2004-11

The Ghost in the Machine: Inferring Machine-Based Strategies from Observed Behavior

ENGLE-WARNICK, Jim
McCAUSLAND, William J.
MILLER, John H.

Université de Montréal

Département de sciences économiques
Université de Montréal
Faculté des arts et des sciences
C.P. 6128, succursale Centre-Ville
Montréal (Québec) H3C 3J7
Canada
http://www.sceco.umontreal.ca
SCECO-information@UMontreal.CA
Téléphone : (514) 343-6539
Télécopieur : (514) 343-7221

Ce cahier a également été publié par le Centre interuniversitaire de recherche en économie quantitative (CIREQ) sous le numéro 15-2004.

This working paper was also published by the Center for Interuniversity Research in Quantitative Economics (CIREQ), under number 15-2004.

ISSN 0709-9231

The Ghost in the Machine: Inferring Machine-Based Strategies from Observed Behavior

Jim Engle-Warnick* William J. McCausland† John H. Miller‡

August 30, 2004

Abstract

We introduce a procedure to infer the repeated-game strategies that generate actions in experimental choice data. We apply the technique to set of experiments where human subjects play a repeated Prisoner's Dilemma. The technique suggests that two types of strategies underly the data.

Keywords: strategy inference, Bayesian methods, experimental economics, Prisoner's Dilemma, repeated games.

JEL classification nos. C11 , C15, C72, C73, C92

Acknowledgements: We thank Alexander Campbell, Julie Héroux, and Anthony Walsh for valuable research assistance. We gratefully acknowledge financial support from the Economic and Social Research Council, UK and Nuffield College, University of Oxford. The Centre for Interuniversity Research and Analysis on Organizations (CIRANO) and the Bell University Laboratory in Electronic Commerce and Experimental Economy graciously provided use of the experimental laboratory.

*McGill University, Department of Economics, 855 Sherbrooke St. W., Montreal, QC, H3A 2T7, Canada, email: jim.engle-warnick@mcgill.ca.

†Université de Montréal, Département de sciences économiques, 3150, rue Jean-Brillant, bureau C-6029 C.P. 6128, Succursale centre-ville Montreal, QC H3C-3J7, Canada, email: william.j.mccausland@umontreal.ca.

‡Carnegie Mellon University, Department of Social and Decision Sciences, Pittsburgh, PA 15213 and the Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, e-mail: miller@santafe.edu.

1 Introduction

Understanding strategic choice has become a key goal of modern economics. Over the last half century a large body of theoretical work has been developed in this area. More recently, experimental work has begun to illuminate the various behavioral tendencies of real strategic agents. Here we develop techniques designed to divine the underlying "program" that directs individual strategic behavior in repeated games. By capturing the computation inherent in actual strategic decision making, we should be able to improve our predictions of social system behavior, as well as provide new empirical and theoretical insights.

The Bayesian inference technique we develop here allows us to draw inferences from choice data regarding the number and characteristics of repeated-game strategies in a population of experimental subjects. Our 'machine"-based repeated-game strategy model is based on finite automata, which capture the computation inherent in strategic choice and represent a rich class of behavior for systems that rely on finite inputs and outputs. Our goal is to develop an empirically based model of the set of strategies that people actually use in repeated games. Since our procedure is applicable to a much wider class of games than we investigate here, it will open the door for researchers to supplement common tests of behavioral conformance to theories with deeper investigation of the strategic heterogeneity behind observed choices in many economically interesting contexts.

The econometric model is based on finite state machines. We make the model stochastic by making action probabilities, but not state transitions, random. The number of machines, their probabilities in the population, numbers of machine states, state transition functions and action probabilities (by machine state) are all variable: using repeated game data from the laboratory, we obtain the joint posterior distribution of all these unknown quantities. Following a large literature on data augmentation, introduced to the statistics literature by Tanner and Wong (1987), we add the function mapping experimental subjects to machines to the list of unknown quantities. Doing this simplifies posterior simulation greatly. In particular, it allows us to integrate out action probabilities, an enormous computational advantage. It also provides us with the posterior distribution (jointly with other unknowns) of the machine assignments, which is useful in its own right. Conditional simulation of transition functions is based partly on a new Metropolis-Hastings proposal that stochastically builds a state transition function based on the tree of supergame play generated by the subjects assigned to a given machine type. We introduce a useful way of identifying machines in a population using a labelling technique.

