein a player's payoffs when in a certain situation (action profile) can change over time the more they find themselves in (or out) of that same situation. The extension to standard evolutionary models of incorporating EBA tells a simple story. Each type within the population has a certain ability at dealing with the other types. If type A is prominent, all other types will begin to improve at dealing with the A types. Likewise, all types might forget how to deal with less prominent types. While the intuitive story is still quite simple, the model is capable of displaying complicated dynamics.

The current chapter has several goals. First, I extend the model of EBA into an evolutionary setting. Doing so creates a model, which I call Experienced Evolution, that allows for rich dynamics, even in a 2 strategy environment. Secondly, I estimate posterior distributions of model parameters based only off of population share data. These parameters include shock variances, learning speeds, and the parameters of the underlying stage game. Together, I hope to make a strong case for this model as a tool for applied game theory, especially as it is used in biology. The estimation technique, which utilizes a particle filter in an M-H algorithm, is actually rather general and not specific to my presented model. Thus, a broader goal is to help demonstrate that advanced, robust empiric work can and should be done utilizing evolutionary game models.

The remainder of the chapter is organized as follows. Section II presents a nontechnical discussion of the progression of evolutionary models and their shortcomings. Section III describes reviews the EBA model and presents the experienced evolution model. Sections IV and V briefly explore theoretical results

in the cases of strategy specific and profile specific ability respectively and provide several simulation examples. Section VI presents the econometric technique while section VII provides estimation examples. Finally, section VIII concludes.

Evolution of Evolutionary Models

From a theoretical standpoint, evolutionary games have been used to explain myriad observed behaviors in biological settings. These settings include animal behavior like the mating habits of lizards (Sinervo and Lively 1996, Sinervo and Zamudlo 2000, Sinervo, Svensson and Comendant 2000), the dynamics of microorganisms like bacteria (Vulic and Kolter 2001) and cancer cell growth (Gatenby and Vincent 2003), and even an explanation of why autumn brings about vibrant displays of changing colors (Archetti 2000). The evolutionary games utilized range from basic prisoner's dilemma and RPS games to ever more complicated models incorporating adaptive dynamics and coevolutionary models (Hofbauer and Sigmund 2003, Nowak and Sigmund 2004,Perc and Szolnoki 2009). However, while it has become an invaluable tool for explaining the existence of basic behavioral patterns and phenomena, there has been a surprising lack of robust econometrics. That is, most research observes a behavioral pattern and presents a model that is in qualitative agreement; e.g. appropriate steady states. While much of this work has been important in theoretically understanding various population behaviors, the technique itself may be unsatisfactory if there are parameters of interest in the model or there are competing models which need to be compared. The remainder of this section presents a nontechnical discussion of progression of evolutionary models and their econometric issues.

The basic evolutionary game model works as follows. A population consists of multiple phenotypes, each genetically hardwired to utilize a certain strategy. Each period, members of the population are randomly matched and an underlying game is played. The payoffs to each individual represent the individual's fitness, and the average fitness of each type determines how well that particular strategy will propagate within the population. Thus, the average fitness of each phenotype is determined by the current makeup of the population, which in turn develops based on the average fitnesses of each type. This creates a dynamic system. The original equilibrium concept was that of an Evolutionarily Stable Strategy (ESS), proposed by John Maynard Smith (Maynard Smith 1974), which requires that the current population makeup be resistant to mutant invaders.

While initially compelling, the concept of ESS is quite rigid and inconsistent with many observations in biological systems. Firstly, an ESS with more than one type existing in the population requires that the average fitness levels of all present types must be equal. Thus, if a researcher observes a population with multiple types present in a consistent ratio, they must demonstrate that the fitness of each type is equal in order to stay consistent with the evolutionary model. While studies such as those done in Ryan,Pease, and Morris (1992) attempt to accomplish this, it is problematic. The issue lies in the fact that researchers must attempt to quantitatively measure fitness, and this is done by collecting data on something which is supposed to represent fitness. But fitness is an abstract concept and hard to define in most settings, and is therefore most likely unobserved (Arganiski and Broom 2012). Even if fitness can be measured, the ESS as an equilibrium concept is still problematic.

Many populations will often experience consistent and predictable cycles in their makeup. Unfortunately, the concept of ESS did not originally allow for the possibility

of cycles; only, at best, a constant proportion mixed population. This issue was known early on within the evolutionary game theory literature. For example, the classic example of Rock-Paper-Scissors demonstrated that approach to a mixed-steady state can be cyclical, or if the parameters are right, there can even exist limit cycles within the most basic evolutionary model. For this reason, Taylor and Jonker (1978) and Zeeman (1980) expanded equilibrium concepts to include attractors which they distinguished from ESS. A classic application of the RPS game and it's capability to produce cycles is the work done investigating the mating habits of lizards. Sinervo and Lively (1996) observe that the number of orange, blue, and yellow throated male lizards cycled in terms of which one was most prominent. Using the number of females available within each lizard's territory to approximate fitness, the authors constructed an underlying stage game which predicted similar cyclical patterns.

One problem with the above approach is that data on fitness proxies is measured over time while the stage game for the model is static. In order to approximate the payoffs of the stage game, data for fitness proxies essentially has to be averaged. Thus, the researcher has to force dynamic data to be static. This relates to another issue with classic evolutionary models. Even though cyclical patterns can emerge once more than 2 strategies are considered, there are still restrictions on the types of data which are consistent with the model. This stems from the fact that in any standard evolutionary model, the direction of the population makeup is a function of the current population makeup. For example, if the current mix is 20% blue, 30% yellow, and 50% red throats, the model predicts precisely how these percentages will change in the immediate future. Thus, if the data observed the same mix in time periods t and t', the model would predict the same subsequent movements in relative populations in each of these time periods. So as soon as the data had different

movements for the similar population mixes, it would be considered inconsistent. One possible solution to this issue might be the inclusion of stochasticity in the model.

Stochasticity has been present in evolutionary models from the start. The concept of ESS was that a population was stable if it was immune to a single random shock. However, other than one time shocks during equilibrium periods, stochasticity was absent from evolutionary game models. Foster and Young (1990) and Kandori, Mailath, and Rob (1993) discuss the importance of including stochasticity and examine models which are continually hit by small random shocks. Both papers proposed a new equilibrium concept of stochastically stable sets. Basically, these are the set of points which, in the long run and under perpetual shocks of variance σ^2, have the property that the probability of the system being in that set goes to 1 as σ^2 goes to 0. These sets can then be used to select between multiple possible steady states. Foster and Young note that if the value of σ^2 is known, the stochastically stable set could be defined as the smallest set of states that is 99% probable.

While the inclusion of stochasticity can allow a broader set of observed data to be consistent with a model, the previous inconsistencies are simply explained away as random shocks and so these movements would be inherently unpredictable. This may be insufficient, especially if the researcher desires a model which generates more complicated and simultaneously predictable dynamics. The field of biology has begun to turn to more advanced models such as adaptive dynamics which allow for continuous strategy spaces (Nowak and Sigmund 2004) and coevolutionary models which allow the rules of interaction and reproduction to change throughout time (Perc and Szolnoki 2009). Unfortunately, the uses of these more advanced models have, for the most part, been for theoretical explanations and not robust empirical work.

The remainder of the paper presents and demonstrates the experienced evolution model. As will become clear in later sections, the full model overcomes several of the issues presented above. First, the shock free model can generate complex dynamics, which include attractors and limit cycles. Most impressively, it can even do so within a 2x2 setting. Secondly, the full model presented incorporates continual shocks to the system. These shocks all share a common variance, which can be econometrically estimated; an important parameter to estimate if one wants to approximate the stochastically stable set as suggested by Foster and Young (1990). Other parameters of interest can also be estimated using the techniques presented, including learning speed parameters for different types and the parameters of the stage game itself. The model allows for a continually changing stage game, which implies it allows proxy measures of fitness to change over time. But more importantly, the parameters of the stage game can be estimated using only data on population shares. As such, the model allows a researcher to recover estimates of unobserved fitness. All together, experienced evolution overcomes most of the above issues with previous evolutionary models and yet still allows the researcher to tell an intuitively simple story.

Model

EBA Games

As discussed extensively in the previous chapter, games involving Experienced-Based Ability (EBA games) are a specific type of dynamic game wherein a player's use of a strategy today increases the payoffs of using that same strategy in the future. More specifically, every player has an ability level in each of their available strategies. As they use a strategy, they gain ability in using it (and may simultaneously lose ability in unused strategies). Higher ability in a strategy generates higher payoffs if

that strategy is utilized. The players involved in the game must also be concerned with the ability levels of their opponents because opponent ability level can also affect their payoffs. While the model presented below appears somewhat complicated, the concept is one we are all familiar with: In order to effectively use a strategy, I need to practice it and gain experience. Using a strategy I have no experience with may result in undesirable outcomes.

This paper will address two types of ability: Strategy Specific and Profile Specific ability. In setting up the basic EBA model, I begin with Strategy Specific ability. Consider a game played between two players. In an EBA-Symmetric game, each player shares the same strategy set, S, which consists of M strategies $S = \{s_1, s_2, ..., s_M\}$. In each time period, each player has a vector of ability levels which describes how skilled the players are at each strategy. Specifically, let $\mathbf{a}_{it} = (a_{it}^1, a_{it}^2, ..., a_{it}^M)$, where a_{it}^m denotes player i's ability at using strategy m in time period t. Once again, I make the restriction that $a_{it}^m \in [0, 1]$. The path that ability takes over time is determined by the previous ability level and the chosen action, i.e. $a_{it}^m = \alpha_m(a_{it}^m, I(m))$, where $I(m) = 1$ if player i chose strategy m in time period t.[1] The remainder of this chapter assumes the following specification:

$$a_{it+1}^m = I(m)[a_{it}^m + \mu(\sqrt{a_{it}^m} - a_{it}^m)] + (1 - I(m))[a_{it}^m - \mu(a_{it}^m - (1 - \sqrt{1 - a_{it}^m}))] \quad (3.1)$$

Payoffs in each period are not only dependent on chosen strategies, but also on the abilities of each player. In general, it may be the case that payoffs can depend on the entire vector of ability, but the remainder of the paper focuses on situations where payoffs depend only on the chosen strategies, and players' abilities in those chosen

[1] In general, $\alpha_m(\cdot)$ could be a function of other variables, e.g. time itself

	A	B
A	$f^{AA}(a^1_{1t}, a^1_{2t})$	$f^{AB}(a^1_{1t}, a^2_{2t})$
B	$f^{BA}(a^2_{1t}, a^1_{2t})$	$f^{BB}(a^2_{1t}, a^2_{2t})$

TABLE 5. General 2x2 Stage Game

strategies. The rest of the paper focuses on 2x2 EBA-symmetric games, wherein each player shares the same strategy set, $S = \{A, B\}$ and payoff functions. In this case, we only need to list the payoff functions for Player 1:

It is important to note that the symmetry arises from the commonality of strategy set and payoff function. However, in any given time period, the stage game itself will only be symmetric if both players share the same ability level.

A more intricate specification is that of Profile-Specific Ability. In this case, each player has a specific ability level at using an available strategy against a specific opponent strategy. Put more simply, each player has a specific ability for each possible action profile. Thus, in the 2x2 case, there would be 4 ability levels to keep track of for each player, each of which evolves over time in a manner similar to that described above. What is interesting about profile-specific ability games is that, in contrast to the strategy-specific case, a player's choice of strategy not only affects the development of their own skill, but also the skill of their opponent. In order to restrict my opponent's ability development, I can play a strategy infrequently so that my opponent doesn't get much practice dealing with it. Unfortunately, as in all economic situations, a tradeoff exists in that by using a strategy infrequently, I may also be hindering my own ability at using that strategy.

Evolution

One of the main goals of this paper is to extend the EBA model into an evolutionary framework. This is done in the following way, similar to standard evolutionary dynamics: First, I assume that there is an infinitely large population that consists of a discrete number of player types (phenotypes). For example, the majority of this paper focuses on 2x2 games, such that there are two types of players, A-types and B-types. A-types only ever use strategy A, and B-types only ever use strategy B.

Each period, players are randomly matched with each other and play one stage of an EBA game. The outcome of the game determines players' fitness levels, which in turn determine the relative quantity of offspring each has. These offspring inherit two things from their parents. First, as usual, the offspring are hardwired to be the same type as their parent. Second, the offspring are endowed with the average ability level within their parent's type. That is, in each period, player's have an ability level at their chosen strategy. The randomly matched interaction will give that player some experience which is embodied by appropriate increases/decreases in ability levels. There would then be a new average ability level within each type's community. It is this average which is passed down to offspring of each type's community.

Model SetUp - Strategy Specific Ability

This model can be formulated as a discrete dynamic system. For simplicity, assume that there are only two types, A-types and B-types. Let λ_t denote the percentage of the population that are A-types. In the strategy-specific ability case, let a_t denote the ability level of A-types at using strategy A in time period t, and let b_t denote the ability level of B-types at using strategy B. Furthermore, assume that

	A	B
A	$f^{AA}(a_t)$	$f^{AB}(a_t, b_t)$
B	$f^{BA}(a_t, b_t)$	$f^{BB}(b_t)$

TABLE 6. Strategy Specific Stage Game

payoffs for an interaction between two A-types is independent of b_t, and likewise for interactions between two B-types.[2] The stage game in each period is then: Average fitness of each type can then be calculated as:

$$\begin{aligned} AvgFitA_t =& \lambda_t f^{AA}(a_t) + (1-\lambda_t) f^{AB}(a_t, b_t) \\ AvgFitB_t =& \lambda_t f^{BA}(a_t, b_t) + (1-\lambda_t) f^{BB}(b_t) \end{aligned} \tag{3.2}$$

Percentage shares of the population are determined using a discrete replicator dynamic:

$$\lambda_{t+1} = \lambda_t \frac{AvgFitA_t(\lambda_t, a_t, b_t)}{\lambda_t AvgFitA_t(\lambda_t, a_t, b_t) + (1-\lambda_t) AvgFitB_t(\lambda_t, a_t, b_t)} \tag{3.3}$$

Lastly, since A-types only play A, and B-types only play B, ability evolves according to:

$$\begin{aligned} a_{t+1} =& a_t + \mu(\sqrt{a_t} - a_t) \\ b_{t+1} =& b_t + \mu(\sqrt{b_t} - b_t) \end{aligned} \tag{3.4}$$

[2]If one wants to consider externalities across types, then the outcomes of an AA interaction could be allowed to depend on the ability levels of B-types. A specification such as that would not alter any results of this paper

	A	B
A	$f^{AA}(a_t^A)$	$f^{AB}(a_t^B, b_t^A)$
B	$f^{BA}(a_t^B, b_t^A)$	$f^{BB}(b_t^B)$

TABLE 7. Profile Specific Stage Game

	A	B
A	$1 + Ca_t^A$	$1 + Da_t^B + Eb_t^A$
B	$1 + Fa_t^B + Gb_t^A$	$1 + b_t^B$

TABLE 8. Linear Payoff Stage Game

Model Setup - Profile Specific Ability

Now consider the case where ability is profile specific. Since A-types only ever play A, it will only be necessary to track two ability levels for A-types, and likewise for B-types. Let a_t^A denote an A-types ability when matched against another A-type, and a_t^B denote an A-types ability when matched against a B-type. Similarly, b_t^A represents the B-types' ability when facing an A-type and b_t^B represents B-types' ability at facing other B-types. The stage game being played each period is then: Where the payoff functions are assumed to satisfy $\lim_{a\to 1, b\to 1} f^{s,s'}(a,b) = M^{s,s'} \in \mathbb{R}$. The remainder of the paper focuses on linear payoff functions of the following form:

The scalar addition term is necessary because the researcher may want the coefficients E and G to be negative. This would imply that as Player 2 gets better at facing opponent strategy A, the payoffs for Player 1 of using strategy A decrease. The model is concerned with the sizes of relative payoffs, and so I restrict all payoffs to be positive. In order to garauntee positive payoffs, the restriction is needed that $E \geq -1$ and likewise $G \geq -1$.

The population share of A-types, λ_t, progresses according to Equation (**??**) as it did in the strategy-specific ability case. However, ability now evolves in a slightly different manner. For simplicity, assume that ability is determined in a Use It or Lose It fashion. That is, any gain in one ability will cause a loss in the other. Specifically, now let $a_t \equiv a_t^A = 1 - a_t^B$, and similarly let $b_t \equiv b_t^B = 1 - b_t^A$. Now ability evolves according to:

$$\begin{aligned} a_{t+1} =& \lambda_t(a_t + \mu(\sqrt{a_t} - a_t)) + (1 - \lambda_t)(a_t - \mu(a_t - (1 - \sqrt{1 - a_t}))) \\ b_{t+1} =& (1 - \lambda_t)(b_t + \mu(\sqrt{b_t} - b_t)) + \lambda_t(b_t - \mu(b_t - (1 - \sqrt{1 - b_t}))) \end{aligned} \tag{3.5}$$

Strategy Specific Ability

Recall that in the case of strategy specific ability, the evolutionary process can be described by the below discrete dynamic system:

$$\begin{aligned} \lambda_{t+1} =& \lambda_t \frac{AvgFitA_t(\lambda_t, a_t, b_t)}{\lambda_t AvgFitA_t(\lambda_t, a_t, b_t) + (1 - \lambda_t)AvgFitB_t(\lambda_t, a_t, b_t)} \\ a_{t+1} =& a_t + \mu(\sqrt{a_t} - a_t) \\ b_{t+1} =& b_t + \mu(\sqrt{b_t} - b_t) \end{aligned} \tag{3.6}$$

Clearly, under the replicator dynamic specified, if there exists any amount of B or A type players at time $t = 0$, then there will always exist at least some of this type in the population for all $t > 0$. This implies that any steady state of the system, other than an initially degenerate state, must have $a^* = b^* = 1$. That is, as $t \to \infty$, all remaining members of the population become experts at their type's strategy. Thus, so long as $\lim_{a \to 1, b \to 1} f^{s,s'}(a, b)$ exists, the analysis of steady states simplifies to examining the below game in the usual fashion:

	A	B
A	$f^{AA}(1,1)$	$f^{AB}(1,1)$
B	$f^{BA}(1,1)$	$f^{BB}(1,1)$

TABLE 9. Limit Stage Game

	A	B
A	$1+2a_t$	$1+2a_t-b_t$
B	$1+3b_t-0.5a_t$	$1+2b_t$

TABLE 10. Monotonicity Stage Game

So what did adding ability bring to the table? At first, it may seem like nothing more than added complication. However, even though finding the steady states of the game boils down to the usual evolutionary analysis, the addition of ability allows for non-monotonic adjustment *to* the steady state. That is, in the usual evolutionary setting, the path of λ_t is necessarily monotonic; λ_t is either everywhere increasing, or everywhere decreasing. This fact, while rather obvious under a continuous RD, is not necessarily straightforward in the discrete case and I could find no previous proof of monotonicity. The concern, in the discrete case, is that on approach to a stable steady state, λ_t might overshoot the steady state slightly, or even continually oscilate above and below the steady state. However, the discrete RD is such that this can never occur. A proof of this monotonicity is provided in the appendix.

On the other hand, no proof is required to show that the evolutionary EBA game allows non-monotonic adjustment; a simple simulation suffices to show this. For example, the below figure shows a simulated path of λ_t where initial conditions were set as $a_0 = 0.3$, $b_0 = 0.1$, $\lambda_0 = 0.3$, learning speeds $\mu_1 = \mu_2 = 0.25$, and the underlying stage game was:

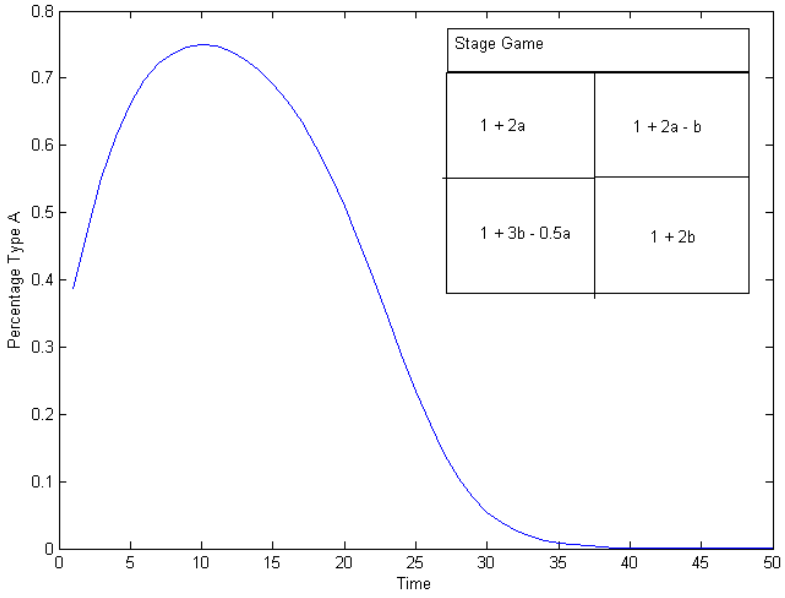


FIGURE 15. Non-Monotonic Adjustment Example

Profile Specific Ability

While the inclusion of strategy-specific ability in an evolutionary model generated a richer set of dynamics on approach to a steady state, the possibilities for steady states themselves remained unchanged from standard 2x2 evolutionary games. The incorporation of profile-specific ability, on the other hand, is not restricted by the usual steady state analysis; even in simple 2x2 symmetric games. Recall that the evolutionary process in this case is described by the below discrete dynamic system:

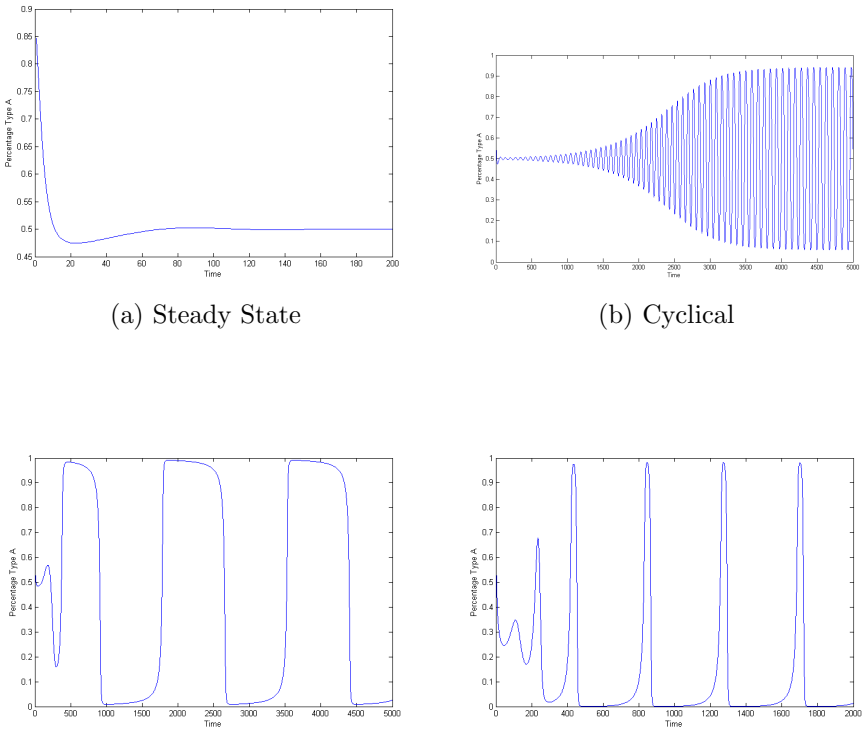
$$\begin{aligned}
\lambda_{t+1} =& \lambda_t \frac{AvgFitA_t(\lambda_t, a_t, b_t)}{\lambda_t AvgFitA_t(\lambda_t, a_t, b_t) + (1-\lambda_t) AvgFitB_t(\lambda_t, a_t, b_t)} \\
a_{t+1} =& \lambda_t(a_t + \mu(\sqrt{a_t} - a_t)) + (1-\lambda_t)(a_t - \mu(a_t - (1 - \sqrt{1-a_t}))) \\
b_{t+1} =& (1-\lambda_t)(b_t + \mu(\sqrt{b_t} - b_t)) + \lambda_t(b_t - \mu(b_t - (1 - \sqrt{1-b_t})))
\end{aligned} \tag{3.7}$$

Where $a_t(b_t)$ represents an A-type's (B-type's) ability when matched against an A-type (B-type), and recall that I am using the assumption $a_t \equiv a_t^A = 1 - a_t^B$; that is, any gain in experience against A-types will simultaneously represent a loss in ability against B-types. The stories which can be told in this setting are much richer. If the population consists of mostly B-types, then the dominating B-types will get better and better at playing against other B's. However, the few A-types remaining in the population will also get better at dealing with the B-types. Moreover, the B-types will become very bad at dealing with A-types. This effect may be enough to eventually help the A-types gain ground in the population. If they do, however, they may not be able fully overtake because as their share grows, B-types will eventually start to get better at dealing with these new upstart A's. Exactly how the population then depends not only on the stage-game performances of each type, but also on how quickly each type adapts to it's changing environment.

This storyline is one which is, unfortunately, absent from the standard evolutionary game model. Evolution, at its core, is about the relationship between individuals and their environment; how the environment affects the individual, how the individual affects the environment, and how the individual adapts to the environment. All of these elements are present in the above model, and this was done so by simply expanding on what the environment actually is. Now, the environment consists not only of how many of each type are in the population, but how skilled these types are. As the below examples show, myriad possibilities become available in the same 2x2 setting.

The four figures below show the path of λ_t for various sets of model parameters, which are shown in the table below. In each, the initial conditions were the same, $\lambda_0 = .7$, $a_0 = .2$, and $b_0 = .2$ The first model simply converges to a steady state. The

second exhibits cyclical behavior. The third exhibits regimes wherein a single type is dominant in the population for a period of time, then becomes virtually extinct. The last demonstrates a situation where B-types dominate the population, until A-types have a short burst, but the B types quickly overcome this burst and once again assume their dominance.



(a) Steady State (b) Cyclical

(c) Regimes (d) Pest Control

FIGURE 16. Example Games

Steady State

$\mu = .10$	A	B
A	$1 + 2a_t$	$1 + 2(1 - a_t) - (1 - b_t)$
B	$1 + 3(1 - b_t) - .5(1 - a_t)$	$1 + 2b_t$

Cyclical

$\mu = .10$	A	B
A	$1 + 1.3a_t$	$1 + 1.9(1 - a_t) - .76(1 - b_t)$
B	$1 + 1.9(1 - b_t) - .76(1 - a_t)$	$1 + 1.3b_t$

Regimes

$\mu = .11$	A	B
A	$1 + 2a_t$	$1 + 2(1 - a_t) - .5(1 - b_t)$
B	$1 + 2(1 - b_t) - .5(1 - a_t)$	$1 + 2b_t$

Pest Control

$\mu = .11$	A	B
A	$1 + a_t$	$1 + 2.05(1 - a_t) - (1 - b_t)$
B	$1 + 2(1 - b_t) - .8(1 - a_t)$	$1 + 2b_t$

TABLE 11. Example Evolution Games

Differences in Learning Speeds

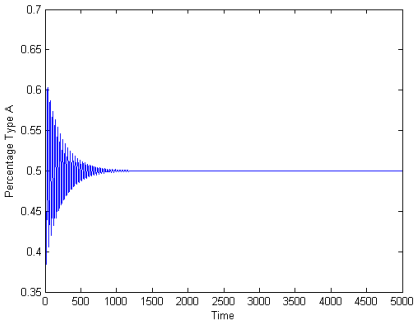
In the above examples, imbalances in the payoff functions were responsible for all of the varying dynamics. Even in the standard evolutionary game model, this has been the prevailing explanation for why population shares fluctuate. However, as the below examples demonstrate, allowing the learning speed parameter, μ, to vary between types can also cause changes in dynamics. If μ is different, the dynamic system becomes:

$$\begin{aligned}
\lambda_{t+1} =& \lambda_t \frac{AvgFitA_t(\lambda_t, a_t, b_t)}{\lambda_t AvgFitA_t(\lambda_t, a_t, b_t) + (1 - \lambda_t) AvgFitB_t(\lambda_t, a_t, b_t)} \\
a_{t+1} =& \lambda_t(a_t + \mu_A(\sqrt{a_t} - a_t)) + (1 - \lambda_t)(a_t - \mu_A(a_t - (1 - \sqrt{1 - a_t}))) \\
b_{t+1} =& (1 - \lambda_t)(b_t + \mu_B(\sqrt{b_t} - b_t)) + \lambda_t(b_t - \mu_B(b_t - (1 - \sqrt{1 - b_t})))
\end{aligned} \tag{3.8}$$

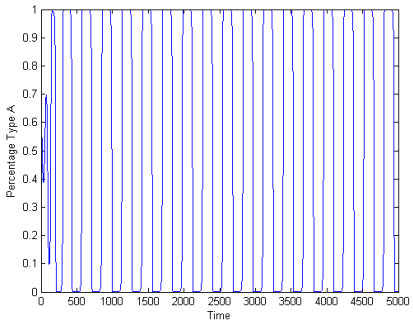
Using the following stage game, the path of λ_t is shown below for various learning speeds μ_A and μ_B:

	A	B
A	$1 + a_t$	$1 + 1.4(1 - a_t) - .9(1 - b_t)$
B	$1 + 1.4(1 - b_t) - .9(1 - a_t)$	$1 + 2b_t$

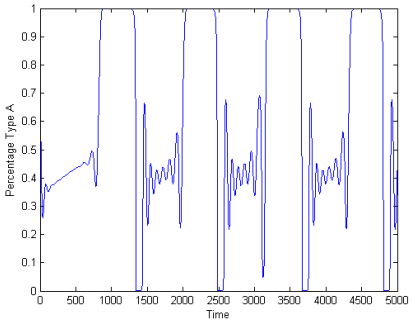
TABLE 12. Stage Game - Changing Learning Speeds



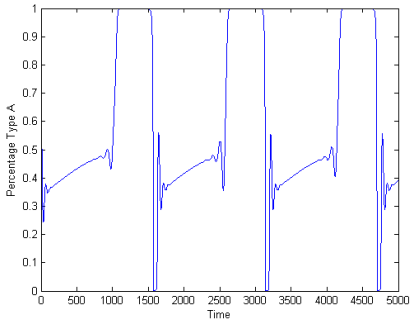
(a) $\mu_A = \mu_B = 0.45$



(b) $\mu_A = \mu_B = 0.1$



(c) $\mu_A = 0.1, \mu_B = 0.005$



(d) $\mu_A = 0.15, \mu_B = 0.005$

FIGURE 17. Example Games

The main concern which comes from examining the above simulations is that changes in learning speeds can generate very similar looking dynamics to changes in the stage game parameters. This implies that any econometric estimation

might be plagued by identification issues. Indeed, my experience running and estimating simulations, discussed in later sections, has revealed identification issues when allowing learning speeds to vary between types. However, it is important to understand that the shifts in parameters affect the dynamics via different mechanisms. In order to find these mechanisms, I focus on the steady states of the system. Steady states are important because even if the system is not eventually driven towards one they will nevertheless be focal points of any periodic behavior around them. A steady state of the above system, (λ^*, a^*, b^*) occurs at the intersection of all three null-spaces:

$$\begin{aligned}
\Delta\lambda = 0 \Rightarrow \lambda^* =& \frac{F^{BB}(a^*, b^*) - F^{AB}(a^*, b^*)}{F^{AA}(a^*, b^*) + F^{BB}(a^*, b^*) - F^{AB}(a^*, b^*) - F^{BA}(a^*, b^*)} \\
\Delta a_t = 0 \Rightarrow \lambda^* =& \frac{a^* - (1 - \sqrt{1 - a^*})}{\sqrt{a^*} - (1 - \sqrt{1 - a^*})} \\
\Delta b_t = 0 \Rightarrow \lambda^* =& \frac{\sqrt{b^*} - b^*}{\sqrt{b^*} - (1 - \sqrt{1 - b^*})}
\end{aligned} \tag{3.9}$$

Notice that changes in the learning speed parameters, μ_A and μ_B appear nowhere in the above equations. Thus, changing the learning speed parameters does not affect the location of the steady states. This means that learning speeds only influence the relative speeds of the dynamics around the steady states. On the other hand, changes in the stage-game parameters (i.e. the coefficients on abilities in the payoffs) will affect the $\Delta\lambda = 0$ surface, and thus have the potential to change both the relative speeds around the steady states and the location of the steady states themselves.

This is a subtle result, but one that has many implications. First of all, this causes issues in the econometric techniques presented later. That is, even though changing learning speeds does not create identical dynamics as changing stage game parameters, they can still be similar enough to cause identification issues (e.g. disperse

posterior distributions and even posterior convergence issues) when allowing the learning speeds between types to vary. More importantly, the result tells us that in order to understand why a population behaves as it does, it is not only necessary to understand the stage game being played but also how each type in the population learns. In terms of policy, this points us to the possibilities of new tools. Instead of trying to "balance" the stage game, it may be equally effective to attempt to influence the adaptability of each type. That is, changing the learning speeds can change which steady state the dynamics go toward, or can change the dynamics from cycling around a steady state to eventually converging on the steady state. However, if one wants to change the location of a steady state, this can only be achieved via manipulation of the stage game. In general, before trying to address an apparent issue within the stage game, the possibility of learning differences must be addressed.

Estimation

Recall the dynamic system from the previous section:

$$\begin{aligned}
\lambda_{t+1} =& \lambda_t \frac{AvgFitA_t}{\lambda_t AvgFitA_t + (1-\lambda_t)AvgFitB_t} \\
a_{t+1} =& \lambda_t(a_t + \mu_A(\sqrt{a_t} - a_t)) + (1-\lambda_t)(a_t - \mu_A(a_t - (1-\sqrt{1-a_t}))) \\
b_{t+1} =& (1-\lambda_t)(b_t + \mu_B(\sqrt{b_t} - b_t)) + \lambda_t(b_t - \mu_B(b_t - (1-\sqrt{1-b_t})))
\end{aligned} \tag{3.10}$$

Notice that the above is a deterministic dynamic system, with no shocks. The earlier discussion in Section II made it clear that the inclusion of stochasticity is important to evolutionary models. Thus, in order to complete the model, a stochastic element must be introduced. The most straightforward way would be to tack an error term onto each equation. Essentially, this would say that one type may see a random increase in population share one period or that there is a random gain/loss to ability

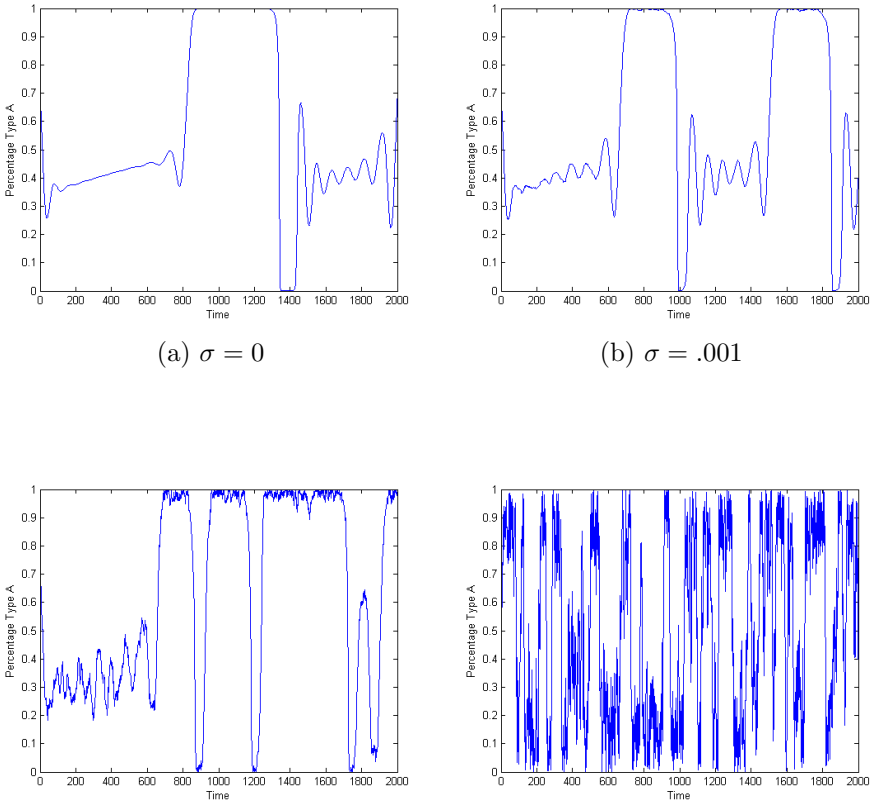
from one generation to the next. That is, both the ability transmission process and the fitness dependent population movements are subject to small random shocks. Unfortunately, the inclusion of these random shock terms is complicated by the fact that each variable in the system is measured on the interval [0,1]. This issue can be dealt with by assuming error terms come from a truncated distribution in the folowing way: For each equation above, define the no-shock transition value as:

$$\begin{aligned}
\bar{\lambda}_t =& \lambda_t \frac{\alpha + AvgFitA(\lambda_t, a_t, b_t)}{\alpha + \lambda_t AvgFitA(\lambda_t, a_t, b_t) + (1-\lambda_t)AvgFitB(\lambda_t, a_t, b_t)} \\
\bar{a}_t =& \lambda_t[a_t + \mu_A(\sqrt{a_t} - a_t)] + (1-\lambda_t)[a_t - \mu_A(a_t - (1-\sqrt{1-a_t})] \\
\bar{b}_t =& (1-\lambda_t)[b_t + \mu_B(\sqrt{b_t} - b_t)] + (\lambda_t)[b_t - \mu_B(b_t - (1-\sqrt{1-b_t})]
\end{aligned} \tag{3.11}$$

Then, incorporating shocks can be done in a straightforward way:

$$\begin{aligned}
\lambda_{t+1} =& \bar{\lambda}_t + \tau_{t+1} \\
a_{t+1} =& \bar{a}_t + \epsilon^a_{t+1} \\
b_{t+1} =& \bar{b}_t + \epsilon^b_{t+1}
\end{aligned} \tag{3.12}$$

Where $\tau_{t+1} \sim TN(0, \sigma^2, -\bar{\lambda}_t, 1-\bar{\lambda}_t)$, $\epsilon^a_{t+1} \sim TN(0, \sigma^2, -\bar{a}_t, 1-\bar{a}_t)$, $\epsilon^b_{t+1} \sim TN(0, \sigma^2, -\bar{b}_t, 1-\bar{b}_t)$, and $TN(0, \sigma, L, R)$ indicates a truncated normal distribution with mean of 0, variance σ^2 and left and right cutoffs, L and R. The below examples show the difference in dynamics for the same model, with $\sigma = 0$, $\sigma = .001$, $\sigma = .01$, and $\sigma = 0.1$:



(a) $\sigma = 0$ (b) $\sigma = .001$

(c) $\sigma = .01$ (d) $\sigma = .1$

FIGURE 18. Varying σ Examples

It should now be clear that the above model has two major complications which prohibit the use of standard estimation techniques. First of all, the equations are all nonlinear. And secondly, the error terms are not normally distributed. Also note that data on ability levels is most likely not observed. Thus, standard state-space estimation techniques such as the Kalman filter, or even the extended Kalman filter, are no longer appropriate. For these reasons, I utilize particle filtering, incorporating it into a Metropolis-Hasting Algorithm, to estimate the posterior distributions of model parameters. A particle filter can be run to approximate the likelihood of observing that data given a set of model parameters. That is, given parameters θ, the particle filter approximates $P(\lambda|\theta)$. In implementing the particle filter, I follow

the procedure described in Fernandez-Villaverde and Rubio-Ramirez (2004, 2007); more specifics can be found in Appendix 3.

Identification

If the parameters of the stage game are of interest, there will most likely be identification issues; especially if the learning speeds are allowed to vary between types. In order to examine the different identification issues that might arise, I once again focus on the null-spaces of the system:

$$\begin{aligned}
\Delta\lambda = 0 \Rightarrow \lambda^* =& \frac{F^{BB}(a^*, b^*) - F^{AB}(a^*, b^*)}{F^{AA}(a^*, b^*) + F^{BB}(a^*, b^*) - F^{AB}(a^*, b^*) - F^{BA}(a^*, b^*)} \\
\Delta a_t = 0 \Rightarrow \lambda^* =& \frac{a^* - (1 - \sqrt{1 - a^*})}{\sqrt{a^*} - (1 - \sqrt{1 - a^*})} \\
\Delta b_t = 0 \Rightarrow \lambda^* =& \frac{\sqrt{b^*} - b^*}{\sqrt{b^*} - (1 - \sqrt{1 - b^*})}
\end{aligned} \tag{3.13}$$

Examining the $\Delta\lambda = 0$ locus reveals two facts. First, adding the same scalar to each payoff function will not alter the $\Delta\lambda = 0$ locus. In addition, multiplying each function by the same scalar will not change the $\Delta\lambda = 0$ locus. However, multiplicative increases in payoff functions will not alter the dynamics, but scalar additions may. To see this, recall that the equation of motion for λ_t is:

$$\lambda_{t+1} = \lambda_t \frac{\lambda_t F^{AA} + (1 - \lambda_t) F^{AB}}{\lambda_t(\lambda_t F^{AA} + (1 - \lambda_t) F^{AB}) + (1 - \lambda_t)(\lambda_t F^{BA} + (1 - \lambda_t) F^{BB})} \tag{3.14}$$

If each payoff function were transformed from F^{ij} to MF^{ij}, the new constant, M, would cancel out of the equation as it is in every term in both the numerator and

denominator. Thus, multiplicative increases of the payoff functions have no impact on the equation of motion, i.e. the system is unchanged. However, adding a scalar to each payoff function *will* change the equation of motion. That is, both the numerator and denominator will have M *added* to it, and thus M will not cancel out. For example, if M was extremely large, and coefficients on ability were relatively small, both types would always have almost equal fitness regardless of a_t, b_t, or even λ_t. Intuitively, this difference is quite simple. Changes in λ are due to differences in fitness between types. A multiplicative change will not change relative fitness, but an additive change in payoffs *will* change the relative payoffs.

The above issues, in addition to the learning speeds discussion from the previous section, lead to the following identification strategy. First of all, if the parameters of the stage game are known, the researcher can easily identify σ, μ_A, and μ_B. However, if the parameters of the stage game are unknown, several assumptions need to be made. In order to alleviate the stage game parameter scale issues, one or more of the payoff coefficients will be assumed known. In determining the structure of the underlying stage game, the interest is on the size of the payoff coefficients relative to one another, so this assumption is not very restrictive. However, it may also be necessary to assume that the learning speeds are equal across types, i.e. that $\mu_A = \mu_B$.

Estimation Examples

In this section, I present simulation and estimation examples from various 2-strategy models all of which include ability. I chose to focus on the 2 strategy case for simplicity and to demonstrate the drastic change in the possibilities as far as system dynamics are concerned. However, in the Appendix 3, I demonstrate that the

estimation technique is not only easily applied to standard models without ability, but that it is also easily extended into cases with more than 2 strategies.

Known Game

First, I will demonstrate the estimation technique's ability to recover learning speed and variance parameters when the underlying stage game is known. Data was simulated for the below model:

	A	B
A	$1 + a_t$	$1 + 1.8(1 - a_t) - 0.8(1 - b_t)$
B	$1 + 1.2(1 - b_t) - 0.8(1 - a_t)$	$1 + b_t$

TABLE 13. Simulation Stage Game

The above model was simulated for 125 time periods, with initial conditions $\lambda_0 = .6$, $a_0 = .2$, $b_0 = .4$, and $\sigma = 0.03$. The simulated time series for λ_t is shown below:

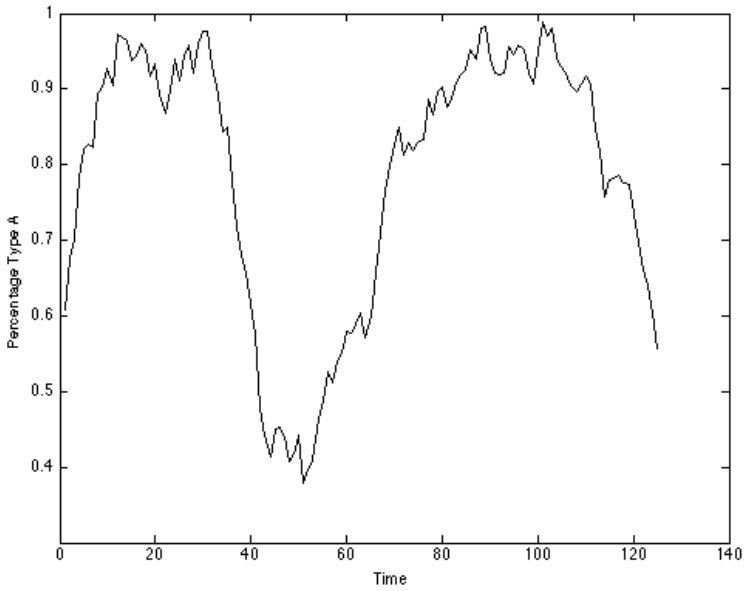
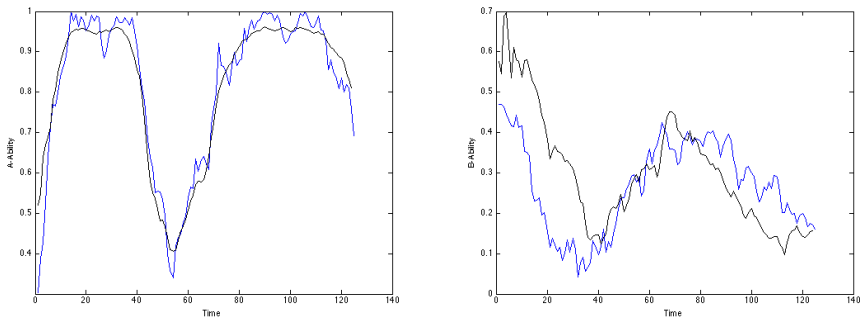


FIGURE 19. Simulated Population Share Data

Assuming the parameters, μ_A, μ_B, and σ are known, but only observing data on λ_t, particle filtering can be used to obtain estimates of the unobserved states a_t and b_t. This was done using a relatively small number of draws, N = 250. The below figures show the true values of the simulated state time series (in Blue), and the means of the Particle Filter estimates (in Black):



(a) A-Ability (b) B-Ability

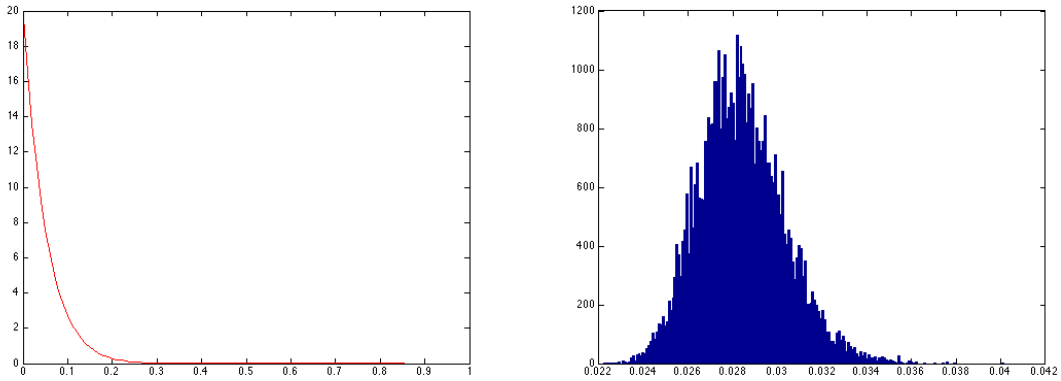
FIGURE 20. PF Estimates When Parameters Are Known

In this model, particle filtering does a good job of recovering the state. However, the main usefulness of the particle filter is that it allows the researcher estimate the likelihood $P(\lambda|\theta)$ for any set of parameters, θ.