We apply our technique to a choice data set from a series of experiments in which human subjects play a repeated Prisoner's Dilemma. We use this data set to attempt to uncover the strategies driving human behavior in such a context. The results give us a new picture of play in repeated prisoner's dilemma games. We find strong evidence for heterogeneity. We find evidence that people use strategies that punish and reward behavior of the opponent. Interestingly, the strategies we infer do not contain harsh enough punishments to support cooperation in equilibrium.

Our procedure is closely related to the early work on identifying subject heterogeneity in experiments by El-Gamal and Grether (1995), who used a Bayesian procedure to estimate decision rules in a population of subjects faced with a static, individual choice task. Their goal was to discover whether or not people are Bayesian, and they did this convincingly by assuming a functional form for the decision rule and using their Bayesian technique to draw inference regarding the number and types of rules that generated their data. By contrast, our application is to repeated-game strategies, which requires both a very different strategy model and computational technique for inference.[1]

Houser, Keane, and McCabe (2004) provide a method to draw inference regarding the number and types of decision rules in a population of subjects playing a dynamic game.

[1] Other probabilistic choice models include Stahl and Wilson (1995) study heterogeneity in levels of reasoning in games solvable through iterated dominance specifying both the form and number of decision rules, and McKelvey and Palfrey (1992), who introduce Quantal Response Equilibrium, which makes it possible to study subject behavior in deviation from optimality (though not subject heterogeneity).

They estimate a polynomial approximation of the value function (that is, the expected value of future payoffs) in state variables similar to Geweke and Keane (1999, 2001), allowing individual decision rules to differ by the parameters in their value functions. They illustrate their technique with a game that subjects play against nature based on a model of school choice, and find evidence for a few interesting behavioral types. This approach is very flexible because the researcher does not have to specify *a priori* the functional form of the decision rules. One simulates the rules to interpret the behavior. Our application gains some efficiency (at the cost of flexibility) by specifying a strategy model. Our strategy model, however, is general for a rather wide class of multi-player games, covering strategies predicted by theory and simpler rules of thumb, and in the end, we are interested in characterizing the strategies.[2]

One can also address the question of strategy types through experimental design. Experimental economists have, for example, attempted to elicit strategies directly from subjects (see, for example, Selten *et al.* (1997)). Others identify strategies by tracking the manner in which subjects collect and process information, as was done by Costa-Gomes, Crawford, and Broseta (2001), and Johnson, Camerer, Sen, and Rymon (2001). Other alternatives include various experimental manipulations and the use of protocol responses. While all of these techniques can provide insight into strategic choice, they may be limited to particular games, require self-reports of behavior, or costly experimental design. As such they are important complements to the larger quest of understanding actual strategic behavior.

[2] Houser, Keane, and McCabe note that the two types of procedures can be complementary: one could use their procedure to identify subject types and then the El-Gamal and Grether procedure to describe the decision rules, and the same holds true for our procedure.

2 Games and Machines

Here we provide some formal definitions, and examples, that will provide the needed framework for the work that follows. We consider a world in which agents play a repeated stage game. We call a single instance of a repeated stage game a *supergame*. An agent's repeated-game strategy is embodied by a machine (a stochastic automaton), a representation that provides a compact description of a broad swath of potential strategies. We assume that each agent has access to a heterogeneous collection of such machines, and randomly chooses one machine when called upon to play.

Agents repeatedly play a *stage game*, γ, defined by the triple $(N, A, (u_i)_{i \in N})$. The set of *players* is given by N. For each player $i \in N$, they have a set A_i of potential actions. Let $A = \times_{i \in N} A_i$ give the *action profile set* (the set of potential actions) and let $a = (a_i)_{i \in N} \in A$ give the *action profile* (the set of actual actions chosen by each player during the stage game). The payoff function for each player $i \in N$, is given by $u_i \colon A \to \Re$. For example, a Prisoner's Dilemma stage game has $N = \{1, 2\}$, $A = A_1 \times A_2 = \{d, c\} \times \{D, C\}$ (where, we use lower case to indicate the actions of Player 1), and $u_1(d, D) = u_2(d, D) = P$ (punishment), $u_1(c, D) = u_2(d, C) = S$ (sucker), $u_1(d, C) = u_2(c, D) = T$ (temptation), and $u_1(c, C) = u_2(c, C) = R$ (reward).

Agents employ machines—represented by stochastic automata—to implement a given strategy. Aumann (1981) suggested that such machines would be a useful way to represent game strategies in economics. Automata model systems that generate discrete outputs in response to discrete inputs, and as such they represent a fundamental class of systems. Automata have been used to explore bounded rationality in repeated games (e.g., Rubinstein, 1986), evolutionary games (Binmore and Samuelson, 1992), and learning (Miller, 1996).[3]

[3] Engle-Warnick and Slonim (2003) use deterministic finite automata to describe play in an indefinitely repeated prisoner's dilemma. Aoyagi and Frechette (2004) fit finite automata with transitions determined by thresholds to data in an indefinitely repeated prisoner's dilemma game with imperfect monitoring; in their model heterogeneity is characterized by the level of the threshold, not the structure of the machine.

We define a *machine*, by the quadruple (Q, λ, μ, q). Each machine has a non-empty and finite set of *states*, Q. One of these states, $q_s \in Q$, is the *initial state*. In any given state, the machine takes an action given by an *action probability mass function*, $\mu\colon A_i \times Q \to \Re$. Thus, if the machine is in state $q \in Q$, it plays $a_i \in A_i$ with probability $\mu(a_i; q)$. We define a (deterministic) *state transition function* for each state of the machine, $\lambda\colon A \times Q \to Q$, that maps the current action profile and state of the machine to the next state that the machine will enter. Thus, if the action profile is $a \in A$ and the machine is in state $q \in Q$, then the machine will enter state $\lambda(a; q)$.

The following two examples illustrate the above ideas. First, we illustrate a machine that implements an "85% grim trigger" strategy in the Prisoner's Dilemma:

$$m_1^1 = (Q_1^1, \lambda_1^1, \mu_1^1, q_1^1), \quad Q_1^1 = \{1, 2\}, \quad q = 1$$

$$\lambda((d, D), 1) = \lambda((c, D), 1) = 2, \quad \lambda((d, C), 1) = \lambda((c, C), 1) = 1$$

$$\lambda((d, D), 1) = \lambda((c, D), 1) = \lambda((d, C), 2) = \lambda((c, C), 2) = 2$$

$$\mu(d; 1) = 0.15, \quad \mu(c; 1) = 0.85, \quad \mu(d; 2) = 0.85, \quad \mu(c; 2) = 0.15.$$

If this machine is in State 1, if the opponent plays C (that is, action profiles (c, C) and (d, C)) the machine will remain in State 1, and if the opponent plays D (action profiles (c, D) and (d, D)) the machine will move to State 2, and so on. Thus, this machine begins by playing c with probability 0.85 and continues to do so as long as the opponent is observed to cooperate. If the opponent ever defects, the machine switches to State 2 for the remainder of the game and plays d with probability 0.85.

Second, we present an "85% tit-for-tat" machine:

$$m_1^2 = (Q_1^2, \lambda_1^2, \mu_1^2, q_1^2), \quad Q_1^2 = \{1, 2\}, \quad q = 1$$

$$\lambda((d, D), 1) = \lambda((c, D), 1) = 2, \quad \lambda((d, C), 1) = \lambda((c, C), 1) = 1$$

$$\lambda((d, D), 1) = \lambda((c, D), 1) = 2, \quad \lambda((d, C), 2) = \lambda((c, C), 2) = 1$$

$$\mu(d;1) = 0.15, \quad \mu(c;1) = 0.85, \quad \mu(d;2) = 0.85, \quad \mu(c;2) = 0.15.$$

This machine differs from grim solely through a modification of the transition function in State 2, whereby if the opponent cooperates the machine reenters State 1. Thus, this machine behaves very much like a traditional Tit-For-Tat except that it always has a slight chance (15%) of taking the opposite of the "traditional" action.