In this first example, I assume that the structure of the underlying game is known, and that the only goal is to estimate the posterior distributions of $\sigma, \mu_A,$ and μ_B. To do so, I implement a Metropolis-Hastingings algorithm. The prior distributions are $\mu_A, \mu_B \sim Beta(1.5, 1.5)$ and $\sigma \sim Beta(20, 1)$. The algorithm utilized a random walk proposal distribution with a diagnol variance-covariance matrix. The distributions shown below were obtained with 50,000 draws after a 5,000 draw burn in. For each draw, the likelihoods were calculated using a particle filter with N = 100. Summary statistics for the posterior distributions are listed in the below table. Histograms of the posterior distributions for each parameter is shown below, alongside its respective prior:

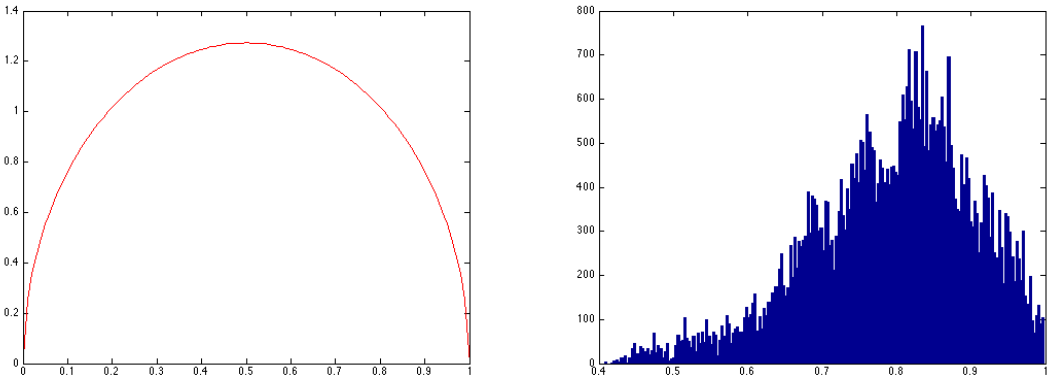
Parameter	True	Mean	95% HPDI
σ	0.03	0.0284	[0.0247, 0.0321]
μ_A	0.7	0.7923	[0.5870, 0.9964]
μ_B	0.1	0.1667	[0.0726, 0.2751]

TABLE 14. Posterior Distributions - Summary Statistics



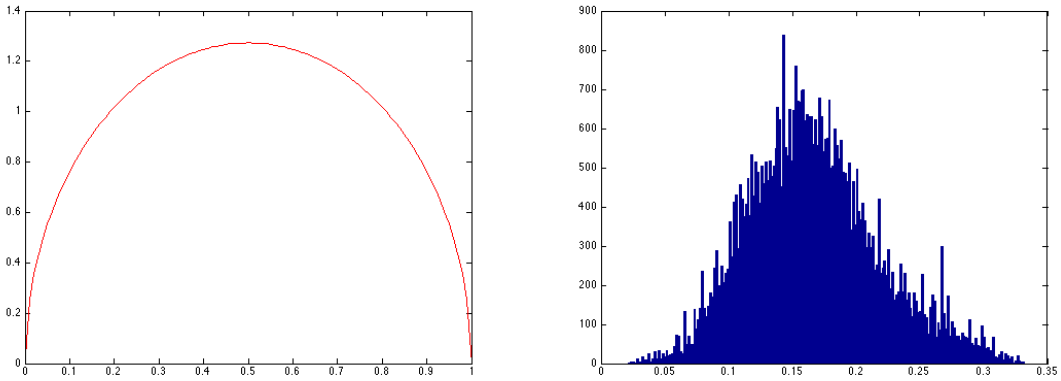
(a) $P(\sigma)$ (b) $P(\sigma|\lambda)$

FIGURE 21. Prior and Posterior for σ - Known Game



(a) $P(\mu_A)$ (b) $P(\mu_A|\lambda)$

FIGURE 22. Prior and Posterior for μ_A - Known Game



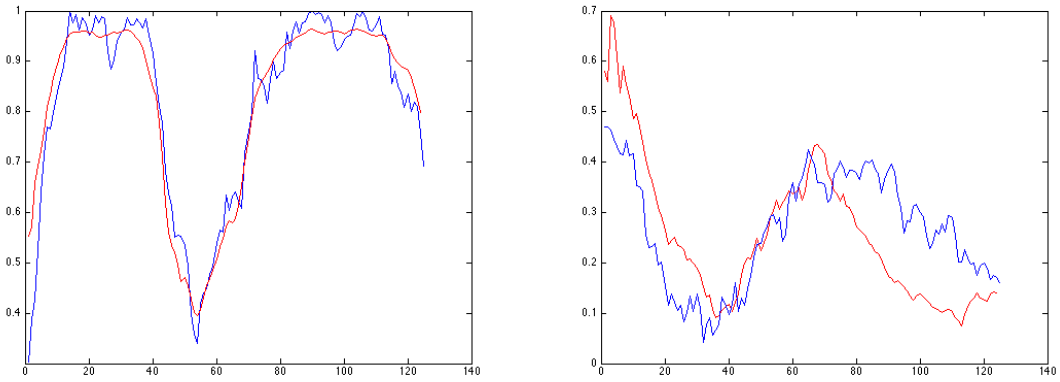
(a) $P(\mu_B)$ (b) $P(\mu_B|\lambda)$

FIGURE 23. Prior and Posterior for μ_B - Known Game

The time required to calculate each likelihood is substantial, especially when used in a Metropolis-Hastings algorithm. Utilizing parallel processing techniques can often drastically improve the computation time. For MH algorithms, one possible way to do this is to run multiple chains in parallel. Unfortunately, the burn in period for this estimation can be quite long, and thus running multiple chains would not speed up the process much. Instead, I used a technique called pre-fetching as suggested by Brockwell (2006) and Strid (2010). At each step, several future proposal draws were selected and the likelihood for each was calculated on a separate processor, i.e. in parallel. Based on these likelihoods, each draw was then accepted or rejected based on the corresponding acceptance probability. For example, I utilized four processors with a static pre-fetching scheme. Assuming the chain is at draw $\theta^{[g]}$, I simulated two proposals of $\theta^{[g+1]}$, and one proposal of $\theta^{[g+2]}$ for each $\theta^{[g+1]}$ proposal. This essentially guaranteed that each step would produce at least two actual draws in the chain: two rejections of $\theta^{[g+1]}$ proposals, an accept/reject or accept/accept based on the first $\theta^{[g+1]}$ proposal. Furthermore, if the first proposal is rejected and the second accepted, it would obtain 3 actual draws of the chain. Thus, in the time it normally takes to do

1 draw, pre-fetching allows 2 or more draws (depending on the number of processors available) to be obtained. In the case of 4 cores, I saw a more than double speed up in the time it took to run.

In addition to the above posteriors, the particle filter allows for the recovery of the unobserved state. A time series of the mean of the distribution of a_t and b_t is shown below compared to their true values:



(a) A-Ability (b) B-Ability

FIGURE 24. Mean estimates of a_t and b_t - Known Game

Unknown Game

While estimation of learning speeds and disturbance variance may be of interest, the estimation technique can also recover parameters of the underlying stage game. Combined with estimates of the unobserved ability levels, a complete time series of the unobserved fitness levels could also be approximated. In order to demonstrate this, data was generated from the following model. The underlying stage game is shown below. The other parameters were set as follows: $\mu_A = \mu_B = 0.25$, $\sigma = 0.03$, $\lambda_0 = .6$, $a_0 = 0.2$, and $b_0 = 0.7$. The generated data is shown below for T = 125:

	A	B
A	$1 + a_t$	$1 + 2(1 - a_t) - (1 - b_t)$
B	$1 + 1.5(1 - b_t) - (1 - a_t)$	$1 + b_t$

TABLE 15. Stage Game - Simulated Data

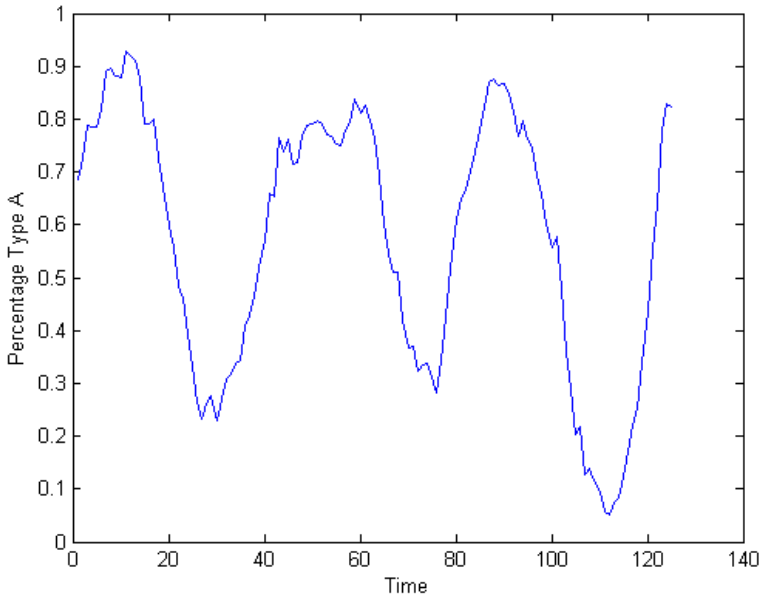


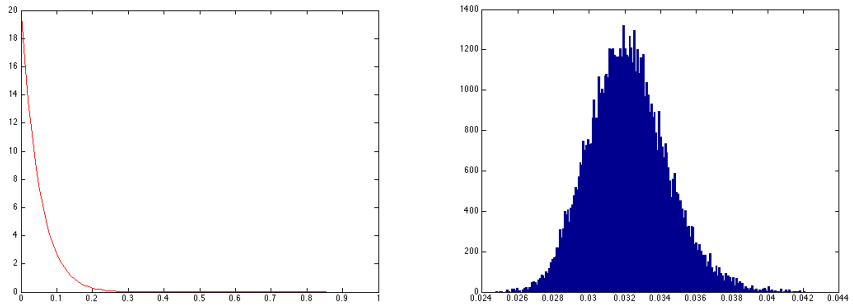
FIGURE 25. Simulated Population Shares

In order to assure that the model is identified, I assumed the following. First, it is assumed that learning speeds are equal; i.e. $\mu_A = \mu_B$ is known. I also assumed that 3 of the stage game parameters were known: C = 1, E = -1, and H = 1. The priors on σ and μ are the same as before, while the priors for the stage game parameters are $B, D \sim N(1, 5)$, and $E \sim TN(0, 5, -1, \infty)$. The rest of the estimation procedure followed that described in the previous section, with 75,000 draws after a burn in of 5000 draws and N = 200 in the particle filter.

Summary statistics for the estimated parameters are shown in the below table, and the posteriors for each parameter, and their corresponding priors, are shown in the below figures:

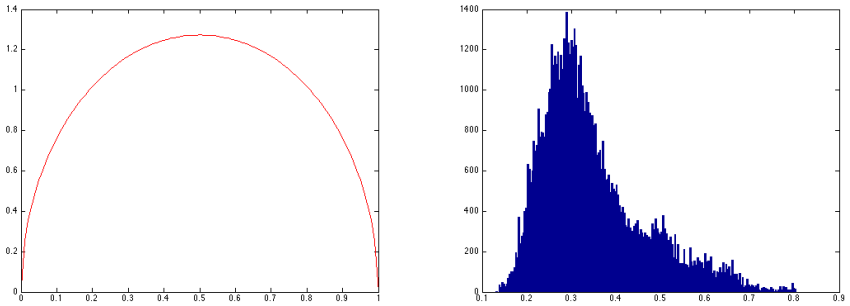
	True	Mean	Median	Mode	90% HPDI
σ	0.03	0.0322	0.0321	0.0305	[0.0284, 0.0356]
μ	0.25	0.3470	0.3177	0.1812	[0.1769, 0.5317]
D	2	2.2284	2.1937	3.2383	[1.5113, 2.8958]
F	1.5	1.3513	1.3302	1.7568	[1.0587, 1.6350]
G	-1	-0.7928	-0.8185	-0.9063	[-0.9997, -0.5874]

TABLE 16. Posterior Summary Statistics



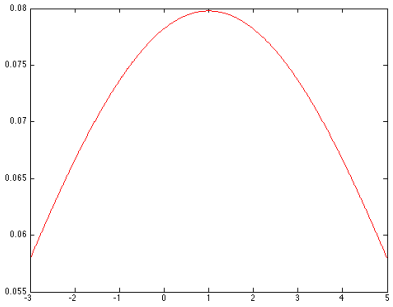
(a) $P(\sigma)$ (b) $P(\sigma|\lambda)$

FIGURE 26. Prior and Posterior for σ - UnKnown Game

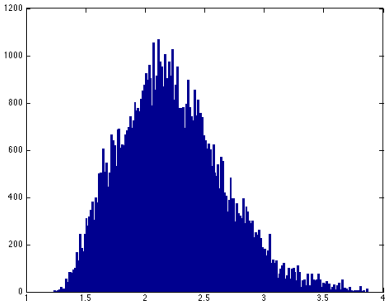


(a) $P(\mu)$ (b) $P(\mu|\lambda)$

FIGURE 27. Prior and Posterior for μ - UnKnown Game

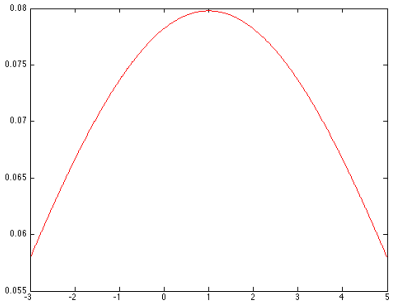


(a) $P(D)$

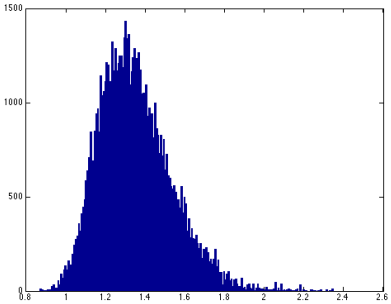


(b) $P(D|\lambda)$

FIGURE 28. Prior and Posterior for D - UnKnown Game

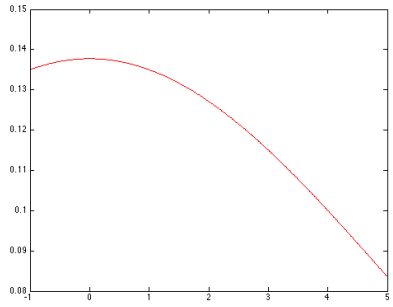


(a) $P(F)$

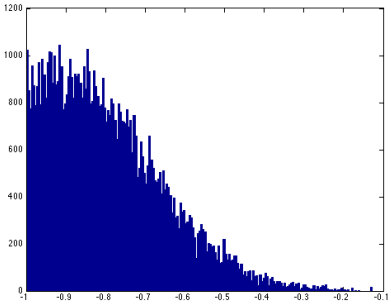


(b) $P(F|\lambda)$

FIGURE 29. Prior and Posterior for F - UnKnown Game



(a) $P(G)$



(b) $P(G|\lambda)$

FIGURE 30. Prior and Posterior for G - UnKnown Game

In addition to those displayed, since the estimation recovers the joint distribution of parameters, other statistics of interest can also be computed. For example, the researcher might be interested in whether D > F, which would indicate that A-types had a structural advantage over B-types. Here, $P(D > F|\lambda)$ can be easily calculated as $P(D > F|\lambda) = 0$. Alternatively, we can look at the distribution of D - F, as the below figure shows:

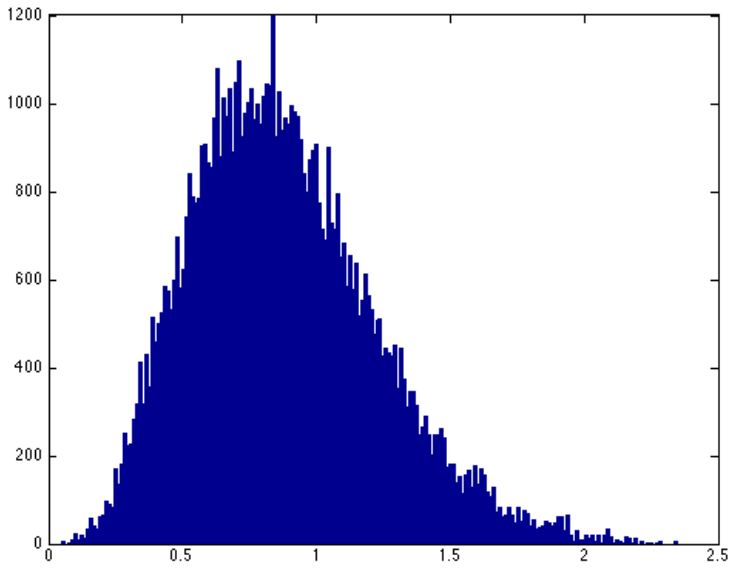


FIGURE 31. $P(D - F|\lambda)$

Thus, upon observing the population data, the researcher can answer questions regarding structural advantages. In this case, it is clear that the A-types have a structural advantage over the B-types; i.e. if an A-type was matched against an equally skilled B-type, the A-type would get a higher payoff.

The above examples illustrate that a wide array of questions can be empirically investigated using the experienced evolution model, and the only data needed is that of population shares. In addition to estimating learning speeds, disturbance variance,

and stage game parameters (and thus unobserved fitness levels), questions regarding competing models can also be addressed by bayesian model comparison using the techniques described in Chib and Jeliazkov (2001) for calculating marginal likelihoods when using MH algorithms. While not done in this paper, in addition to testing between the differences in learning speeds, it would not be difficult to test between different proposed learning processes (as opposed to the single learning specification used in this paper). This would be useful in any setting where understanding the learning process was important, e.g. in experimental economics data. Moreover, instead of estimating all game parameters, two likely underlying games could simply be compared.

Conclusion

Evolutionary game theory has made a large impact on both economic theory and theoretical work in outside fields. However, as this paper discussed, the theoretical models often failed to produce clear empirical techniques. This shortcoming forces researchers to rely on quantitative matching and approximation of unobservables by proxy variables, or to move on to more complicated models that lose the simple intuitions that were so originally attractive about evolutionary game theory. As this paper demonstrated, this does not necessarily have to be the case, and experienced evolution offers one way to navigate these issues.

The experienced evolution model offers theoretical tool for explaining various types of observed behavioral patterns that would be inconsistent with many other evolutionary game models. It also does so in a way which does not sacrifice the researcher's ability to tell a simple, intuitive story. Besides theoretically explaining complex behavior, the model also offers a clear econometric approach which allows

empiric studies to go beyond qualitative matching. In addition to estimating unknown parameters of interest, it also allows the recovery of unobserved data such as the progression of ability and fitness. More importantly, while it may not be appropriate in every circumstance, the experienced evolution model demonstrates that clear, robust empirical work can be done when applying evolutionary game theory.

CHAPTER IV

Q-LEARNING

Introduction

A wide variety of models in economics center around individuals making choices that maximize the sum of their expected discounted utility. In these situations, the individual is assumed to be rational and thus makes decisions according to an optimal policy. An important question, then, is what happens if the individual is not rational and does not know what the optimal policy is. While movements away from rationality have been thoroughly investigated in other areas, the subject of policy learning has been explored very little in the economics literature. This paper presents the Q-learning model of Watkins (1989) and proposes it as a model of individual policy learning. Moreover, it demonstrates how it can be used in the estimation of dynamic discrete choice (DDC) models.

In terms of modeling individual behavior in dynamic environments, I argue that policy learning is an area that deserves much more attention than it has received so far. Indeed, the idea that an individual has solved for an optimal value function seems questionable in many circumstances, especially when it is the very calculation of optimal policies that makes the researcher's estimation computationally challenging. That is, if well trained economists need very fast computers to solve for optimal policy functions, expecting lay people to have solved for one may be unreasonable. To be clear, I am not arguing that it is impossible for individuals to act optimally. For example, there may be some situations where individuals innately know the optimal

solution. On the other hand, it seems equally likely that individuals are thrown into situations they have never experienced and have no idea how to act optimally initially.

DDC models are often used in economics; in labor, I/O, behavioral, and even macro economics. Structural DDC estimation models are a particular class of DDC models that attempt to uncover parameters governing individual preferences and expectations. This method of estimating choice probabilities is not only valuable in understanding behavior, but also allows for greater insight in terms of policy implications because there is a theoretical explanation as to what affects choice probabilities and why (Keane,Todd,and Wolpin 2011). While very demanding computationally, advances in both techniques and technology have made structural DDC models much more feasible. However, the fact remains that all current structural models are built on the assumption of rationality; i.e. that individuals have solved for an optimal value function prior to making decisions. This should be a concern because if individuals are not rational, standard structural models may inaccurately estimate parameters. The Q-learning model offers not only an alternative structural DDC model for researchers who want to move away from the rationality assumption, but also provides a way to test for the significance of learning effects if they are a concern.