To complete the framework we combine the above ideas. For a given game, with stage game γ, we assign each player $i \in N$ a machine, $m_i = (Q_i, \lambda_i, \mu_i, q_i)$. We call such an assignment a *machine profile* and can imagine that these N machines repeatedly play the stage game against one another.

We assume that the machine, m_i, chosen by a player is the result of a random draw from a heterogeneous population of machines. Precisely, a *population* for player $i \in N$ is given by the triple $(K_i, \pi_i, (Q_i^k, \lambda_i^k, \mu_i^k, q_i^k)_{k \in K_i})$, where K_i is a set of *machine types* and $\pi_i \colon K_i \to \Re$ is a *machine probability mass function* giving the probability with which each machine is selected. For example, we could have a population of players that with, say, probability 0.50 plays 85% Grim Trigger and with probability 0.50 plays 85% Tit-For-Tat. A *population profile* is an assignment to each player type of a machine population. This is useful to think about in asymmetric games where players of different types play different roles, and thus may use different strategy sets. We can think of drawing a machine from each of the N populations, and having the resulting machines repeatedly play the stage game.

3 Data and Inference

In this section we describe how to draw inferences about unknown populations and the assignment of subjets to machines, given observed behavior in repeated games. Our goal is to be able to numerically approximate functions of posterior expectations of functions of these unknown quantities, including

- the posterior probability that one machine accounts for more than 50% of the population,

- the posterior mean and variance of the number of machine states of the machine used by a random subject,

- the same for a particular subject,

- the probability that some machine in the population has an absorbing state,

- the posterior mean and variance of the probability that a random subject cooperates on her first move, and

- the posterior probability that subjects s and s' use the same machine.

We accomplish this goal by simulating, using Markov Chain Monte Carlo techniques, a Markov chain on the space of the unknown quantities. The chain is ergodic and its unique invariant distribution is the posterior distribution of the unknown quantities, so Birkhoff's ergodic theorem implies that sample moments of the simulated chain converge almost surely to their respective population moments, as long as the latter exist. We first introduce notation for the relevant observed and unknown quantities, as well as for a latent variable representing the assignment of subjects to machines. We then give distributions for the observed data given unknown quantities, both conditional and unconditional on the latent machine assignments. Next, we complete the statistical model for unknown and observed quantities by providing prior distributions for the unknown quantities. After deriving some useful conditional posterior distributions, we describe algorithms for posterior simulation. After discussing the identification of machine types, we describe how to do draw inference on the number of machine types in a population.

3.1 Quantities

Observed data

A convenient unit of observation is the supergame, which consists of a duration T, determined stochastically as part of the experimental design, and a sequence of T action profiles, representing choice sequences recorded in the laboratory. A *supergame realization* for a stage game $(N, A, (u_i)_{i\in N})$ is a pair $\sigma = (T, (a_1, \ldots, a_T))$, where positive integer T is a number of periods and $(a_1, \ldots, a_T)$ is an observed action profile sequence. For each $t \in \{1, \ldots, T\}$, action profile $a_t = (a_{ti})_{i\in N} \in A$.

In an experiment, there are a number of subjects, each of whom plays a number of supergames. For each player, we have a set of subjects, whom we track through all the supergames they play. Since we ignore learning in this paper, we represent the supergame realizations played by a subject as a set rather than an ordered tuple. An *experiment realization* is a $(S_i, (R_s, (\sigma_s^r)_{r\in R_s})_{s\in S_i})_{i\in N}$, where for each $i \in N$, S_i is a set of subjects playing as player i, R_s is a set of indices to supergame realizations involving subject s and $\sigma_s^r = (T^r, (a_1^r, \ldots, a_{T^r}^r))$ is a supergame realization involving subject s.

The observed data consists of an experiment realization $\mathbf{e} \equiv (e_i)_{i\in N} \equiv (S_i, (R_s, (\sigma_s^r)_{r\in R_s})_{s\in S_i})_{i\in N}$. Since the S_i, the R_s and the T^r are part of the experimental design, we condition on these values throughout and suppress notation for this conditioning. When we treat $\mathbf{e}$ as a random variable below, we consider the action profiles to be random and the rest to be constant.