The investigation into policy learning is implemented as follows. First, I demonstrate that the Q-learning model is a simple and flexible model of policy learning. As a behavioral model, it has a small number of economically meaningful parameters. These include the usual discount factor, but introduce new parameters regarding adjustment speeds and expectation formation. I then show that this model can be easily used for estimation of DDC models. Using Bayesian MCMC techniques on simulated data, I show that the Q-learning model performs well at recovering true

parameter values. In addition, the simulated data are used to illustrate possible issues with standard structural estimation if the rationality assumption is incorrect. Lastly, using marginal likelihood analysis, I demonstrate that the Q-learning model can be used to test for the significance of learning effects.

The rest of the chapter is organized as follows. Section II outlines the classic structural model and describes the Bayesian DP estimation procedure of Imai, Jain, and Ching (2009). Section III introduces the Q-learning model and discusses how individuals might use it in a continuous state setting. Sections IV and V discuss the simulation and estimation of data coming from both structural models, while Section VI compares the performance of each estimation technique. Finally, Section VII concludes and discusses possible future research involving Q-learning.

Structural Discrete Dynamic Choice Models

Modeling dynamic choices has been increasingly important in economics, and a widely used model is that of a rational, forward looking agent who maximizes the sum of their expected discounted utility. The typical assumption is that agents facing such a dynamic programming problem make decisions following an optimal policy function. This setup has been the basis for many economic models, in both macroeconomics and microeconomics. While Section III of this paper discusses how to move away from the rationality assumption, this section describes the standard model through the lens of structural discrete dynamic choice estimation in order to use as a baseline for discussion and comparison later on.

Classic Framework

Oftentimes, we observe individuals making choices across time. If choices made in the present have an impact on choices made in the future, static choice models may need to be replaced with ones that allow these choices to be correlated across time. One type of estimation strategy involves modifying the static models to incorporate inter-temporal correlation. For example, this is accomplished by the dynamic probit model, which allows the latent variable to be modeled as an autoregressive process. Another approach is to use structural models. Structural models assume individuals are forward looking and attempt to maximize their expected discounted reward over time.

Structural models are a valuable tool for understanding individual behavior, but have a downside in that they are often difficult to estimate, being very computationally demanding. In fact, until recently, structural discrete dynamic choice models represented one of the few cases where Bayesian estimation methods were infeasible compared to classic approaches. However, recent advances in techniques and technology have made them much more feasible. The remainder of this section lays out the basic framework, and describes the Bayesian DP algorithm of Imai, Jain, and Ching (2009) in the estimation of such models.

This paper considers individuals in infinite horizon Markov decision problems, and the classical model is set up as follows. Each period, an individual observes the current state, $\mathbf{s}_t \in S$ and must choose an action, $c_t \in C$. Upon choosing an action, the individual will receive an immediate reward, $R(\mathbf{s}_t, c_t)$. The individual also knows that the state transition is a Markov process that depends on the current state and action chosen; specifically, denote the probability of observing $\mathbf{s}_{t+1}$ based on current state and chosen action $\mathbf{s}_t$ and c_t as: $f(\mathbf{s}_{t+1}|\mathbf{s}_t, c_t)$. The objective of the individual is

formulate a plan of action that maximize their expected sum of discounted payoffs. A standard result is that the individual will solve for an optimal policy function that results in a value function satisfying the below Bellman equation:

$$V(\mathbf{s}_t) = \max_{c_t \in C} R(\mathbf{s}_t, c_t) + \beta E[V(\mathbf{s}_{t+1})|\mathbf{s}_t, c_t] \tag{4.1}$$

This framework of rational decision making has been used in many applications. One of the first applications was in Rust (1987) who modeled a manager's choices regarding bus maintenance. Individual decisions regarding schooling (Cameron and Heckman 1998), risky behavior (Arcidiacono, Sieg, and Sloan 2007), contraception choice (Hotz and Miller 1993, Carro and Mira 2006) and labor supply decisions (Imai and Keane 2004 and Stinebrickner 2001) also commonly use the structural estimation framework. Extensions of the framework to dynamic games have also been used to investigate firm decisions, such as in the concrete industry (Collard-Wexler 2011) and Radiostation format choice (Sweeting 2007).

Estimation and the Bayesian Dynamic Programming Algorithm

Now suppose there is a set of parameters that are of interest, called θ that includes the individual's discount rate, β, and any other parameters regarding the reward function and state transition equations. For clarity in later discussion, let $\theta = \{\beta, \theta_R, \theta_F\}$, where θ_R and θ_F represent parameters involved in the reward and transition equations respectively. In order to use the above model as an econometric model, some adjustments are needed. Following Imai, Jain, and Ching (2009), it is assumed there is an unobserved state that impacts the reward function which is then denoted as $R(\mathbf{s}, \epsilon, c, \theta_R)$, where ϵ is a vector with *iid* individual shocks $\epsilon_c \sim N(0, \sigma_\epsilon)$ for each choice $c \in C$. Throughout the remainder of the paper, I assume $R(\mathbf{s}, \epsilon, c, \theta_R) =$

$P(\mathbf{s}, c, \theta_R) + \epsilon_c$ where $P(\mathbf{s}, c, \theta_R)$ represents the deterministic element of the reward function. Lastly, because of the addition of ϵ and θ, I change notation slightly and denote the true value function as $V(\mathbf{s}, \epsilon, \theta)$.

Denote the value of a particular choice, c, in state $(\mathbf{s}, \epsilon)$ as:

$$\bar{V}(\mathbf{s}, \epsilon, c, \theta) = P(\mathbf{s}, c, \theta_R) + \epsilon_c + \beta E[V(\mathbf{s}', \epsilon', \theta)|\mathbf{s}, c, \theta] \tag{4.2}$$

Then the probability of observing choice c_t is:

$$Pr(c_t = c) = Pr(\bar{V}(s, \epsilon, c, \theta) > \bar{V}(s, \epsilon, c', \theta) \forall c' \neq c \in C) \tag{4.3}$$

The likelihood function combines choice probabilities with the observations on rewards and states. Specifically, let $Y = \{c_t, s_t, s_{t+1}, R_t\}_{t=1}^{T}$. Then the likelihood can be stated as:

$$L(\theta|Y) = \prod_{t=1}^{T} Pr(\bar{V}(s_t, \epsilon, c_t, \theta) > \bar{V}(s_t, \epsilon, c', \theta) \forall c' \neq c_t) \phi(R_t - P(s_t, c_t, \theta_R), 0, \sigma_\epsilon^2) f(s_{t+1}|s_t, c_t, \theta_F) \tag{4.4}$$

where $\phi(\mathbf{x}, \mu, \Sigma)$ denotes a multivariate normal pdf with mean vector μ and variance-covariance matrix Σ. The above likelihood has two features that will be important distinctions later on. Notice that without taking choices into account, we could just estimate the payoff and transition parameters based on the state and payoff observations. However, the observed choices also give us information about the payoff and transition parameters as all parameters are necessary to calculate $\bar{V}(\theta, s, \epsilon, c)$. More importantly, what the above illustrates is that the value function must be known in order to calculate choice probabilities. This feature of structural

DDC models is what causes them to be computationally intensive. For any set of parameters, value function iteration must be performed to calculate the value function and only then can a likelihood be computed.

One solution to this issue is the method of conditional choice probabilities (CCP) suggested by Hotz and Miller (1993). This method suggests restating the value function as a function of choice probabilities. In a first stage, choice and transition probabilities can be calculated nonparametrically based only on observations, then in a second stage differences in value functions can be stated in terms of these probabilities and can be used to estimate the structural parameters. More recently, Imai, Jain, and Ching (2009) developed what is known as the Bayesian DP Algorithm to alleviate the computational burden. The Bayesian DP algorithm accomplishes this by essentially nesting the Bellman operator into a Metropolis-Hastings(MH) algorithm. This allows the algorithm to estimate and solve the dynamic programming problem simultaneously, greatly reducing the computational cost of estimating DDC models. Bayesian estimation methods allow for relatively easy model comparison via marginal likelihood analysis. For this reason, all estimation of the standard structural model will utilize the Bayesian DP algorithm. Appendix B briefly explains its implementation in this paper, but more detailed and general explantations can be found in Imai, Jain, and Ching (2009) and in Ching et al. (2012).

Q-Learning

The field of machine learning has a large existing literature on learning optimal policy functions. One of the major models in this literature is the Q-learning model, originally proposed by Watkins (1989) and further investigated in Watkins and Dayan (1992). Q-Learning is a reinforcement style learning model that has been extensively

analyzed and extended in the machine learning literature. The rest of this section will briefly outline how Q-Learning works in a Markov decision problem. For clarity, it will be introduced in the case of a finite state space, and then the extension to continuous states will be discussed.

Finite State Space Q-Learning

Consider an individual facing a Markov decision problem (MDP) where the state and action spaces are discrete and finite. Specifically, let c_t and $\mathbf{s}_t$ denote the action chosen and state in time t. At every time period, the individual has values assigned to every state-action pair, called Q-values. Denote the Q-value for state-action pair $(\mathbf{s}, c)$ at time t as $Q_t(\mathbf{s}, c)$. If the individual chooses action c, they will update the associated Q-value; all other Q-values will remain the same. The update process works as follows:

$$Q_{t+1}(\mathbf{s}, c) = \begin{cases} (1-\alpha)Q_t(\mathbf{s}, c) + \alpha(R_t + \beta \max_{c' \in C} Q_t(\mathbf{s}_{t+1}, c')) & \text{if } c = c_t \text{ and } \mathbf{s} = \mathbf{s}_t \\ Q_t(\mathbf{s}, c) & \text{else} \end{cases} \tag{4.5}$$

Where β represents the individual's discount rate and α represents the learning rate. Essentially, the learning rate determines how much weight the individual places on new experiences. A high learning rate implies sharp adjustments, whereas a lower α has smaller, smoother adjustments. Watkins and Dayan (1992) show that this learning process will converge to the optimal policy as long as enough experimentation across the state and action spaces occurs and if action spaces are discrete. One way to achieve this is to simply give the individual an initial experimentation phase where they choose actions randomly for M-periods. The above represents the

basic Q-learning model that is the baseline model for a class of learning algorithms called reinforcement learning within the machine learning literature. More advanced algorithms have been developed including extensions into continuous state and action spaces (Smart and Kaelbling 2000, Gaskett, Wettergreen, and Zelinsky 1999).

The individual makes a decision based on the current Q-values associated with the current state. There are several ways to model this choice rule. For example, one could assume that the individual simply takes the action with the highest Q-value. Waltman and Kaymak (2008) utilize a logit choice rule:

$$Pr(c) = \frac{\exp(Q_t(\mathbf{s}_t, c))}{\sum_{c' \in C} \exp(Q_t(\mathbf{s}_t, c'))} \tag{4.6}$$

I choose to utilize the above choice rule to stay consistent with Waltman and Kaymak (2008), although none of the estimation techniques are reliant on the specification.

In this specification, the individual is assumed to make a probabilistic choice based on their Q-values. While this is a standard way to model Q-learning, there may be a concern that the update process is not consistent with the choice rule in as far as the value of the subsequent state is attributed to a single option. An alternative model might have the Q-values hit with type I extreme value shocks, and the individual choosing the option with the highest Q-value. This might be preferable in that it alleviates the issue previously described and this is also a typical interpretation of standard choice models. Normally, these two models would be equivalent, but in learning models the unobserved shocks carry over into the Q-values of subsequent periods, increasing the difficulty of estimation. While the remainder of this paper utilizes Equation 4.7, above, appendix 9 details how to go about estimating the

alternative model utilizing a particle filter. Most importantly, the main results of this paper are the same in either case.

These assumptions of choice rule have some notable implications; specifically that the learning process is no longer guaranteed to converge to the optimal policy. For example, while the choice rule allows for some experimentation in states, if the individual started with very extreme initial Q-values they might be unlikely to explore different options in different states within some feasible time frame. However, the paper's goal is not to investigate convergence. It is simply to propose a valid and estimable model that describes how individuals might learn policies. Indeed, this process would allow both the learning of an optimal policy and the learning of a non-optimal policy depending on initial values and model parameters. This is beneficial because it might not only be the case that real individuals have to learn optimal policies, there might also be situations where individuals have learned non-optimal policies.

To my knowledge, there has been little economic research involving Q-learning. The work that does exist focuses on investigating Q-learning and cooperative behavior. Walter and Kaymak (2007) investigate Q-learning agents in an iterated Prisoner's Dilemma game and find that cooperative behavior can be a result. More notable is the paper cited earlier by Waltman and Kaymak (2008). In this paper, the authors apply finite state Q-learning to a Cournot model of competition. They find that Q-learning was able to generate behavior consistent with collusion. This result was important because none of the standard reinforcement learning models used in economics, could generate this type of behavior.

Continuous State Q-Learning

Being able to extend the Q-learning model to continuous state space is very important for several reasons. First, discretizing the state space can make the lookup table approach to the Q-values infeasible as the number of necessary Q-values to track can grow rapidly; i.e. it is subject to the curse of dimensionality. Moreover, allowing for a continuous state space greatly increases the applicability of the Q-learning model.

The issue in a continuous setting is that an individual in state $\mathbf{s}$ has most likely never been at this exact point in the the state space. So the question is, how does the individual form Q-values for each action at this state? Fortunately, the machine learning literature has investigated several ways of extending reinforcement learning into continuous states. While there are a litany of advanced methods, I propose that individuals simply use a locally weighted average to estimate Q-values in a particular state based off past experiences. This process is sometimes referred to as "lazy" learning, and has been used previously in the machine learning literature (Atkenson, Moore, and Schaal 1997 and Forbes and Andre 2000).

Consider an individual in a binary choice (C = {A,B}) MDP with a continuous, Euclidean state space of dimension K. At time t, the individual is in state $\mathbf{s}_t \in S$. Based on past experiences, the individual must assign a value to action A and action B. In determining these values, the individual looks to their past experiences in choosing A and B. Specifically, let $\mathbf{Q}^{At}$ denote the $N \times 1$ vector of previous Q-values for action A that were taken in corresponding states $\mathbf{S}^{At}$, which is an $N \times K$ matrix, where N indicates the number of previous A-experiences. The subscript t indicates that these sets of previous experiences may change as time moves forward. Because the individual may not have any previous experiences (e.g. has never chosen option A), initial beliefs must also be specified. I make the assumption that initial beliefs

can be characterised by two constants, q_0^A and q_0^B. These represent the initial value placed on each choice in any state.

For each of the N data points, the individual determines the distance each past experience is from the current state. That is, for each $K \times 1$ column vector $\mathbf{s}_0 \in \mathbf{S}^{At}$, the individual finds $D(\mathbf{s}_t, \mathbf{s}_0)$. This distance function can be simple or complex, and the remainder of this paper assumes individuals use a skewed Euclidean distance function:

$$D(\mathbf{s}_t, \mathbf{s}_0) = \sqrt{\sum_{k=1}^{K} \omega_k (s_{tk} - s_{0k})^2} \tag{4.7}$$

Where ω_k represents the weight that the individual places on dimension k of the state space, and these weights satisfy $\sum_{k=1}^{K} \omega_k = 1$. After determining distances, the individual then places a weight, W_n, on each datapoint that depends on each point's distance from the current state. Again, there are multiple options that could be used here, such as the Gaussian Kernel $W_n = exp(\frac{-(D_n)^2}{\rho})$, or the nearest neighbors weight, places a weight of 1 on the closest J points, and a weight of 0 on all others. Throughout the remainder of the paper, I assume individuals use the Gaussian Kernel weight with scale parameter ρ. Neither the distance or weighting functions presented here are new to reinforcement learning and more detailed examples can be found in Atkenson, Moore, and Schaal (1997) and Forbes and Andre (2000).

Once weights, W_n are assigned, the individual forms their expectation of the value of A using a weighted average of their past experiences and initial belief:

$$\widetilde{Q}(\mathbf{s}_t, A) = \frac{\sum_{n=1}^{N} W_n \mathbf{Q}_n^{At} + q_0^A}{\sum_{n=1}^{N} W_n + 1} \tag{4.8}$$

Note two important features of the above model. First, the influence of the initial condition decreases as the number of past experiences, N, increases. Second, the importance of the initial belief increases as ρ decreases. This is because as ρ decreases the weights of all other experiences will decrease while the initial point will always have a weight of $\exp(0) = 1$.

The individual then forms their expectation of the value of option B, $\widetilde{Q}(\mathbf{s}_t, B)$, in a similar manner (but the weights will not be the same since B-experiences will have occurred in different states). Once both values $\widetilde{Q}(\mathbf{s}_t, A)$ and $\widetilde{Q}(\mathbf{s}_t, B)$ are determined, the individual makes a choice in accordance with the choice rule stated earlier, which in the case of binary choice becomes:

$$Pr(c_t = A) = \frac{\exp(\widetilde{Q}(\mathbf{s}_t, A))}{\exp(\widetilde{Q}(\mathbf{s}_t, A)) + \exp(\widetilde{Q}(\mathbf{s}_t, B))} \tag{4.9}$$

Finally, after making a choice, c_t, the individual receives a reward, $R_t = P(\mathbf{s}_t, c_t) + \epsilon_{ct}$ and observes the subsequent state $\mathbf{s}_{t+1}$. Based on this information, the individual will update their estimate of $\widetilde{Q}(\mathbf{s}_t, c_t)$ using the standard Q-learning update:

$$Q(\mathbf{s}_t, c_t) = \alpha\widetilde{Q}(\mathbf{s}_t, c_t) + (1 - \alpha)[R_t + \beta \max_{c \in C} \widetilde{Q}(\mathbf{s}_{t+1}, c)] \tag{4.10}$$

Where the expectations of Q-values in the subsequent state, $\widetilde{Q}(\mathbf{s}_{t+1}, c)$, are formed in the same manner using a weigthed average of past experiences and the initial point. Once the update is complete, the individual adds the value-state pair to the appropriate set of past experiences. That is, the updated value $Q(\mathbf{s}_t, A)$ will be added to the set $\mathbf{Q}^{At}$ to create $\mathbf{Q}^{At+1}$ only if action A is taken. This makes sense because the individual cannot update their expectations unless the particular action

is chosen, just like in finite space Q-learning. If action A is not chosen, $\mathbf{Q}^{At+1} = \mathbf{Q}^{At}$ and $\mathbf{S}^{At+1} = \mathbf{S}^{At}$; i.e. the set of past experiences remains the same.

While this sounds complicated, it is actually quite straightforward. In deciding what action to take, the individual simply looks at past experiences, and gives more attention to those that occurred at states that are close to the current situation. After deciding which action to take and viewing the consequences, the individual updates a Q-value for that action in the state just visited. This updated value is a weighted average between their previous expectation, and the sum of the current reward and their discounted estimate of the value of the next state. Moreover, note that at each step, the individual is only adding one additional observation to their set of initial Q-values off of which they form expectations. Just like in the finite state case, if the individual never takes action c, they will never update a Q-value for that action.

Overall, individuals then differ on three dimensions: How much you value future payoffs (β), how quickly you update your valuations (α), and how you define closeness (the ω_k weights and ρ) when forming expectations. Note that, while not explored in this paper, time could also be included in the state so that individuals give more weight to observations that occurred more recently. Another important extesion not explored in this paper would be to allow the individual to view certain actions as similar. That is, in the current setup, the individual bases expectations for a particular action only off of previous experiences with that action. The individual could expand the set of past experiences to include those of actions deemed similar. This would be especially useful in extending the model to continuous actions, an area which has been investigated by the machine learning literature. Lastly, note that the way that expectations of current Q-values are formed and updated is not specific to the estimation model. It only requires that a process, that can be defined with a set of

parameters, is specified. Moreover, the use of a Bayesian estimation procedure would allow for the comparison of alternative expectation/update processes via marginal likelihood analysis.

As a whole, the Q-learning model is a straightforward and flexible way to model individual policy formation. In doing so, also offers up a host of questions that are important for understanding individual behavior in such settings. For example, are there parameter values that lead to better policies; or are there any relationships between the parameters themselves? In addition to theoretically explaining an individual's learning process, all of the model parameters are estimable. Thus, as the next section describes, the Q-learning model also functions as a tool in discrete dynamic choice estimation.

Estimation

Given a set of parameters, $\theta = \{\alpha, \beta, \omega_A, \rho, q_0^A, q_0^B\}$, and given information on choices, payoffs, and states, the Q-values for the individual can be reconstructed. Once again, let $Y = \{c_t, s_t, s_{t+1}, R_t\}_{t=1}^{T}$, and let $Y^t = \{c_j, s_j, s_{j+1}, R_j\}_{j=1}^{t-1}$ represent the history of observations up until time t. Then the likelihood for any set of data can be formed as:

$$L(Y|\theta) = \prod_{t=1}^{T} \frac{\exp(\widetilde{Q}_t(\mathbf{s}_t, c_t, \theta, Y^t))}{\sum_{c' \in C} \exp(\widetilde{Q}_t(\mathbf{s}_t, c', \theta, Y^t))} \tag{4.11}$$

Where $\widetilde{Q}_t(\mathbf{s}_t, c_t, \theta, Y^t)$ again denotes the individuals evaluation of $Q_t(\mathbf{s}_t, c_t,)$ prior to making a choice, but it is now explicit that these values are dependent not only on the observed history up to that point, but more critically on the parameter set θ. Note that parameters regarding payoff and transition functions are not necessary

to estimate. This is because Q-learning is a *model-free* method of learning, and the particular functional form of payoffs and transitions is not considered by the individual. The individual still forms expectations to help make choices; but these are formed based off past experiences only. Thus, since the individual's choices do not take functional form into account, the observed choices give no information regarding functional forms. However,the researcher can still estimate the underlying payoff and transition function parameters using:

$$L(Y|\theta) = \prod_{t=1}^{T} \frac{\exp(\widetilde{Q}_t(\mathbf{s}_t, c_t, \theta, Y^t))}{\sum_{c' \in C} \exp(\widetilde{Q}_t(\mathbf{s}_t, c', \theta, Y^t))} \prod_{t=1}^{T} \phi(R_t - P(\mathbf{s}_t, c_t, \theta_R), 0, \sigma_\epsilon^2) f(s_{t+1}|s_t, c_t, \theta_F) \tag{4.12}$$

As following sections show, the estimation procedure performs very well at recovering parameter values when taken to simulated data. This good identification in the Q-learning model relies on observing the individual re-visiting similar areas of the state space throughout time. One implication of this is that the effects of learning should be a concern in any problem that features this type of data. Policy learning, then, may not be a concern in several classic applications of DDC models such as optimal stopping time, fertility decisions, schooling choices, etc. that don't often feature an individual revisiting a point in the state space. At the same time, though, there are many applications that do have this feature, such as consumer brand choice and strategic interactions between firms. As subsequent sections demonstrate with a simple example, the presence of learning can severely bias the results of standard structural DDC models. Thus, while policy learning in these situations is an area that should be explored, even if it's not the researcher's primary objective they can

still use the Q-learning model to test for the presence of learning effects via marginal likelihood analysis.

Simulation

In order to demonstrate the performance of both the traditional and the Q-learning structural models, I simulate data from an individual facing a particular MDP, then use the simulated data to recover estimates of the parameter values. Data is generated from both a rational individual and a Q-learning individual facing the same problem. I first describe the MDP facing the individual, and then describe the simulated data.

The Individual's Markov Decision Problem

Every period, the individual must choose between options A and B. Choosing an option will increase future skill in this option. For example, picking A today will make you more experienced at option A in the future. Experience in each option in conjunction with the individual's choice determines the immediate payoff whose functional form is discussed shortly . Similarly, failing to choose an option will result in a loss of experience. Experience in option A or B is measured on the interval $(0, 1)$ and evolves as follows:

$$a_{t+1} = \begin{cases} a_t + \gamma_A(1 - a_t) + v_t & \text{if } c_t = A \\ a_t - \frac{1}{2}\gamma_A a_t + v_t & \text{else} \end{cases} \tag{4.13}$$

Where v_t is distributed truncated normal, with mean 0 and standard deviation σ_A, where the truncation happens appropriately to ensure $a_{t+1} \in (0, 1)$, and b_t evolves in the exact same manner with parameters γ_B and σ_B. In the remainder of the paper,

let $\mathbf{s}_t = (a_t, b_t)$ denote the current state, and let $F(s_t)$ denote the transition function with no shock ($\sigma_A = \sigma_B = 0$); i.e. $\mathbf{s}_{t+1} = F(\mathbf{s}_t)$ is the deterministic transition.

The payoff function for the individual depends on which action is chosen and is listed below:

$$P(c_t, a_t, b_t) = \begin{cases} 30 + 10a_t - 80((a_t - H_A)^2 + (b_t - H_B)^2) - \frac{3}{.1+\sqrt{(a_t-H_A)^2+(b_t-H_B)^2}} & \text{if} c_t = A \\ 30 + 10b_t - 80((a_t - H_A)^2 + (b_t - H_B)^2) - \frac{3}{.1+\sqrt{(a_t-H_A)^2+(b_t-H_B)^2}} & \text{if} c_t = B \end{cases} \tag{4.14}$$

In words, the payoff function is essentially a hill, but with a sudden sink where the top should be, located at $a_t = H_A$ and $b_t = H_B$. The bowl is rotated slightly different for the payoffs associated with A and B such that choosing option A will yield a higher payoff if $b_t = 0$, and likewise for picking option B. The highest points lie just around the sink at (H_A, H_B) . Thus, individuals have to figure out the best way to navigate through the state space. Appendix C contains figures showing two angles of the B-Payoff function and an overlay of the B and A Payoffs when $H_A = H_B = 0.5$.

Simulation

Data was simulated from each model with shared parameters $\beta = 0.9$, $\gamma_A = 0.2$, $\gamma_B = 0.2$, $\sigma_A = \sigma_B = 0.15$, $\sigma_\epsilon = 1$, $H_A = H_B = 0.5$. Each model was simulated for $T = 200$ time periods, with the same initial state $\mathbf{s}_0 = (0.1, 0.1)$.

Rational Individual Simulation

I first use value function iteration to approximate the solution to the optimal value function. In order to do so in the continuous state space, I utilize the random grid approach suggested by Rust (1997). A random grid of 1000 points was generated

and at each point, 1000 shocks, (ϵ_A, ϵ_B), were drawn from a N(0,1) distribution. The below picture shows an approximation of the implied policy function, which shows the probability of choosing option A or B.

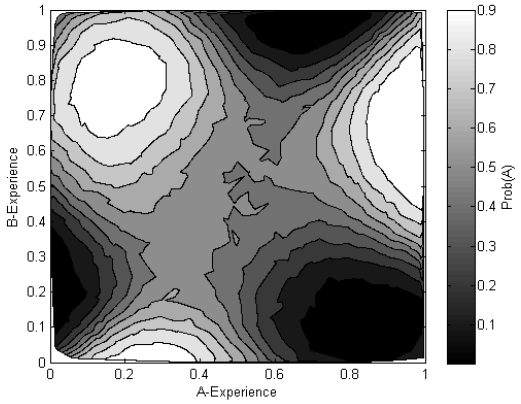
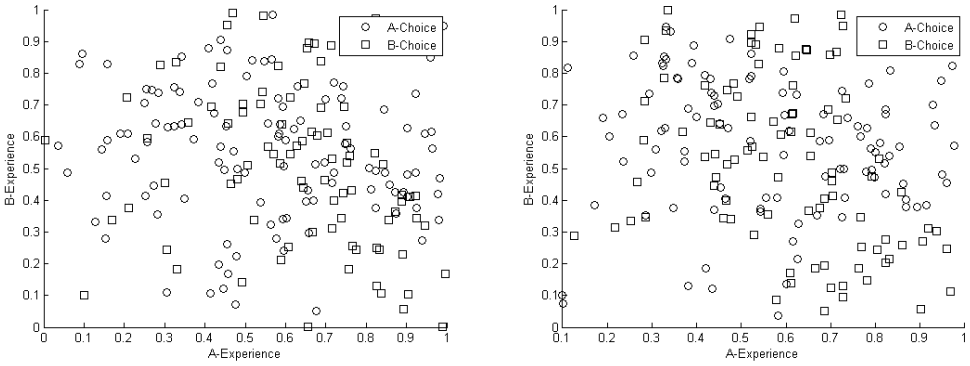


FIGURE 32. Simulated Data - Policy Function

After the value function has converged, choice data was generated using those values as follows. Given a state, $\mathbf{s}_t$, the individual calculates:

$$\widetilde{V}(\mathbf{s}_t, \theta, c) = P(\mathbf{s}_t, c, \theta) + \beta \sum_{n=1}^{N} \frac{V(\mathbf{s}_n, \theta) f(\mathbf{s}_n | \mathbf{s}_t, c)}{\sum_{k=1}^{N} f(\mathbf{s}_n | \mathbf{s}_t, c)} \tag{4.15}$$

Where the $V(\mathbf{s}_n, \theta)$ values are the converged values over the N random grid points. This is done for actions A and B. Then ϵ_A and ϵ_B are simulated, and the individual chooses action A if $\widetilde{V}(\mathbf{s}_t, \theta, B) + \epsilon_A > \widetilde{V}(\mathbf{s}_t, \theta, B) + \epsilon_B$. The below figure shows 2 examples of the choice history and state history for an individual over 200 time periods.



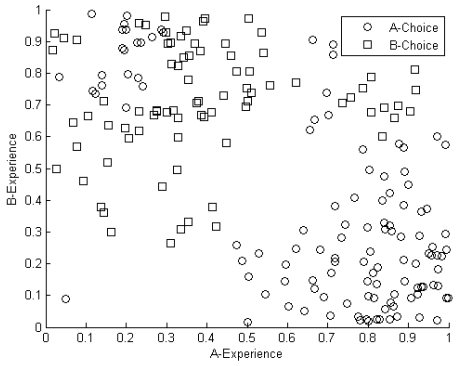
(a) Simulated Choices 1 (b) Simulated Choices 2

FIGURE 33. Example of Simulated Choices

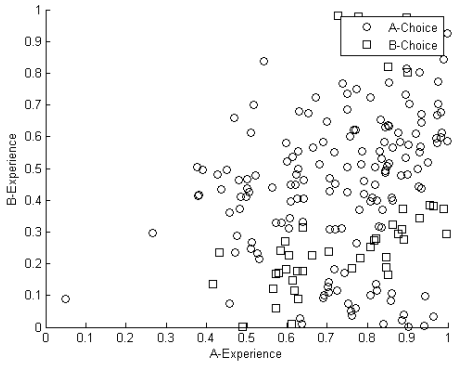
For use later in the paper, 250 simulations were generated based on the above policy function, but all generated data looked similar in that choices followed the policy function. For example, the lower-right area of the state space is dominated by B-choices, while the upper-right is mostly A-choices. The 250 simulated data sets all had similar patterns.

Q-Learning Simulation

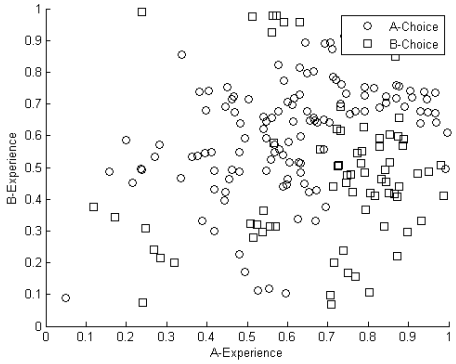
In the case of Q-learning, $\beta = 0.9$, $\alpha = 0.75$, $\omega_A = 0.6$, $\rho = .003$, $q_0^A = q_0^B = 10$ and payoffs followed the function presented earlier. Multiple sets of simulated data were generated, and the below scatterplots show the location and choices made across the sample period for four example histories:



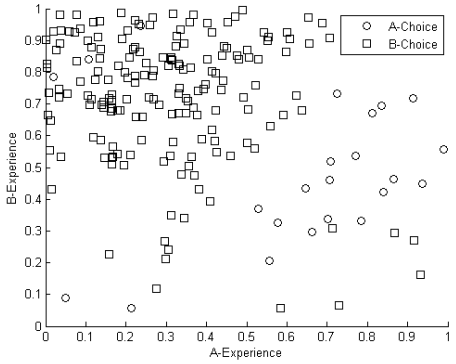
(a) Simulated Choices 1



(b) Simulated Choices 2



(c) Simulated Choices 3



(d) Simulated Choices 4

FIGURE 34. Example of Simulated Choices - Q-learning

These figures demonstrate that unlike the rational agent simulations, there can be very different histories from the same initial conditions under Q-learning. The rational agent follows a set policy that, in this problem, creates consistent movements across the state space. In contrast, Q-learners do not have a set policy, allowing for different resulting histories, some of which do not explore the state space to the same degree as the rational agent's path.

Estimation

Standard Structural Model Estimation

In this model, $\theta = \{\beta, \sigma_\epsilon, \gamma_A, \gamma_B, \sigma_A, \sigma_B, H_A, H_B\}$. The priors for $\sigma_\epsilon, \sigma_A, \sigma_B$ were assumed to be Gamma(2,2), all other parameters had a uniform prior over the interval $[0, 1]$. Because the focus of the paper is on the structural parameters, and not the payoff or transition parameters, all other parameters were known (e.g. the coefficients on the a_t term in the payoff function, but those could be estimated as well if they were of interest). 10,000 draws were obtained from the posterior after a 5,000 period burn in. $N(g)$ was set so that by the 15,000th draw, the random grid consisted of 1,000 points. Proposals came from a random walk proposal distribution. The acceptance rate of the BDP algorithm was 35%.[1] The below table lists summary statistics on the marginal posteriors for each parameter:

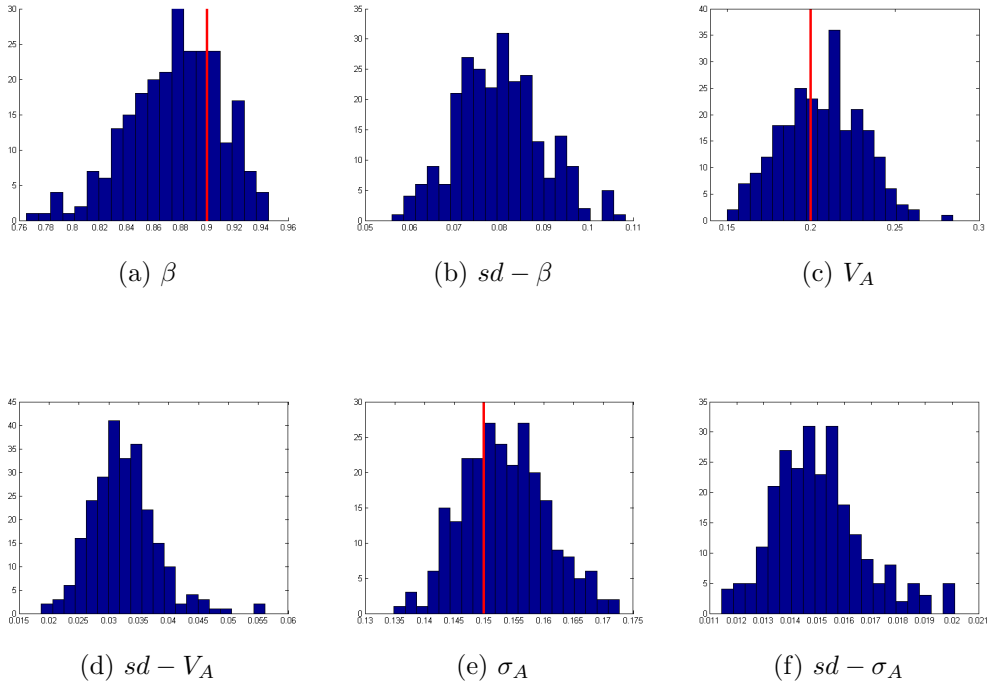
Parameter	Mean	90% HPDI	True Value
β	0.8892	(0.8010,0.9988)	0.9
σ_ϵ	1.1517	(0.9925,1.2705)	1
γ_A	0.1903	(0.1448,0.2534)	0.2
γ_B	0.2086	(0.1517,0.2681)	0.2
σ_A	0.1576	(0.1389,0.1866)	0.15
σ_B	0.1559	(0.1421,0.1853)	0.15
H_A	0.5028	(0.4991,0.5062)	0.5
H_B	0.4971	(0.4936,0.5006)	0.5

TABLE 17. Posterior Summary Statistics - Traditional Model

[1] Computation time of the algorithm was somewhat slow, around 4 hours. While this is partly due to inefficient coding, it is worth noting that the BDP algorithm cannot take advantage of prefetching (Strid 2010), a parallel processing technique used in conjunction with standard MH algorithms. This is because the calculation of the likelihood depends on previous proposal draws.

As the above table shows, the Bayesian DP estimation procedure does a good job of recovering the parameter values. All posterior means are close to true values, and all HPDIs contain the true parameter value.

In order to demonstrate that this was not just a result of a particularly good draw of data, 250 sets of simulated data were generated, and estimated in the same fashion. Across all datasets, similar results were found. The below figures illustrate this, showing the distribution of posterior means, and posterior standard deviations for several model parameters



(a) β (b) $sd-\beta$ (c) V_A

(d) $sd-V_A$ (e) σ_A (f) $sd-\sigma_A$

FIGURE 35. Monte Carlo Experiment - BDP - Posterior Means and S.d.

Q-Learning Estimation

In this model, $\theta = \{\beta, \alpha, \omega_A, \rho, \sigma_\epsilon, \gamma_A, \gamma_B, \sigma_A, \sigma_B, H_A, H_B\}$. Once again, the priors for $\sigma_\epsilon, \sigma_A, \sigma_B$ were assumed to be Gamma(2,2), ρ had a Gamma(1,2) prior, all other parameters had a uniform prior over the interval $[0, 1]$. Estimation of

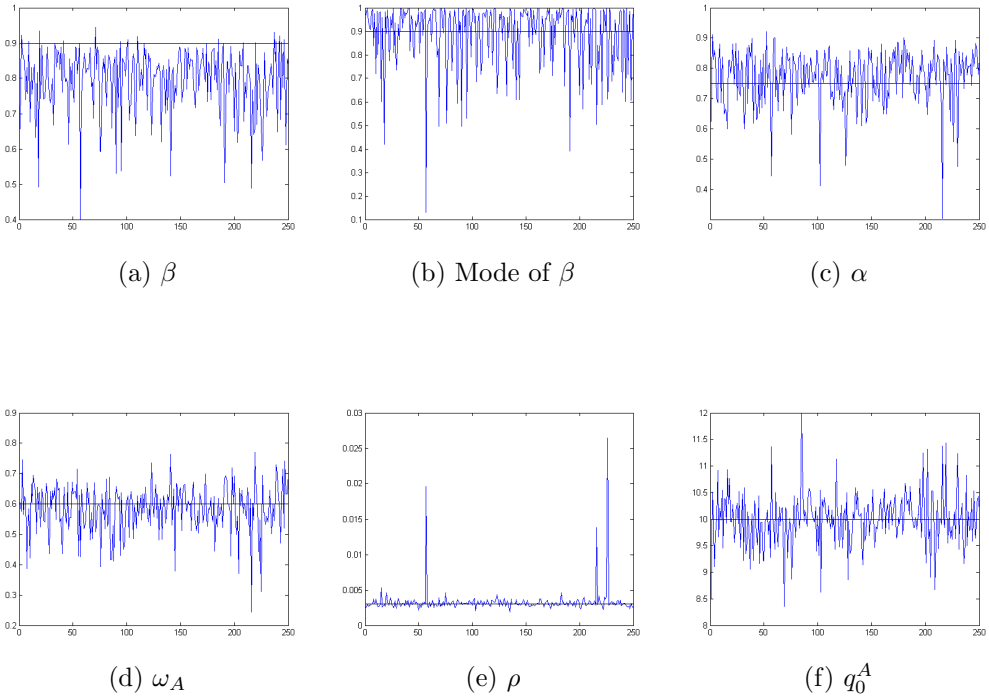
the Q-Learning model used a Metropolis-Hastings algorithm with a random walk proposal. Again, 10,000 draws were obtained from the posterior after a 5,000 period burn in. The acceptance rate for the sampler was 31.7%. Parallel processing was taken advantage of by utilizing pre-fetching (Strid 2010), which essentially generates multiple proposal paths during each run. Using 12 cores, this decreased the total computation time by a factor approximately 3. Compared to the Bayesian DP algorithm, the estimation of the Q-learning model was much faster in obtaining the same number of posterior draws (15 minutes versus multiple hours). The below table lists summary statistics on the marginal posteriors for each parameter:

Parameter	Mean	90% HPDI	True Value
β	0.8604	(0.7188,0.9998)	0.9
α	0.8195	(0.6654,0.9944)	0.75
ω_A	0.6228	(0.5630,0.6831)	0.6
ρ	0.0029	(0.0025,0.0032)	0.003
q_0^A	9.7216	(9.0420,10.3776)	10
σ_ϵ	0.9458	(0.8682,1.0215)	1
γ_A	0.2201	(0.1653,0.2538)	0.2
γ_B	0.2037	(0.1497,0.2477)	0.2
σ_A	0.1359	(0.1260,0.1438)	0.15
σ_B	0.1575	(0.1438,0.1706)	0.15
H_A	0.4950	(0.4916,0.4987)	0.5
H_B	0.4982	(0.4943,0.5014)	0.5

TABLE 18. Posterior Summary Statistics - Q-Learning

Clearly, when the data is coming from a Q-learning individual, the estimation procedure is able to identify between the structural parameters of interest, $\alpha, \beta, \rho,$ and ω_A. The parameter ρ has an especially accurate posterior. In contrast to the standard structural model, the recovery of β is not as good, with a lower mean and wider HPDI. While this is by no means a bad performance, but it is important to note it is not as accurate as the standard structural model in this particular case.

Good parameter identification in any simulation experiment is contingent on the characteristics of the dataset. In order to demonstrate these results are consistent, 250 datasets from the same model were generated, and the same estimation procedure was performed. The below figures plot the means of the posteriors for β, α, ω_A, ρ, and q_0^A across all 250 simulated data sets:



(a) β (b) Mode of β (c) α

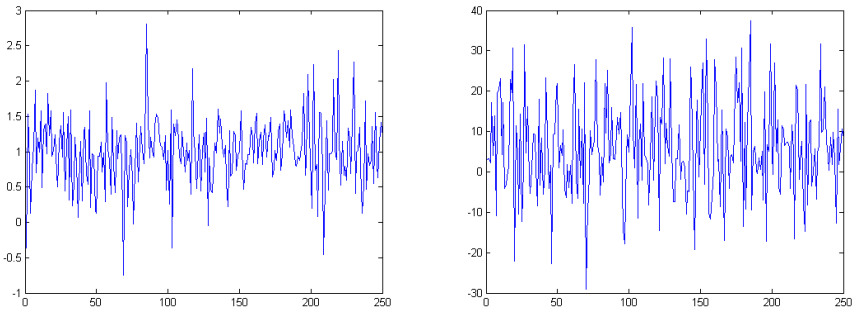
(d) ω_A (e) ρ (f) q_0^A

FIGURE 36. Monte Carlo Experiment - Q-learning

Note that, on average, the performance is very good. The discount parameter, β, consistently has a mean below the true value. This is because this is a posterior of a truncated variable whose true value is close to the truncation point. A better statistic may be an approximation of posterior mode, which is shown in panel (b) above using histograms of 150 bins.

Another investigation was done into the sensitivity of the intial condition assumptions. Recall that the estimation procedure assumed to know the true value

of $q_0^B = 10$. The same estimation was done on all datasets assuming $q_0^B = 1$, and very similar results were obtained as before. The estimation was also done assuming the initial values were unknown, but that $q_0^A = q_0^B$. Again, similar results obtained. The main difference across the estimations was the posterior for q_0^A. In the case of $q_0^B = 1$, the posterior for q_0^A was centered around 1. In the latter case, the posterior for q_0^A wandered, indicating it may not be well identified. Demonstrating this, the means of the marginal posterior for q_0^A is are shown below for each case:



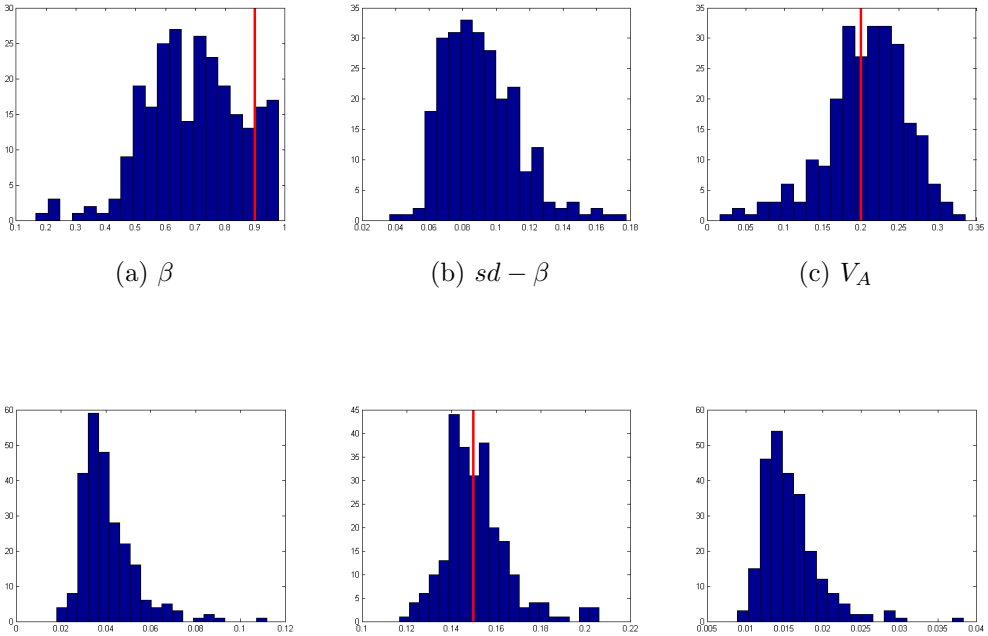
(a) $q_0^B = 1$ (b) $q_0^A = q_0^B$

FIGURE 37. Monte Carlo Experiment - Alternate Initial Conditions

While this has demonstrated the good performance in the case of binary choice models, it should also be easily extended to cases involving more than two choices. To see why, note that the estimation procedure is very similar to other standard methods in the following sense. For each observation, values are calculated for each available option. Based on these values, the probability of observing a particular choice can be calculated. In the case of the Q-learning model, the calculation of each value is computationally simpler than in the standard structural model, and so extending the model beyond binary choice would be no more difficult than extending the standard structural model.