Unknown quantities

The unknown quantity that we are trying to learn about is the population profile $(p_i)_{i\in N} = (K_i, \pi_i, (Q_i^k, \lambda_i^k, \mu_i^k, q_i^k)_{k\in K})_{i\in N}$. For convenience and without loss of generality we assume hereafter that $A = \{1, \ldots, |A|\}$ and that for every $i \in N$ and $k \in K_i$, $Q_i^k = \{1, \ldots, |Q_i^k|\}$. We use the $|\cdot|$ notation for the cardinality of a set.

In a inferential context, two issues arise over the initial states q_i^k and state transition functions λ_i^k. First, any permutation of states gives an observationally equivalent combination of initial state and transition function. Second, if some states are unreachable from the initial state, the transition function is observationally equivalent to a transition function for a smaller state set.

We resolve these issues by selecting one combination of initial state and transition function from each equivalence class to represent the class. We will assign positive probability to the chosen combination, and zero probability to the other elements of its class. We choose the initial state q_i^k to be the first state. We form a list of the values $\lambda_i^k(a,q)$ in q-first lexicographic order and insist that each $q > 1$ appears in the list before the first occurence of each $q' > q$. We also insist that all states are reachable. More precisely, for all $i \in N$ and $k \in K_i$, we choose the q_i^k and λ_i^k that satisfy the following three conditions.

1. (order of initial state) $q_i^k = 1$,

2. (order of non-initial states) for every $a, a' \in A$ and every $q, q' \in Q_i^k \backslash \{1\}$ such that $(q' < q)$ or $(q' = q) \wedge (a' < a)$,

$$\lambda_i^k(a,q) \leq \lambda_i^k(a',q') + 1,$$

3. (no unreachable states) for every $a \in A$ and every $q \in Q_i^k \backslash \{1\}$, there exists a $q' \in Q_i^k$ and $a' \in A$ such that $q' < q$ and $\lambda_i^k(a',q') = q$.

Having set all the q_i^k equal to one, we suppress notation for initial states. We will call a state transition function λ *regular* if it satisfies conditions 2 and 3, and denote by $\Lambda(A,Q)$ the set of regular state transition functions on $A \times Q$.

We will use the following notational shortcuts.

$$\mathbf{K} \equiv (K_i)_{i \in N} \quad \boldsymbol{\pi} \equiv (\pi_i)_{i \in N} \quad \mathbf{Q} \equiv ((Q_i^k)_{k \in K_i})_{i \in N}$$

$$\boldsymbol{\lambda} \equiv ((\lambda_i^k)_{k \in K_i})_{i \in N} \quad \boldsymbol{\mu} \equiv ((\mu_i^k)_{k \in K_i})_{i \in N} \quad \mathbf{p} \equiv (\mathbf{K}, \boldsymbol{\pi}, \mathbf{Q}, \boldsymbol{\lambda}, \boldsymbol{\mu})$$

We augment the data by adding a latent variable representing machine assignments. For each $i \in N$, let *machine assignment function* $\phi_i \colon S_i \to K_i$ be the function assigning subjects to machine types. Let $\boldsymbol{\phi} = (\phi_i)_{i \in N}$. In Section 3.5, we will describe how to simulate the joint posterior distribution of machine assignments and other unknowns. Augmenting the data in this way simplifies posterior simulation enormously, and the posterior distribution of machine assignments is interesting in its own right.

3.2 Data distributions

We assume that for each player $i \in N$, the actions of each subject $s \in S_i$ in all the supergames $(\sigma_s^r)_{r \in R_s}$ are generated by one of the machines of population p_i. Machine assignments are independant across subjects, and governed by the machine probability mass function π_i.