Model Comparison

While the above showed that both estimation models performed well, another goal of this paper is to investigate how the standard model performs if learning effects are present. In order to make these comparisons, sets of data were generated from both models, Q-Learning and the traditional model, and both estimation procedures were carried out on both sets of data. First, the standard model was taken to 250 sets of simulated Q-learning data. The below graphs show the distribution of posterior means and standard deviations when the standard model is taken to the Q-learning data.



(a) β (b) $sd - \beta$ (c) V_A

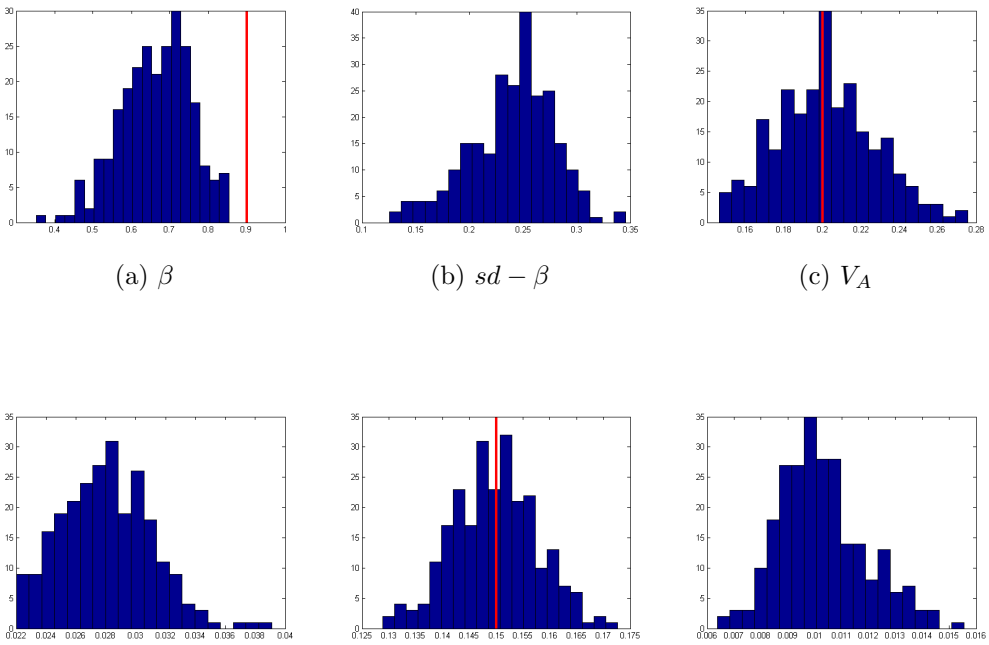
(d) $sd - V_A$ (e) σ_A (f) $sd - \sigma_A$

FIGURE 38. Monte Carlo Experiment - BDP on Q-Learning Data

As the above figures show, if learning effects are present the standard model may inaccurately estimate parameters of interest. Specifically, note that multiple datasets caused the standard model to severely underestimate the true value of β. Over 70%

of the estimation runs have a posterior mean less than 0.8; compared to 41% in the Q-learning estimation runs. Another concerning feature is that the standard deviations of each posterior was still rather small. Most have a standard deviation below 0.1, and none have one more than 0.2, implying that the researcher would see rather tight posteriors for the parameter β. Thus, the inappropriate model in this case does not give any sign that there is an issue without comparing it directly to the Q-learning model.

A similar investigation could be done concerning Q-learning estimation given data from a rational agent. To do so, the Q-learning model was estimated for the 250 rational agent datasets. The below graphs show the mean and standard deviations of the same parameters for the Q-learning estimation on the rational agent data:



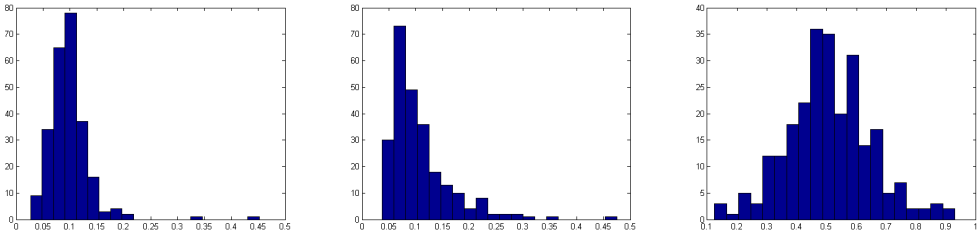
(a) β (b) $sd - \beta$ (c) V_A

(d) $sd - V_A$ (e) σ_A (f) $sd - \sigma_A$

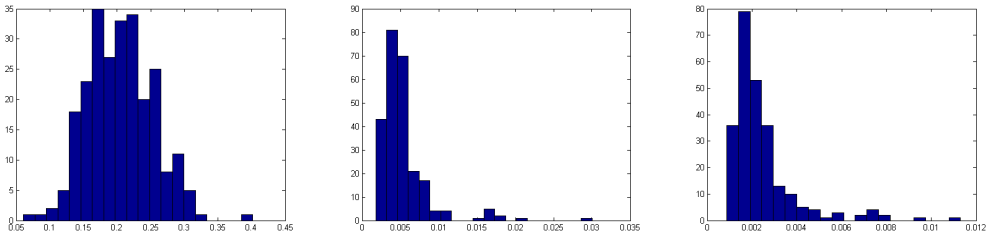
FIGURE 39. Monte Carlo Experiment -Q-learning on Rational Agent Data

Notice that, as indicated by the standard deviation histogram, the Q-learning model gives very disperse posteriors for β, especially compared to the standard model in this case. However, in terms of the payoff and transition parameters, the Q-learning estimation still performs well. This is because the Q-learning estimation procedure does not use observations on choice to recover these.

Another point of interest is the values for the other parameters in the Q-learning model when estimating data from a rational agent. The below figures show a histogram of the posterior means and standard deviations for α, ω_A, and ρ across all 250 rational agent datasets:



(a) α (b) $sd - \alpha$ (c) ω_A



(d) $sd - \omega_A$ (e) ρ (f) $sd - \rho$

FIGURE 40. Monte Carlo Experiment -Q-learning on Rational Agent Data

The most interesting feature of the above figures is the consistently low estimates of α. Recall, that these are estimates based on observations from a rational individual. There is then no true value of α for comparison, and it might be expected that the

posteriors would be centered at 0.5 with a large standard deviation (like most of the posteriors for ω_A). Instead, the posteriors almost all feature means below 0.15, and with standard deviations below 0.2. In fact, out of all the 250 data sets, only 12 had 95% HPDIs that contained anything above 0.2. This can be explained in the following way: The parameter α essentially represents how quickly an individual updates their policies. A fully rational individual follows an optimal policy, and thus never adjusts their policy. From this perspective, it actually seems rather consistent that the estimated values of α would be very low, as this indicates the individual makes very small policy adjustments. If a researcher was concerned with the possibility of learning effects, low estimates of α would be one indication that any learning effects are small. A more robust test afforded by the Bayesian approach lies in the comparison of marginal likelihoods.

Marginal Likelihood Analysis

Another benefit of using a Bayesian approach is that competing models can be directly compared using Marginal Likelihood analysis. Both models use an MH-algorithm, so marginal likelihoods are calculated using the method of Chib and Jeliazkov (2001). If model fit and prediction are the researcher's main concern, non-structural models of dynamic choice may also be considered. One popular model is the Dynamic Probit model that I will also estimate and compare for each dataset. I give a brief explanation below, but for more detailed examples involving dynamic probit models, please see Chauvet and Potter (2005), Franses and Paap (2000), or Fossati (2011).

The basic idea behind a dynamic probit model is that the value of the latent variable can influence future values of the latent variable. Specifically, let Z_t denote

the latent variable, such that if $Z_t > 0$, the individual will choose option A at time t. This latent value is determined by the following equation:

$$Z_t = \theta Z_{t-1} + \beta' X_t + \epsilon_t \tag{4.16}$$

Where $\epsilon_t \sim N(0,1)$ and X_t denotes a column vector of observables at time t. In my specific example, X_t included the current skill levels, the square of skill levels, the product of skill levels, current average payoffs from each option, and the most recent payoff received from each option. The estimation of dynamic probit models typically uses a Gibbs sampling technique, and as such the marginal likelihood calculation of Chib (1995) must be used.

Marginal likelihood values were calculated for each of the three models across both the rational agent datasets and the Q-learning datasets. For any dataset, there are six possible orderings. The below table lists the number of times each ordering was observed across both types of data:

Ordering	BDP Datasets	Q-Learning Datasets
$BDP > Q > DP$	246	1
$BDP > DP > Q$	3	0
$Q > BDP > DP$	1	5
$Q > DP > BDP$	0	240
$DP > BDP > Q$	0	3
$DP > Q > BDP$	0	1

TABLE 19. Marginal Likelihood Analysis

As the above figures demonstrate, the true model is almost always selected by marginal likelihood analysis as the most likely. Specifically, the correct model was selected 249 times in the rational agent data, and 245 times in the Q-learning data.

In the case of Q-learning data, the dynamic probit model usually performed better than the standard structural model. There were only three cases where the standard model was more likely than the Q-learning model, and the dynamic probit performed better only four times. In the rational agent datasets, dynamic probit was never selected as the most likely model, and only outperformed the Q-learning model three times.

From this exercise, it should be clear that if the true data-generating process is in fact Q-Learning, the most appropriate estimation model is the Q-Learning model. Thus, in addition to looking at the estimated values of α, a researcher can use marginal likelihood analysis to test for the significance of learning effects in the data. The estimation of the Q-learning model does not require much computation time, and so this comparison does not add much time for the researcher assuming the standard model was being estimated in a Bayesian framework.

Discussion

Important differences exist between each of these models in their assumptions and uses that deserve more attention. The estimation goals of the classic structural model are very different from those of Q-learning. While the discount rate, β, is a shared parameter of interest, it is the only one. The remaining parameters of interest in the structural model are with regards to the payoff function and transition equations. That is, the researcher wants to understand the underlying problem facing the individual, and uses the individual's choices to help uncover these parameters. However, this recovery is done assuming the individual has indeed solved the problem. This is in stark contrast with the Q-learning model. Other than β, the three parameters of interest in the Q-learning model (at least as discussed in this paper)

were the learning rate α and the parameters regarding expectations, ω_k and ρ. The model does not use choice observations in trying to recover the underlying payoff function or transition equations, because it is assumed that the individual does not know these. Thus, if the individual does not know the underlying structure of the problem, the observed choices cannot tell us anything about it.

These two very different structural models lie at opposite ends on a spectrum of rationality. To clarify the objectives of this paper, I am not arguing that the rationality assumptions are any less plausible than those of the Q-learning model. Rather, both are somewhat extreme assumptions necessary to help simplify the real world and create models from which clear insights can be drawn. As was stated earlier, there are different situations where policy learning may be of interest, and others where it most likely does not apply. The contribution of this paper lies not in presenting a "better" structural model, but rather in presenting an *alternative* model; one that allows researchers to move away from the assumptions of rationality if learning is a concern for their particular problem.

Finally, an important point highlighted in earlier examples is that if a researcher is only concerned with estimating payoff and transition parameters, they should still be implicitly concerned about Q-learning. If an individual is acting according to Q-learning, their choices will not accurately reflect the underlying payoff and transition parameters. Furthermore, a Q-learner's decisions may be very path-dependent whereas the traditionally rational agent has a time-invariant policy they adhere to. So while a Q-learner may learn an optimal policy, and later choice observations may more accurately reflect the traditional model's assumptions, it is also entirely possible that lack of proper experimentation reinforces a sub-optimal policy. Thus, as was shown, estimating structural parameters based off Q-learning data may lead to inaccurate

estimates, and this should be a concern if observations have the individual re-visiting similar areas of the state space. Marginal likelihood analysis comparing the two models would then give an indication of the possible biases. To my knowledge, this represents the first model that can be used to test for the presence and significance of policy learning effects in dynamic choice data.

Conclusion: Q-Learning as an Economic Model of Behavior

This paper has shown that the Q-Learning model serves not only as a behavioral model to explore policy learning, but also serves as a valid structural estimation model for DDC problems. In addition to showing it is a useful theoretical model that can be estimated, I argue that policy learning (Q-Learning or otherwise) is something that belongs in the field of economics. One reason has already been demonstrated: If policy learning effects are important to an individual, current structural models may fail at accurately estimating model parameters. It was shown that if learning is something that is a concern, the Q-learning model can be used to test for the significance of learning effects. Beyond this, though, there are more interesting behavioral questions that arise with regards to the policy learning.

Structural models attempt to estimate parameters that govern preferences and expectations. A large literature exists within the behavioral economics literature that investigates relationships between these parameters. For example, the relationship between risk aversion and the discount rate has been investigated in several different ways (Dean and Ortoleva 2012, Andreoni and Sprenger 2012). Q-learning, or policy learning in general, offers new parameters that may be of interest: the learning rate α, and the expectation parameters ω_A and ρ. Rates of learning are not necessarily

new to economics, but this specific parameter is different, measuring how quickly you adjust beliefs based on new versus past information.

Besides introducing new parameters, policy learning introduces a wide array of interesting questions that may be of importance to researchers. Are there any relationships between the new parameters? Are individuals with lower discount rates more likely to have higher/lower learning rates? An interesting question might be about the relationship between risk aversion and optimal policy formation, specifically: Are more risk averse individual's more likely to learn bad policies? That is, risk aversion may lead to a lower willingness to explore the state space, which in turn may lead to learning of worse policies. Since models of dynamic choice such as the dynamic probit model are often used in macroeconomics, this is not only interesting on a micro level but on a macro level as well. For example, cross country differences in learning rates and state-space weights could be investigated. These are just a few examples of the breadth of research questions thinking about policy learning leads to.

Overall, this paper has demonstrated the importance of models like Q-learning to economics on both a theoretic and econometric level. Theoretically, it offers a new model of individual behavior with multiple facets that should be explored. Econometrically, it represents an alternative structural DDC model and sheds light on the need for concern regarding learning effects and DDC estimation. Taken together, these form a powerful argument for the incorporation of Q-learning in the field of economics.

APPENDIX A

PROOF THAT $A_T^A = 1 - A_T^B$ IS A STEADY STATE

Consider the situation involving only two strategies. Furthermore, suppose each ability follows the below deterministic ability specification:

$$a_{t+1}^s = a_t^s + [I_t(s_t = s)(\mu(G(a_t^s))) - (1 - I_t(s_t = s))\mu L(a_t^s)] \tag{A.1}$$

Which satisfies $G(1-a) = L(a)$, an example of such would be $G(a) = 1 - a$ and $L(a) = a$. Let $a_t \equiv a_t^A$ and $b_t \equiv a_t^B$. Define $x_t = a_t + b_t$. To simplify notation, let $I_t = I_t(s_t = A)$ denote an indicator for if strategy A was chosen in period t. Clearly, $x_{t+1} = a_{t+1} + b_{t+1}$ by definition. Substituting in the above specification leads to:

$$\begin{aligned} x_{t+1} &= a_t + b_t - \mu(L(a_t) + L(b_t)) + \mu(G(b_t) + L(b_t)) + I_t\mu(G(a_t) + L(a_t) - G(b_t) - L(b_t)) \\ &= x_t + \mu[G(b_t) - L(a_t) + I_t(G(a_t) + L(a_t) - G(b_t) - L(b_t))] \end{aligned} \tag{A.2}$$

If $a_t = 1 - b_t$, which implies $x_t = 1$, then $G(a_t) = G(1 - b_t) = L(b_t)$ and likewise $G(b_t) = L(a_t)$. Thus, if $a_t = 1 - b_t$, it is the case that $x_{t+1} = x_t$. I.e. $x_t = 1$ is a steady state of system for x_t. Therefore, for any such ability specification, the assumption that $a_t = 1 - b_t$ is exactly the same as simply assuming the initial conditions are such that $a_0 + b_0 = 1$.

APPENDIX B

ABILITY EXAMPLES

While the particular situation being modeled will dictate exactly how ability evolves, it will still be useful to describe some early general ways to model ability. For this set of examples, I utilize functional forms that essentially normalize ability to be on the interval $[0, 1]$. However, this range is arbitrary, and the numerical examples in Section V have ability measured on the interval $[0, 100]$. As was stated earlier, there are many possibilities for the evolution of abilities and I focus on the basic case where it only depends on a player's own action history, briefly discussing other extensions to ability evolution in Section VI. While these example specifications may seem somewhat ad hoc, they exhibit features already prevalent within current learning models, such as diminishing returns to learning or S-shaped learning curves.

First, consider the very simple case where ability does not depreciate. Letting h_{it}^s denote the number of times strategy s has been chosen by player i, ability could be described by the following relationship:

$$a_{it}^s = \frac{(\lambda h_{it}^s)^n}{(\lambda h_{it}^s)^n + 1} \tag{B.1}$$

In this case, n simply modifies the curvature and λ controls how quickly one gains experience. The basic idea behind this is the shape of the $\frac{x^n}{x^n+1}$ graph. This function is everywhere increasing in x, and also features a change in curvature. For all values of n, the individual will experience dimishing returns to learning after a certain point, and always in the case of $n = 1$. The shape is such that experience builds slowly, then rapidly increases, and then tapers off as shown in the below figure:

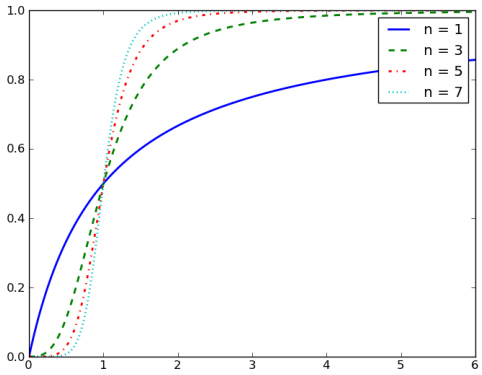


FIGURE 41. Graph of $\frac{x^n}{x^n+1}$

A more intricate specification might allow ability to depend on time as well. Let $a^s_{it} = \frac{\gamma^s_{it}}{\Psi_{it}}$ where Ψ and γ evolve according to the below specification, which has exogenous parameters λ_s, β, Ψ_{i0}, and γ^s_{i0}:

$$\Psi_{it+1} = \Psi_{it} + \beta \Rightarrow \Psi_{it} = \Psi_{i0} + \beta t \tag{B.2}$$

$$\gamma^s_{it+1} = \gamma^s_{it} + \lambda_s I_t(s) \Rightarrow \gamma^s_{it+1} = \gamma^s_{i0} + \lambda_s h^s_{it} \tag{B.3}$$

Again, $I_t(s)$ is an indicator for whether strategy s was chosen in period t and it is assumed that $\gamma^s_{i0} \leq \Psi_{i0}$ and $\lambda_s < \beta \ \forall s$. In this specification, λ_s, β and the initial values modify exactly how quickly a player will gain ability. By making ability dependent on time, this specification exhibits an "old dog, new tricks" feature which has players gaining ability more easily in earlier periods as opposed to later periods.

Clearly, in a manner similar to the last few specifications, any method of measuring the relative frequency of strategies suggested in the FD literature could be implemented for measuring ability. For example, setting $\gamma^s_{it+1} = \gamma^s_{i0} + h^s_{it}$ creates an environment where ability approaches relative frequency over time. More explicitly, it

could actually just be the case that ability equals relative frequency. Thus, it should be clear that the EBA framework easily nests FD games.

APPENDIX C

THE PARTICLE FILTER

Suppose only data on λ_t is observed over time. A particle filter can be run to approximate the likelihood of observing that data given a set of model parameters. That is, given parameters θ, the particle filter approximates $P(\lambda|\theta)$

In my estimation examples, I assume that only the population share, λ_t, is observed. For any set of parameters, e.g. μ_A,μ_B,and σ, particle filtering offers a way to obtain an estimate of $P((\lambda_1, ..., \lambda_T)|\mu_A, \mu_B, \sigma)$. In implementing the particle filter, I follow the procedure described in Fernandez-Villaverde and Rubio-Ramirez (2004, 2007).

Step 1: Draw N samples of a_0 and b_0 from their respective prior distributions. Denote the n-th draw as $a_0^{*|n}$ and $b_0^{*|n}$.

Step 2: For each draw of the above draws, calculate τ_1 using λ_0, λ_1, $a_0^{*|n}$, $b_0^{*|n}$, and the transition equation for λ. Call this τ_1^n

Step 3: For each τ_1^n calculate its likelihood based on the known distribution of the disturbance terms. The average value across all N draws will yield an approximation of $P(\lambda_1|\lambda_0)$.

Step 4: Re-sample (with replacement) N new draws from our initial draws, $a_0^{*|n}$ and $b_0^{*|n}$, using the likelihoods found in the last step as weights. Call each of these re-sampled draws a_0^n and b_0^n.

Step 5: Draw N samples of a_1 and b_1 according to the transition equations and the N draws, a_0^n and b_0^n from the previous step. Call these draws $a_1^{*|n}$ and $b_1^{*|n}$.

Step 6: Each individual draw of the state, a_1 and b_1, combined with the observed value of λ_1, allow us to find τ_2, from which we can calculate the likelihood of observing

that τ; i.e. we know τ comes from a truncated normal distribution that we know. Doing this for each of our N-draws of $a_1^{*|n}$ and $b_1^{*|n}$, the average likelihood value will give us an approximation of $p(\lambda_2|\lambda_1)$.

Step 7: Re-sample (with replacement) from our initial N draws of $a_1^{*|n}$ and $b_1^{*|n}$, using the likelihoods found in the last step as weights. Call these re-sampled draws a_1^n and b_1^n.

Step 8: Use the N re-sampled draws a_1^n and b_1^n, and the observed λ_1 to draw $a_2^{*|n}$ and $b_2^{*|n}$.

Step 9: Repeat steps 6 - 8

Doing the above will give us, sequentially, $p(\lambda_1|\lambda_0)p(\lambda_2|\lambda_1), p(\lambda_3|\lambda_2), ..., p(\lambda_T|\lambda_{T-1})$.

As N gets larger, the approximation gets closer and closer to the true value of $p(\lambda|\mu_A, \mu_B, \sigma)$.