The data distribution $\mathbf{e}|\mathbf{p}$ is given by the probability mass function

$$f(\mathbf{e}|\mathbf{p}) = \prod_{i \in N} \prod_{s \in S_i} \sum_{k \in K_i} \pi_i(k) \prod_{r \in R_s} \prod_{t=1}^{T^r} \mu_i^k(a_{ti}^r; q(a_1^r, \ldots, a_{t-1}^r; \lambda_i^k),$$

where $q(a_1, \ldots, a_{t-1}; \lambda_i^k)$ is the state that machine with initial state 1 and transition function λ_i^k will be in at period t, after the observed action profile sequence $(a_1, \ldots, a_{t-1})$. That is,

$$q(a_1, \ldots, a_{t-1}; \lambda_i^k) \equiv \begin{cases} q_i^k & t = 1 \\ \lambda_i^k(a_{t-1}, q_{i,t-1}^k(a_1, \ldots, a_{t-2})) & t = 2, \ldots, T. \end{cases} \tag{1}$$

The joint distribution $\mathbf{e}, \boldsymbol{\phi}|\mathbf{p}$ of data and machine assignments given unknown parameters is given by the distributions $\mathbf{e}|\boldsymbol{\phi}, \mathbf{p}$ and $\boldsymbol{\phi}|\mathbf{p}$, with probability mass functions

$$f(\mathbf{e}|\boldsymbol{\phi}, \mathbf{p}) = \prod_{i \in N} \prod_{s \in S_i} \prod_{r \in R_s} \prod_{t=1}^{T^r} \mu_i^{\phi_i(s)}(a_{ti}^r; q(a_1^r, \ldots, a_{t-1}^r, \lambda_i^{\phi_i(s)}))$$

and

$$f(\boldsymbol{\phi}|\mathbf{p}) = \prod_{i \in N} \prod_{s \in S_i} \sum_{k \in K_i} \pi_i(\phi_i(s))).$$

3.3 Prior Distributions

We complete the model by providing a prior distribution for $\mathbf{p}$. Populations in a population profile are independant and machines in a population are i.i.d. We leave the priors on the number of machines in a population and the number of machine states of a machine for the user to specify. These quantities are thought to be small positive integers, so this flexibility does not come with a significant burden. The prior on the transition functions is uniform on the set of regular transition functions.

The priors on the machine probability mass functions π_i and the action probability mass functions μ_i^k are both Dirichlet. Dirichlet distributions are commonly used as distributions over discrete distributions. In order to treat machines in populations and actions in action sets symmetrically, we impose exchangeability of machine probabilities and action probabilities, which reduces the number of parameters of each Dirichlet distribution to one.

The choice of the Dirichlet distribution for action probabilities allows us to integrate out all the action probabilities from the posterior distribution, an enormous computational advantage. The user chooses a single parameter ν for this distribution, which has a meaningful interpretation, discussed below.

We now give details on the prior distribution for $\mathbf{p}$. It has the following conditional independance structure:

$$f(\mathbf{K}, \boldsymbol{\pi}, \mathbf{Q}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{K}) \cdot f(\boldsymbol{\pi}|\mathbf{K}) \cdot f(\mathbf{Q}|\mathbf{K}) \cdot f(\boldsymbol{\lambda}|\mathbf{K}, \mathbf{Q}) \cdot f(\boldsymbol{\mu}|\mathbf{K}, \mathbf{Q}).$$

Distribution K

The numbers $|K_i|$ of machine states are i.i.d. across players i, with probability masses $\omega_k \equiv \Pr[|K_i| = k]$, $k = 1, \ldots, \infty$. Thus the probability mass function for $\mathbf{K}$ is given by

$$f(\mathbf{K}) = \prod_{i \in N} f(K_i) = \prod_{i \in N} \omega_{|K_i|}.$$

The $|K_i|$ are thought to be small numbers, and we recommend truncating the prior to a small set, such as $\{1, 2, 3\}$ or $\{1, 2, 3, 4\}$.

Distribution $\boldsymbol{\pi}|\mathbf{K}$

The machine probability mass functions π_i are independent, with each π_i distributed as a Dirichlet random variable with $|K_i|$ parameters, all equal to α. Thus the density for $\boldsymbol{\pi}|\mathbf{K}$ is given by

$$f(\boldsymbol{\pi}|\mathbf{K}) = \prod_{i \in N} f((\pi_i(k))_{k \in K_i}|K_i) = \prod_{i \in N} \frac{\Gamma(\alpha \cdot |K_i|))}{[\Gamma(\alpha)]^{|K_i|}} \prod_{k \in K_i} [\pi_i(k)]^{\alpha - 1}.$$

The choice of α determines how close to zero or one machine probabilities are likely to be. For $\alpha = 1$, the density is uniform on the $|K_i|$-dimensional simplex. Values $\alpha > 1$ favor more moderate machine probabilities such as $(\pi_i(0), \pi_i(1), \pi_i(2)) = (0.4, 0.3, 0.3)$. Values $\alpha \in (0, 1)$ favor more extreme machine probabilities such as $(\pi_i(0), \pi_i(1), \pi_i(2)) = (0.9, 0.04, 0.06)$.