APPENDIX D

MONOTONICITY PROOF

Suppose we have a standard 2x2 symmetric game shown below, where A,B,C,and D are all positive. Again, let λ_t denote the a-type population share at time t.

	a	b
a	(A, A)	(B, C)
b	(C, B)	(D, D)

TABLE 20. Monotonicity Stage Game

In this case, the average fitness of the a-type population is given by: $\lambda_t A + (1 - \lambda_t)B$, and similarly, the fitness of b-types is given by $\lambda_t C + (1 - \lambda_t)D$. In this case, the discrete replicator dynamic is given by:

$$\lambda_{t+1} = \lambda_t \frac{\alpha + \lambda_t A + (1 - \lambda_t)B}{\alpha + \lambda_t(\lambda_t A + (1 - \lambda_t)B) + (1 - \lambda_t)(\lambda_t C + (1 - \lambda_t)D)} \tag{D.1}$$

Obviously, if the only stable steady state is $\lambda_t = 0$ or $\lambda_t = 1$, the path of λ_t will be monotonic, and the system can't overshoot the steady state as λ_t is bound between 0 and 1. So the only concern would be the case of the existence of a mixed-population steady state. This would be a situation where average fitness of the a-types was exactly equal to the average fitness of the b-types. In fact, we can solve for the steady state as follows:

$$\lambda^* = \frac{D - B}{A - B - C + D} \tag{D.2}$$

In order for this mixed-population steady state to exist, λ^* must be positive and less than 1, which leads to the following two cases:

Stable Steady State: $B > D$ and $C > A$

Unstable Steady State: $D > B$ and $A > C$

In order for λ^* to be stable, it must be the case that $\lambda_t A + (1-\lambda_t)B < \lambda_t C + (1-\lambda_t)D$ whenever $\lambda_t > \lambda^*$. That is, if $\lambda_t = \lambda^* + \epsilon$, where $\epsilon \in (0, 1-\lambda^*)$, it must be the case that the b-types are doing better than the a-types in order to have λ_t decrease. In other words, it must be the case that:

$$(\lambda^* + \epsilon)A + (1 - \lambda^* - \epsilon)B < (\lambda^* + \epsilon)C + (1 - \lambda^* - \epsilon)D. \tag{D.3}$$

But notice that, by definition, $\lambda^* A + (1-\lambda^*)B = \lambda^* C + (1-\lambda^*)D$, so the above simplifies to: $(A-B) < (C-D)$. This requirement will be true if $B > D$ and $C > A$, and will not be true if $D > B$ and $A > C$.

Now consider $\lambda_t = \lambda^* + \epsilon$ again. Since it is clear that $\lambda_{t+1} < \lambda_t$, in order to prove monotonicity, it must be shown that $\lambda_{t+1} > \lambda^*$. For simplicity, let x represent the numerator of the RD, and y represent the denominator. The objective can then be restated as following. Show that the following relationship is true: $\lambda_{t+1} = (\lambda^* + \epsilon)(\frac{x}{y}) > \lambda^*$, or equivalently: $\lambda^*(\frac{y-x}{x}) < \epsilon$

In order to show this, the elements of the RD, x and y, must be determined. In order to simplify things, recall that $\lambda_t = \lambda^* + \epsilon$:

$$\begin{aligned} x =& \alpha + (\lambda^* + \epsilon)A + (1 - \lambda^* - \epsilon)B \\ y =& \alpha + (\lambda_t)((\lambda^* + \epsilon)A + (1 - \lambda^* - \epsilon)B) + (1 - \lambda_t)((\lambda^* + \epsilon)C + (1 - \lambda^* - \epsilon)D) \end{aligned} \tag{D.4}$$

First, rewrite y as:

$$y = \alpha+\lambda_t[(\lambda^*+\epsilon)A+(1-\lambda^*-\epsilon)B-(\lambda^*+\epsilon)C-(1-\lambda^*-\epsilon)D]+[(\lambda^*+\epsilon)C+(1-\lambda^*-\epsilon)D] \tag{D.5}$$

Now recall that $\lambda^*A + (1-\lambda^*)B = \lambda^*C + (1-\lambda^*)D$. Using this simplifies y to:

$$y = \alpha + \lambda_t\epsilon[A - B - C + D] + (\lambda^* + \epsilon)C + (1 - \lambda^* - \epsilon)D \tag{D.6}$$

Now consider $y - x$:

$$y-x = \alpha+\lambda_t\epsilon[A-B-C+D]+(\lambda^*+\epsilon)C+(1-\lambda^*-\epsilon)D-\alpha-(\lambda^*+\epsilon)A-(1-\lambda^*-\epsilon)B \tag{D.7}$$

Once again, $\lambda^*C + (1-\lambda^*)D$ will cancel with $\lambda^*A + (1-\lambda^*)B$, which leaves the following:

$$y - x = \epsilon(A - B - C + D)(\lambda_t - 1) \tag{D.8}$$

We can then write $\lambda^*(\frac{y-x}{x})$ as:

$$\lambda^*(\frac{y-x}{x}) = \lambda^*\frac{\epsilon(A - B - C + D)(\lambda_t - 1)}{\alpha + (\lambda^* + \epsilon)A + (1 - \lambda^* - \epsilon)B} \tag{D.9}$$

Now substitute $\lambda^* = \frac{D-B}{A-B-C+D}$, which yeilds:

$$\lambda^*(\frac{y-x}{x}) = \frac{\epsilon(D - B)(\lambda_t - 1)}{\alpha + (\lambda_t)A + (1 - \lambda_t)B} \tag{D.10}$$

Remember that our goal was to show whether or not $\lambda^*(\frac{y-x}{x}) < \epsilon$. What the above equation implies is that this will only be true if $\frac{(D-B)(\lambda_t-1)}{\alpha+(\lambda_t)A+(1-\lambda_t)B} < 1$, or:

$$(D-B)(\lambda_t-1) - \alpha - (\lambda_t)A - (1-\lambda_t)B < 0 \tag{D.11}$$

Clearly, for a large enough value of α, the above will always be true. But will it be true for any value of α? Suppose $\alpha = 0$:

$$(D-B)(\lambda_t-1) - (\lambda_t)A - (1-\lambda_t)B = D(\lambda_t-1) - (\lambda_t)A < 0 \tag{D.12}$$

Since $\lambda_t \in [0,1]$, and A,B,C and D are positive, it will always be the case that $D(\lambda_t-1) - (\lambda_t)A < 0$. Thus, it will always be the case that $\lambda^*(\frac{y-x}{x}) < \epsilon$ which implies that $\lambda_{t+1} > \lambda^*$. So, if $\lambda_t = \lambda^* + \epsilon$, $\lambda_{t+1} \in (\lambda^*, \lambda_t)$, which means the path of λ_t will be monotonically decreasing. In a similar fashion, one can show that if $\lambda_t = \lambda^* - \epsilon$ it will be the case that $\lambda_{t+1} \in (\lambda_t, \lambda^*)$.

To sum up, the above shows that even in discrete time, the path of λ_t will be monotonic in any 2x2 symmetric game; or equivalently, it shows that λ_t can never cross over a steady state λ^*.

APPENDIX E

3 STRATEGY RPS EXTENSION

Suppose we have a Rock-Paper-Scissors type game being played. At any one time, the proportion of each in the entire population can be observed. Call λ_t the proportion Rock, α_t the proportion Paper, and $1 - \lambda_t - \alpha_t$ the proportion Scissor. Each race, Rock, Paper, Scissors, now has 3 ability levels: How good am I against R, against P, and against S. Thus, we specify nine different ability levels, denoted as RR, RP, RS, PR, PP, PS, SR, SP, SS; where ij represents race i's ability at dealing with race j. Ability now evolves according to the following:

$$ij_{t+1} = p_{jt}(ij_t + \mu(\sqrt{ij_t} - ij_t)) + (1 - p_{it})(ij_t - \mu(ij_t - (1 - \sqrt{1 - ij_t}))) \qquad \text{(E.1)}$$

Where p_{jt} represents the proportion of the population that is type j. For example:

$$SP_{t+1} = \alpha_t(SP_t + \mu(\sqrt{SP_t} - ij_t)) + (1 - \alpha_t)(SP_t - \mu(SP_t - (1 - \sqrt{1 - SP_t}))) \quad \text{(E.2)}$$

In order to add stochasticity, we follow the previous setup and allow $ij_{t+1} \sim TN(m_{ijt}, \sigma, 0, 1)$ where $m_{ijt} = p_{jt}(ij_t + \mu(\sqrt{ij_t} - ij_t)) + (1 - p_{it})(ij_t - \mu(ij_t - (1 - \sqrt{1 - ij_t})))$.

It is important to note that here we track each ability individually, but the "Use it or Lose it" mechanism still exists even though we are not making any assumptions (like previously) that RR = 1 - RS - RP.

The evolution of the population makeup then proceeds as follows:

$$\begin{aligned}\lambda_{t+1} =& \lambda_t \frac{AvgFitR}{TotalAvgFit} + \tau_R = \bar{\lambda}_t + \tau_R \\ \alpha_{t+1} =& \alpha_t \frac{AvgFitP}{TotalAvgFit} + \tau_P = \bar{\alpha}_t + \tau_P\end{aligned} \tag{E.3}$$

In order to ensure that all values stay between 0 and 1, it is necessary to make restrictions on the error terms: $\tau_R \geq -\bar{\lambda}_t$, $\tau_P \geq -\bar{\alpha}_t$, and $\tau_R + \tau_P \leq 1 - \bar{\lambda}_t - \bar{\alpha}_t$. In other words, τ_R and τ_P are drawn from a joint-Truncated Multivariate Normal distribution:

$$\tau \sim TMVN(0, \Sigma, A\tau \leq B) \tag{E.4}$$

Where $A\tau \leq B$ is:

$$\begin{pmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \tau_R \\ \tau_P \end{pmatrix} \leq \begin{pmatrix} -\bar{\lambda}_t \\ -\bar{\alpha}_t \\ 1 - \bar{\lambda}_t - \bar{\alpha}_t \end{pmatrix} \tag{E.5}$$

and Σ is:

$$\begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix} \tag{E.6}$$

Estimation

Estimation proceeds in the same manner as before, however there are a few complications to running the particle filter. First off, instead of 2 unknown states, there are 9 in this model. Thus, for each time period, we must draw N samples of 9 states. Furthermore, calculating the likelihood is complicated by the TMVN error terms. That is, for each drawn particle in a period, t, we calculate $\bar{\lambda}_t$ and $\bar{\alpha}_t$

and using the actual data we recover τ_R and τ_P. To calculate the pdf values of this truncated distribution, we take $\phi(\tau)/\Phi$, where Φ is the total probability that τ lies within the area concern. $\phi(\tau)$ is calculated easily, as this is just the regular pdf of a MVN distribution. Φ on the other hand presents a challenge. We can either perform multiple integrations, or use Monte Carlo integration. I choose the latter, for the following reason.

Monte Carlo integration to find Φ works in the following way. At the start of each particle filter run, draw M random samples $\sim MVN(0, \Sigma)$. In my estimation M is typically set to 10,000. To find Φ we simply find the number of these samples which lie in our truncation region, call this C, and set $\Phi = \frac{C}{M}$. The most useful feature of this is that each run has the same Σ throughout, at every time period; and all share the same mean vector, $\mathbf{0}$. Thus, the sample only needs to be drawn once. For each particle, in each time period, we calculate λ_t^* and α_t^* which also defines our area of truncation. Thus, we can calculate every Φ value within a run via Monte Carlo integration but only have to draw samples once. While this still adds computation time, it does not do so prohibitively as numeric integration for every single particle in every time period would be infeasible.

Identification

One concern may be that as you increase the number of types in the population, the number of stage-game parameters to be estimated grows rapidly. This may mean more restrictions are needed to identify parameters of interest. However, this is only a concern if stage-game parameters are unknown. If the researcher just wants to estimate learning speeds, that can be done in the same manner. Moreover, initial simulations seem to indicate that identification does not become too much of a

	Rock	Paper	Scissors
Rock	2	1	3
Paper	3	2	1
Scissors	1	3	2

TABLE 21. Rock-Paper-Scissors

problem. For example, the following estimation recovered 10 parameters quite well after fixing only 5. Thus, one is still able to answer a wide range of questions regarding balance and relationships within the stage game.

No-Ability Example

Data was simulated from the below example, with no ability.

Similar techniques were used to recover the following posterior estimates of stage game coefficients, after assuming that the payoff to Rock vs Rock is 2. This estimation was markedly quicker because no particle filter was required:

	True	Mean	Mode
σ	0.01	0.0107	0.0105
RvP	1	0.9919	0.9643
RvS	3	2.8427	2.7386
PvR	3	3.0078	2.9959
PvP	2	1.9409	1.9008
PvS	1	0.9428	0.9090
SvR	1	0.9990	1.0049
SvP	3	2.9004	2.8741
SvS	2	1.9095	1.8404

TABLE 22. Posterior Summary - RPS

Thus, after fixing only one parameter, all of the other 8 were well identified. This seems to verify the earlier discussion regarding identification of the model.

	Rock	Paper	Scissors
Rock	$1 + RR$	$1 + 1.4RP - 0.8PR$	$1 + 1.5RS - 0.5SR$
Paper	$1 + 1.2PR - 0.2RP$	$1 + PP$	$1 + 2PS - 0.9SP$
Scissors	$1 + 2SR - 0.2RS$	$1 + 0.7SP - .5PS$	$1 + SS$

TABLE 23. RPS with Ability Stage Game

Full Model

The full model was run for the below stage game:

In this example, 80 time periods were simulated. 40,000 draws were obtained, after a 10,000 draw burn in. Lastly, 5 coefficients were assumed to be known. In order from Left to Right, Top to bottom, coefficients 1, 2, 4, 6, 11 were set to their true values. In the particle filter, N was set at 150, and M at 4000 (to find truncated pdf values via MC integration).

	True	Mean	Mode
σ	0.01	0.0106	0.0094
μ	0.15	0.2376	0.3102
P3	-.8	-0.6855	-.4436
P5	-.5	-.6775	-.5503
P7	-.2	-.3485	-.3462
P8	1	1.0885	1.1076
P9	2	1.8513	1.8277
P10	-.9	-.6748	-1
P12	-.2	-.4702	-.5462
P13	.7	0.6550	0.7677
P14	-.5	-.3952	-.4427
P15	1	1.2814	1.3365

TABLE 24. RPS-Ability Posterior Summary

Thus, even with a small amount of time periods, and relatively few draws, the procedure recovers the large number of parameters rather well. The same techniques can be extended to cases with more than 3 strategies, however, the number of stage-

game parameters will grow quickly, possibly limiting the number of parameters that can be identified, and increasing the number of parameters which must be assumed known by the researcher.

APPENDIX F

MARGINAL LIKELIHOOD CALCULATION

In general, the marginal likelihood can be found from Baye's Rule as:

$$P(Y) = \frac{P(Y|\theta P(\theta)}{P(\theta|Y)} \tag{F.1}$$

Since this is true for any θ, I calculated it for $\tilde{\theta} = median(\{\theta^{[g]}\})$.

The dynamic Probit models was estimated using a Gibbs sampler, and so the calculation of marginal likelihood values was done using the method of Chib (1995).

Calculating $P(Y|\tilde{\theta})$ is usually infeasible for latent variable models, and as such it is calculated via MC integration: $P(Y|\tilde{\theta}) = \int_{Y^*} P(Y|\tilde{\theta}, Y^*)P(Y^*|\tilde{\theta})dY^* = \frac{1}{J}\sum_{j=1}^{J} P(Y|\tilde{\theta}, Y^{*[j]})$, where $Y^{*[j]}$ represents draws of the latent variable conditioned on $\theta = \tilde{\theta}$. The method proposed Chib (1995) then breaks the posterior ordinate $P(\tilde{\theta}|Y)$ into $P(\tilde{\theta}_1|\tilde{\theta}_2, \tilde{\theta}_3, ..., \tilde{\theta}_K, Y)P(\tilde{\theta}_2|\tilde{\theta}_3, ...\tilde{\theta}_K, Y)...P(\tilde{\theta}_K|Y)$ where K is the number of blocks in the Gibbs sampler. These values can be calculated by generating J draws from "reduced runs", where appropriate blocks of θ are fixed to their respective $\tilde{\theta}$ value. It is important to note that the presence of latent data implies that $P(\tilde{\theta}_1|\tilde{\theta}_2, \tilde{\theta}_3, ..., \tilde{\theta}_K, Y)$ must also be calculated via MC integration, whereas normally it can just be found from the full conditional distribution. For a more detailed explanation, please refer to Chib (1995).

The BDP and Q-learning models utilized MH-algorithms, and so the method of Chib and Jeliaskov (2001) was utilized. This method recognizes that the posterior

ordinate can be expressed as:

$$P(\tilde{\theta}|Y) = \frac{\int_\theta \alpha(\theta, \tilde{\theta}) q(\tilde{\theta}|\theta) p(\theta|Y) d\theta}{\int_\theta \alpha(\tilde{\theta}, \theta) q(\theta|\tilde{\theta}) d\theta} \tag{F.2}$$

Where $\alpha(\cdot)$ is the acceptance probability and $q(\cdot)$ is the proposal density. The numerator of the above is the expected value of $\alpha(\theta, \tilde{\theta}) q(\tilde{\theta}|\theta)$ with respect to $p(\theta|Y)$. This can be estimated as:

$$\frac{1}{G} \sum_{g=1}^{G} \alpha(\theta^g, \tilde{\theta}) q(\tilde{\theta}|\theta^g) \tag{F.3}$$

Where θ^g is a draw from $P(\theta|Y)$, which we have stored from the MH-algorithm. The numerator is the expected value of $\alpha(\tilde{\theta}, \theta)$ with respect to the propoal density. This can be estimated as:

$$\frac{1}{J} \sum_{j=1}^{J} \alpha(\tilde{\theta}, \theta^j) \tag{F.4}$$

Where θ^j is a draw from $q(\theta^j|\tilde{\theta})$. In both cases, I set J = G + 1000. This step adds significant computation time as the likelihoods must be calculated for each proposal θ^j.

APPENDIX G

BAYESIAN DP ESTIMATION

Recently, Imai and Jain (2009) developed the Bayesian DP algorithm that significantly decreases the computation time for estimating these models. The Bayesian DP algorithm starts with an initial guess of the value function and works exactly like an MH-algorithm, but at each step one iteration of the Bellman operator is conducted, updating the value function.

This paper focused on the case of a continuous state space, but as Imai and Jain (2009) point out, the Bayesian DP algorithm easily applies to the random grid approximation method of Rust (1997). The MH-Algorithm depends on the calculation of the likelihood for a set of parameters, and this calculation requires a value function. The Bayesian DP algorithm in a continuous state calculates these as follows. At each step g, a proposal is drawn θ^{*g}, a shock ϵ^g is drawn, and a point in state space is randomly drawn $\mathbf{s}^g$.[1] Given a history of past proposals, shocks, states, and values $H^g = \{\theta^{*n}, \epsilon^n, \mathbf{s}^n, V^n(\mathbf{s}^n, \epsilon^n, \theta^{*n}, H^n)\}_{n=1}^g$, the value of a particular choice is calculated as:

$$\begin{aligned}
\bar{V}(\mathbf{s}, \epsilon, c, \theta, H^g) =& P(\mathbf{s}, c, \theta) + \beta \hat{V}(\mathbf{s}, c, \theta, H^g) + \epsilon_c \\
\hat{V}(\mathbf{s}, c, \theta, H^g) =& \sum_{n=1}^{N(g)} V^{g-n}(\mathbf{s}^{g-n}, \epsilon^{g-n}, \theta^{*g-n}, H^{g-n}) \frac{K(\theta - \theta^{*g-n}) f(\mathbf{s}^{g-n}|\mathbf{s}, c, \theta)}{\sum_{n=1}^{N(g)} K(\theta - \theta^{*g-n}) f(\mathbf{s}^{g-n}|\mathbf{s}, c, \theta)} \\
V^g(\mathbf{s}^g, \epsilon^g, \theta^{*g}, H^g) =& \max_{c \in C} \bar{V}(\mathbf{s}^g, \epsilon^g, c, \theta^{*g}, H^g)
\end{aligned} \tag{G.1}$$

[1] More than one could be drawn, but as Imai and Jain (2009) point out, when applying the BDP algorithm to the random grid approximation method of Rust (1997), only one is needed.

Where $N(g)$ is some increasing function of g, but it is required that $N(g) \to \infty$ as $g \to \infty$ and $g - N(g) \to \infty$ as $g \to \infty$, and K is some kernel weight function. In the above, previous values are weighted by how far their corresponding θ^* is from the current parameter vector θ. Suppose we observe $Y_t = \{c_t = A, s_t, s_{t+1}, R_t\}$. The rest of the paper uses a binary choice example, in which the individual must choose A or B each period. Furthermore, I assume the state transition has the following form: $\mathbf{s}_{t+1} = F(\mathbf{s}_t, c_t) + \mathbf{u}_t$, where $\mathbf{u}_t \sim MVN(\mathbf{0}, \Sigma_{AB})$. The choice probability at iteration g would be calculated as:

$$Pr(c_t|\theta, R_t, \mathbf{s}_t, H^g) = \Phi(R_t + \beta\hat{V}(\mathbf{s}, A, \theta, H^g) - P(\mathbf{s}_t, B, \theta) - \beta\hat{V}(\mathbf{s}, B, \theta, H^g), \sigma_\epsilon) \quad \text{(G.2)}$$

Where Φ denotes a normal CDF with standard deviation σ. Then, the complete likelihood can be stated as:

$$L(Y|\theta) = \prod_{t=1}^{T} Pr(c_t|\theta, R_t, \mathbf{s}_t, H^g)\phi(R_t - P(\mathbf{s}_t, c_t, \theta), 0, \sigma_\epsilon^2)\phi(\mathbf{s}_{t+1}, F(\mathbf{s}_t), \Sigma_{AB}) \quad \text{(G.3)}$$

Where $\phi(\mathbf{x}, \mu, \Sigma)$ denotes a MVN pdf with mean vector μ and variance-covariance matrix Σ. In the above, $\Sigma_{AB} = \begin{bmatrix} \sigma_A^2 & 0 \\ 0 & \sigma_B^2 \end{bmatrix}$. Note that, as Equation (6) indicates, one iteration of the Bellman operator is performed at each step:

$$V^g(\mathbf{s}^g, \epsilon^g, \theta^{*g}, H^g) = \max_{c \in C} \bar{V}(\mathbf{s}^g, \epsilon^g, c, \theta^{*g}, H^g) \quad \text{(G.4)}$$

This final step is key in the Bayesian DP algorithm, and in this way the algorithm solves the DP problem and estimates the parameters simultaneously. Imai and Jain (2009) also note that as the number of draws increases, the accuracy of the algorithm increases. However, this increased accuracy also requires increased computational

time because as the number of draws increases, the computation time of each step will increase as $N(g)$ will be increasing also. Overall, though, this algorithm represents one of the most successful and efficient ways to estimate the classic structural model.

APPENDIX H

PAYOFF FUNCTION PICTURES

The payoff function for the individual depends on which action is chosen and is listed below:

$$P(c_t, a_t, b_t) = \begin{cases} 30 + 10a_t - 80((a_t - H_A)^2 + (b_t - H_B)^2) - \frac{3}{.1+\sqrt{(a_t-H_A)^2+(b_t-H_B)^2}} & \text{if} c_t = A \\ 30 + 10b_t - 80((a_t - H_A)^2 + (b_t - H_B)^2) - \frac{3}{.1+\sqrt{(a_t-H_A)^2+(b_t-H_B)^2}} & \text{if} c_t = B \end{cases} \tag{H.1}$$

In helping visualize, the below figures show two angles of the B-payoff function when $H_A = H_B = .5$, and an overlay of the B and A Payoffs demonstrating their differences:

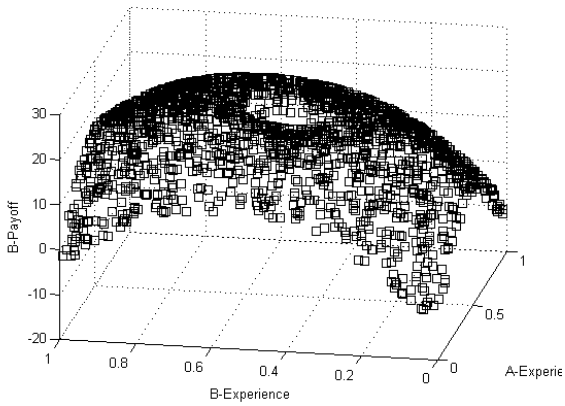


FIGURE 42. B-Payoff Angle 1

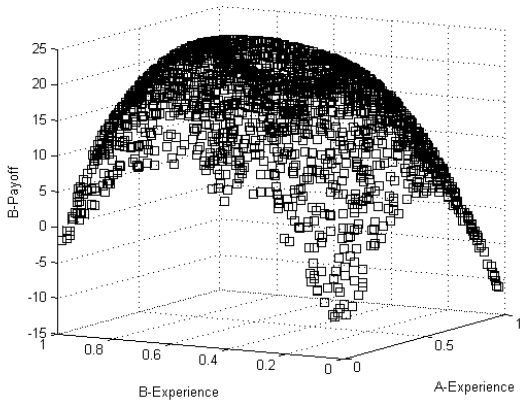


FIGURE 43. B-Payoff Angle 2

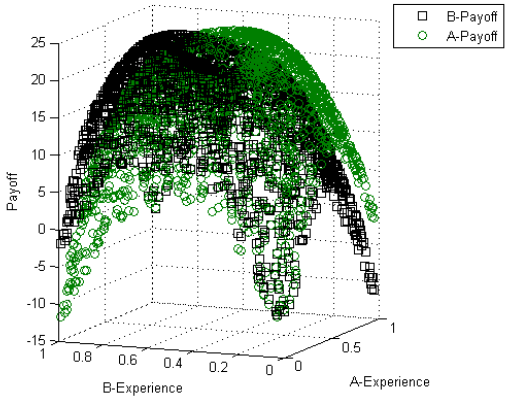


FIGURE 44. Payoff Function Overlay

APPENDIX I

ALTERNATIVE CHOICE RULE

In DDC estimation, the logit choice rule is commonly used. However, it is usually assumed that the individual is actually picking the option with the highest value, but these values are hit with type I extreme value shocks unobserved by the researcher. Normally the distinction is not critical to the estimation process. However, if shocks are incorporated into Q-learning, estimation must proceed differently because these shocks would carry over. Specifically, consider the following change to the Q-learning model:

$$Q_t(\mathbf{s_t}, \mathbf{c}) = \widetilde{\mathbf{Q}}(\mathbf{s_t}, \mathbf{c}) + \mathbf{u_{ct}} \tag{I.1}$$

Where u_{ct} is distributed type I extreme value. Now, given $\widetilde{Q}(\mathbf{s}_t, A)$ and $\widetilde{Q}(\mathbf{s}_t, B)$, we can state that the probability A is chosen is:

$$Pr(c_t = A) = \frac{\exp(\widetilde{Q}(\mathbf{s}_t, A))}{\exp(\widetilde{Q}(\mathbf{s}_t, A)) + \exp(\widetilde{Q}(\mathbf{s}_t, B))} \tag{I.2}$$

This is just like before. However, after making the choice, the individual must update his Q-values. If the individual was to ignore the unobserved shock when updating, we would have the exact same model presented earlier. However, this may not make sense. The unobserved shock was something important enough to the individual to cause them to choose one option over another. So it may be more reasonable to assume the individual uses their actual Q-value, $Q_t(\mathbf{s_t}, \mathbf{c})$, when updating. That is, the update process is now:

$$Q_{t+1}(\mathbf{s},c)=\begin{cases}(1-\alpha)(\widetilde{Q}_t(\mathbf{s},c)+u_{ct})+\alpha(R_t+\beta\max\limits_{c'\in C}Q_t(\mathbf{s}_{t+1},c')) & \text{if } c=c_t \text{ and } \mathbf{s}=\mathbf{s}_t\\ \widetilde{Q}_t(\mathbf{s},c) & \text{else}\end{cases} \tag{I.3}$$

That is, when the individual adds an item to their set of experiences, they use the udpated value including the unobserved shock. This means that, given the correct paramter values and observations on choices and payoffs, the researcher will be unable to perfectly reconstruct the series of Q-values. Moreover, these unobserved values play a role in determining every other Value in the future if the state space is continuous. Thus, the estimation procedure used earlier would not be valid. However, one can utilize a particle filter to approximate the likelihood in the following way.

Start with initial values, q_0^A and q_0^B. These may or may not be paramters that are being estimated. Create N samples of $Q_1(s_1,A)$ and $Q_1(s_1)$ by adding simulated shocks from a type I extreme value distribution to the initial values. Call these samples $\hat{Q}_1^n(s_1,c)$. Based on the observed choice, the likelihood of observing that choice can be approximated as:

$$\hat{P}(c_1=c)=\frac{\sum_{n=1}^{N}\hat{Q}_1^n(s_1,c_1)>\hat{Q}_1^n(s_1,-c_1)}{N} \tag{I.4}$$

Where $-c_t$ indicates the option that was not chosen by the individual. From the set of samples that satisfy $\hat{Q}_1^n(s_1,c_1)>\hat{Q}_1^n(s_1,-c_1)$, draw with replacement N new samples of Q-values for the observed choice. Using the observation on payoffs, and the current parameter values, update these N Q-values in the usual way, and denote them $Q_1^{n*}(s_1,c_1)$. Stack these in a column vector called $\mathbf{Q}_1^*$; this is the sample of past experience for the individual.

Now, consider time period t, given N samples of past experiences $\mathbf{Q}^*_{t-1}$. For each of the N samples, use the current parameter values to form the expected Q-value for each option $\widetilde{Q}^n_t(\mathbf{s}_t, c)$. Then add a shock to each value forming $\hat{Q}^n_t(\mathbf{s}_t, c) = \widetilde{Q}^n_t(\mathbf{s}_t, c) + u_{ct}$. Again, the probability of observing choice c_t can be approximated by:

$$\hat{P}(c_t = c) = \frac{\sum_{n=1}^{N} \hat{Q}^n_t(\mathbf{s}_t, c_t) > \hat{Q}^n_t(\mathbf{s}_t, -c_t)}{N} \tag{I.5}$$

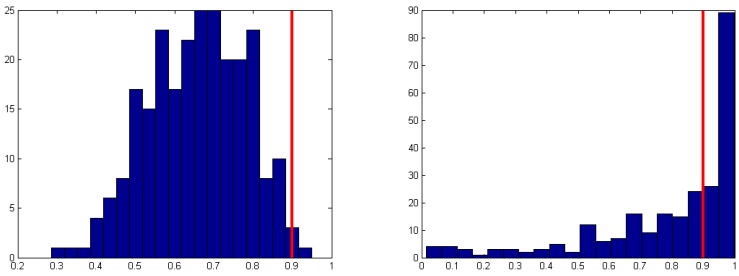
Now, re-sample N times with replacement from the set of past experiences that generated values satisfying $\hat{Q}^n_t(\mathbf{s}_t, c_t) > \hat{Q}^n_t(\mathbf{s}_t, -c_t)$. Using the observations on payoffs, and the current parameter values, update each of the N $\hat{Q}$ values corresponding to the re-sampled histories. Add these updated values to the re-sampled histories to form $\mathbf{Q}^*_t$.

This process is similar to standard particle filters, but at each re-sample stage the entire history of updated Q-values has to be re-sampled, not just the current Q-value. Again, this is because the individual's expectation is formed using all these values. The approximate likelihood value can be formed as $\prod_{t=1}^{T} \hat{P}(c_t = c)$ and then used in a Metropolis-Hastings Algorithm.

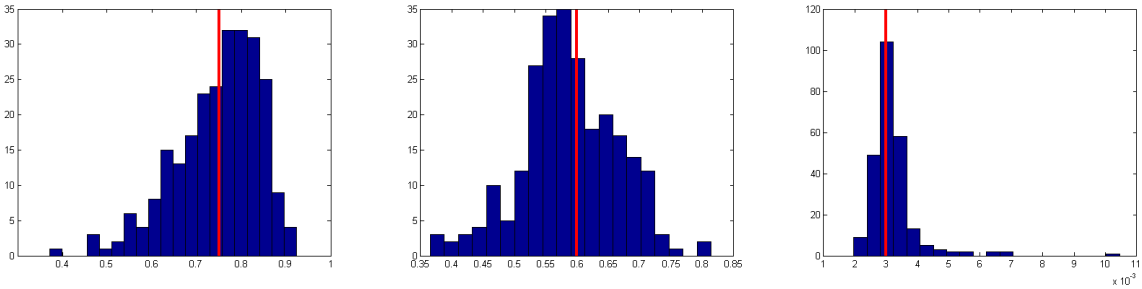
In a manner similar to that done previously, 250 data sets were simulated from individuals using this Q-learning process. The estimation technique described above was carried out. All parameter values and priors were exactly the same as previously stated. The computation time was substantially longer, and was now about the same speed as the BDP estimation procedure. Histograms of posterior means and standard deviations are shown below. Again, the estimation performs quite well, although it is not as accurate as the original model; especially in the posteriors for β. Of course, this is to be expected since there is much more uncertainty in this model. It is also still the case that marginal likelihood analysis consistently selects the Q-model as the

most appropriate. Indeed, in all 250 data sets, the BDP model never had a higher marginal likelihood value than the Q-learning model, and the dynamic Probit was higher than the Q-learning model only twice.

Further research into this distinctive model might include investigating the differences between the original model, and how these differences change as the parameter values change. Specifically, I would expect the importance of the Q-value shocks to change as the updating parameter α changes. The intricacies of the model differences also apply to standard approach taken in the experimental literature, where the Logit rule is typically assumed. It may be that there are substantial differences in results regarding learning between these two models, and this is an area that should be explored more.



(a) β (b) $mode - \beta$



(c) α (d) ω_A (e) ρ

FIGURE 45. Monte Carlo Experiment - Alternative Q-model

REFERENCES CITED

[1] Albert, J. and Chib, S. . Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

[2] Andreoni, J. and Spregner, C. . Risk preferences are not time preferences. *The American Economic Review*, 102(7):3357–3376, 2012.

[3] Archetti, M. . The origin of autumn colours by coevolution. *Journal of Theoretical Biology*, 205:625–630, 2000.

[4] Arcidiacono, S. H. , P. and Sloan, F. . Living rationally under the volcano? an empirical analysis of heavy drinking and smoking. *International Economic Review*, (1):37–65, 2007.

[5] Argasinski, K. and Broom, M. . Ecological theatre and the evolutionary game: How environmental and demographic factors determine payoffs in evolutionary games. *Journal of Mathematical Biology*, pages 1–28, 2012.

[6] Atkeson, C. , Moore, A. , and Schaal, S. . Locally weighted learning for control. *Artificial Intelligence Review*, 11:75–113, 1999.

[7] Axelrod, R. and Hamilton, W. . The evolution of cooperation. *Science*, 211(4489):1390–1396, 1981.

[8] Brenner, T. and Witt, U. . Melioration learning in games with constant and frequency-dependent payoffs. *Journal of Economic Behavior and Organization*, 50:429–448, 2003.

[9] Brock, W. and Hommes, C. . Heterogeneous beliefs and routes to chaos in a simple asset pricing model. *Journal of Economic Dynamics and Control*, 22(8):1235–1274, 1998.

[10] Brockwell, A. . Parallel markov chain monte carlo simulation by pre-fetching. *Journal of Computational and Graphical Statistics*, 15(1):246–261, 2006.

[11] Cameron, S. and Heckman, J. . Life cycle schooling and dynamic selection bias: models and evidence for five cohorts of american males. *The Journal of Political Economy*, 106(2):262 – 333, 1998.

[12] Carro, J. and Mira, P. . A dynamic model of contraceptive choice of spanish couples. *Journal of Applied Econometrics*, 21(7):955 – 980, 2006.

[13] Chakrabarti, S. . Markov equilibria in discounted stochastic games. *Journal of Economic Theory*, 85:294–327, 1999.

[14] Chauvet, M. and Potter, S. . Forecasting recessions using the yield curve. *Journal of Forecasting*, 24:77–103, 2005.

[15] Ching, I. S. I. M. , A. and Jain, N. . A practitioner's guide to bayesian estimation of discrete choice dynamic programming models. *Quantitative Marketing and Economics*, 10(2):151–196, 2012.

[16] Clark, C. . *Mathematical Bioeconomics*. John Wiley & Sons Inc., New Jersey, 2010.

[17] Collard-Wexler, A. . Productivity dispersion and plant selection in the ready-mix concrete industry. *US Census Bureau Center for Economic Studies Paper*, 2011.

[18] Conrad, J. and Clark, C. . *Natural Resource Economics*. Cambridge University Press, New York, 1987.

[19] Dasgupta, P. and Stiglitz, J. . Learning-by-doing, market structure and industrial and trade policies. *Oxford Economic Papers*, 40(2):246–268, 1988.

[20] Dean, M. and Ortoleva, P. . Estimating the relationship between economic preferences: A testing ground for unified theories of behavior. *mimeo*, 2012.

[21] Dick, A. . Learning by doing and dumping in the semiconductor industry. *Journal of Law and Economics*, 34:133–59, 1991.

[22] Dutta, P. and Sundaram, R. . Markovian equilibrium in a class of stochastic games: Existence theorems for discounted and undiscounted models. *Economic Theory*, 2:197–214, 1992.

[23] Dutta, P. . A folk theorem for stochastic games. *Journal of Economic Theory*, 66:1–32, 1995.

[24] Fernandez-Villaverde, J. and Rubio-Ramirez, J. . Sequential monte carlo filtering: An example. *University of Pennsylvania*, 2004.

[25] Fernandez-Villaverde, J. and Rubio-Ramirez, J. . Estimating macroeconomic models: A likelihood approach. *The Review of Economic Studies*, 74:1059–1087, 2007.

[26] Forbes, J. and Andre, D. . Practical reinforcement learning in continuous domains. *University of California, Berkeley*, 2000.

[27] Fossati, S. . Dating u.s. business cycles with macro factors. *mimeo*, 2011.

[28] Foster, D. and Young, P. . Stochastic evolutionary game dynamics. *Thoeretical Population Biology*, 38(2):219–232, 1990.

[29] Fudenberg, D. and Tirole, J. . Learning-by-doing and market performance. *The Bell Journal of Economics*, 14(2):522–530, 1983.

[30] Gaskett, W. D. , C. and Zelinsky, A. . Q-learning in continuous state and action spaces. *Australian Joint Conference on Artificial Intelligence*, pages 417–428, 2003.

[31] Gatenby, R. and Vincent, T. . Application of quantitative models from population biology and evolutionary game theory to tumor therapeutic strategies. *Molecular Cancer Therapeutics*, 2(9):919–927, 2003.

[32] Gatenby, R. and Vincent, T. . An evolutionary model of carcinogenesis. *The Journal of Cancer Research*, 63(19):6212–6220, 2003.

[33] Gruber, H. . The learning curve in production of semiconductor memory chips. *Applied Economics*, 24:885–894, 1992.

[34] Gruber, H. . Learning by doing and spillovers: Further evidence for the semiconductor industry. *Review of Industrial Organization*, 13(6):697–711, 1998.

[35] Hatch, N. and Reichelstein, S. . Learning effects in semiconductor fabrication. *mimeo*, 1995.

[36] Haveman, R. . Common property, congestion, and environmental pollution. *Quarterly Journal of Economics*, 87(2):278–287, 1973.

[37] Hofbauer, J. and Sigmund, K. . Evolutionary game dynamics. *Bulletin of the American Mathematical Society*, 40(4):479–519, 2003.

[38] Horst, U. . Stationary equilibria in discounted stochastic games with weakly interacting players. *Games and Economic Behavior*, 51:83–108, 2005.

[39] Hotz, V. and Miller, R. . Conditional choice probabilities and the estimation of dynamic models. *Bulletin of the American Mathematical Society*, 60(3):497–529, 1993.

[40] Imai, J. N. , S. and Ching, A. . Bayesian estimation of dynamic discrete choice models. *Econometrica*, 77(6):1865–1899, 2009.

[41] Imai, S. and Keane, M. . Intertemporal labor supply and human capital accumulation. *International Economic Review*, 45(2):601–641, 1998.

[42] Irwin, D. and Klenow, P. . Learning-by-doing spillovers in the semiconductor industry. *Journal of Political Economy*, 102(6):1200–1227, 1994.

[43] Joosten, R. , Brenner, T. , and U., W. . Games with frequency-dependent stage payoffs. *International Journal of Game Thoery*, 31:609–620, 2003.

[44] Jovanovic, B. and Nyarko, Y. . Learning-by-doing and the choice of technology. *Econometrica*, 64(6):1299–1310, 1996.

[45] Kandori, M. G. , M. and Rob, R. . Learning, mutation, and long run equilibria in games. *Econometrica*, 61(1):29–56, 1993.

[46] Keane, T. P. , T. and Wolpin, K. . The structural estimation of behavioral models: Discrete dynamic choice dynamic programming methods and applications. *Handbook of Labor Economics*, 4:331–461, 2011.

[47] Klemperer, P. and Meyer, M. . Price competition vs. quantity competition: The role of uncertainty. *Rand Journal of Economics*, 17(4):618–638, 1998.

[48] Klenow, P. . Learning curves and the cyclical behavior of manufacturing industries. *Review of Economic Dynamics*, 1(2):531–550, 1998.

[49] Kolter, R. and Vulic, M. . Evolutionary cheating in escherichia coli stationary phase cultures. *Genetics*, 158(2):519–526, 2001.

[50] Martín-Herrán, G. and Rincón-Zapatero, J. . Efficient markov perfect nash equilibria: Theory and application to dynamic fishery games. *Journal of Economic Dynamics and Control*, 29:1073–1096, 2005.

[51] Maynard Smith, J. . The theory of games and the evolution of animal conflicts. *Journal of Theoretical Biology*, 47(1):209–221, 1974.

[52] Nowak, M. and Sigmund, K. . Evolutionary dynamics of biological games. *Science*, 303:793–799, 2004.

[53] Paap, R. and Hans Franses, P. . A dynamic multinomial probit model for brand choice with different long-run and short-run effects of marketing-mix variables. *Journal of Applied Eeconometrics*, 15:717–744, 2000.

[54] Pakes, A. and McGuire, P. . Stochastic algorithms, symmetric markov perfect equilibrium, and the curse of dimensionality. *Econometrica*, 69(5):1261–1281, 2001.

[55] Perc, M. and Szolnoki, A. . Coevolutionary games - a mini review. *Biosystems*, 99:109–125, 2000.

[56] Rust, J. . Optimal replacement of gmc bus engines: An empirical model of harold zurcher. *Econometrica*, 55(5):999–1033, 1987.

[57] Rust, J. . Using randomization to break the curse of dimensionality. *Econometrica*, 65(3):487–516, 1997.

[58] Ryan, P. C. , M. and Morris, M. . A genetic polymorphism in the swordtail xiphophorus nigrensis: testing the prediction of equal fitnesses. *American Naturalist*, 139(1):21–31, 1992.

[59] Siddhartha, C. and Jeliazkov, I. . Marginal likelihood from metropolis-hastings output. *Journal of the American Statistical Association*, 96(453):270–281, 2001.

[60] Siddhartha, C. . Marginal likelihood from the gibbs output. *Journal of the American Statistical Association*, 90(432):1313–1321, 1995.

[61] Sinervo, B. and Lively, C. . The rock-paper-scissors game and the evolution of alternative male strategies. *Nature*, 380:240–243, 1996.

[62] Sinervo, B. and Zamudlo, K. . Polygyny, mate-guarding, and posthumous fertilization as alternative male mating strategies. *Proceedings of the National Academy of Sciences*, 97(26):14427–14432, 2000.

[63] Sinervo, S. E. , B. and Comendant, T. . Density cycles and an offspring quantity and quality game driven by natural selection. *Nature*, 406:985–988, 2000.

[64] Singh, N. and Vives, X. . Price and quantity competition in a differentiated duopoly. *Rand Journal of Economics*, 15(4):79–100, 1984.

[65] Smart, W. and Kaelbling, L. . Practical reinforcement learning in continuous spaces. *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 903–910, 2000.

[66] Sorger, G. . Markov-perfect nash equilibria in a class of resource games. *Economic Theory*, 11:79–100, 1998.

[67] Spence, M. . The learning curve and competition. *The Bell Journal of Economics*, 12(1):49–70, 1981.

[68] Stinebrickner, T. . A dynamic model of teacher labor supply. *Journal of Labor Economics*, 19(1):196–230, 2001.

[69] Strid, I. . Efficient parallelisation of metropolis-hastings algorithms using a prefetching approach. *Computational Statistics & Data Analysis*, 51(11):2814–2835, 2010.

[70] Svensson, E. and Raberg, L. . Resistance and tolerance in animal enemy victim coevolution. *Trends in Ecology and Evolution*, 25(5):267–274, 2004.

[71] Sweeting, A. . Dynamic product repositioning in differentiated product markets: The case of format switching in the commercial radio industry. *National Bureau of Economic Research*, 2007.

[72] Taylor, P. and Jonker, L. . Evolutionarily stable strategies and game dynamics. *Mathematical Biosciences*, 40:145–156, 1978.

[73] Thompson, P. . Learning by doing. *Handbook in Economics*, 1:430–476, 2010.

[74] Waltman, L. and Kaymak, U. . A theoretical analysis of cooperative behavior in multi-agent q-learning. *Proceedings of the 2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pages 84–91, 2007.

[75] Waltman, L. and Kaymak, U. . Q-learning agents in a cournot oligopoly model. *Journal of Economic Dynamics and Control*, 32(10):3275–3293, 2008.

[76] Watkins, C. . *Learning from delayed rewarads*. PhD thesis.

[77] Watkins, C. and Dayan, P. . Technical note: Q-learning. *Machine Learning*, 8:279–292, 1992.

[78] Weintraub, L. , G. Benkard and Van Roy, B. . Oblivious equilibrium: A mean field approximation for large-scale dynamic games. *NIPS*, 2005.

[79] Zanchettin, P. . Differentiated duopoly with asymmetric costs. *Journal of Economics and Management Strategy*, 15(4):999–1015, 2006.

[80] Zeeman, E. . Population dynamics from game theory. *Global Theory of Dynamical Systems*, pages 471–497, 1980.