Distribution $\mathbf{Q}|\mathbf{K}$

The sets Q_i^k of machine states are i.i.d. across machines k and players i, with the cardinality of each Q_i^k governed by the probability masses $\theta_Q \equiv \Pr[|Q_i^k| = Q]$, $Q = 1, \ldots, \infty$. Thus the probability mass function for $\mathbf{Q}|\mathbf{K}$ is

$$f(\mathbf{Q}|\mathbf{K}) = \prod_{i \in N} \prod_{k \in K_i} f(Q_i^k) = \prod_{i \in N} \prod_{k \in K_i} \theta_{|Q_i^k|}.$$

The $|Q_i^k|$ are thought to be small numbers, and we recommend truncating the prior to a small set, such as $\{1, 2, 3\}$ or $\{1, 2, 3, 4\}$.

Distribution $\boldsymbol{\lambda}|\mathbf{K}, \mathbf{Q}$

Given $\mathbf{K}$ and $\mathbf{Q}$, the state transition functions λ_i^k are independant across players i and machine types k, with a uniform distribution over the set $\Lambda(A, Q_i^k)$ of regular transition functions $\lambda\colon A \times Q_i^k \to Q_i^k$.

Thus the probability mass function for $\boldsymbol{\lambda}|\mathbf{K},\mathbf{Q}$ is

$$f(\boldsymbol{\lambda}|\mathbf{K},\mathbf{Q}) = \prod_{i\in N}\prod_{k\in K_i} f(\lambda_i^k|Q_i^k) = \prod_{i\in N}\prod_{k\in K_i} \frac{1}{n(|A|,|Q_i^k|)},$$

where $n(|A|,|Q|)$ is the number of regular state transition functions $\lambda\colon A\times Q\to Q$.

We now derive a expression for $n(|A|,|Q|)$. We first derive a recursive expression for the number $\bar{n}(|A|,|Q|)$ of maps $\lambda\colon A\times Q\to Q$ statisfying condition 2 (no unreachable states) but not necessarily condition 3 (order of non-initial states).

The total number of maps $\lambda : A\times Q\to Q$ is $|Q|^{|A||Q|}$. The number of maps with exactly m unreachable states is

$$\binom{|Q|-1}{m}\cdot \bar{n}(|A|,|Q|-m)\cdot |Q|^{m|A|}$$

The first factor gives the number of choices of m unreachable states out of $|Q|-1$ non-initial states. The second factor gives the number of maps on the $|Q|-m$ reachable states where all states are indeed reachable. The third factor gives the number of maps $A\times Q^*\to Q$, where Q^* is a set of m unreachable states. For $|Q|=1$, the total number of maps is 1, and for this map, all states are reachable. Therefore $\bar{n}(|A|,1)=1$. For $|Q|>1$, we can calculate recursively

$$\bar{n}(|A|,|Q|) = |Q|^{|A||Q|} - \sum_{m=1}^{|Q|-1}\binom{|Q|-1}{m}\cdot \bar{n}(A,|Q|-m)\cdot |Q|^{m|A|}.$$

We now obtain $n(|A|,|Q|)$ by dividing by the number of permutations of the non-initial states:

$$n(|A|,|Q|) = \frac{\bar{n}(|A|,|Q|)}{|Q-1|!}.$$

Distribution $\boldsymbol{\mu}|\mathbf{K},\mathbf{Q}$

Given $\mathbf{K}$ and $\mathbf{Q}$, the action probability mass functions $\mu_i^k(\cdot;q)$ are independant across players i, machine types k and machine states $q\in Q_i^k$, with each distributed as a Dirichlet random variable with $|Q_i^k|$ parameters, all equal to ν. Thus the density for $\boldsymbol{\mu}|\mathbf{K},\mathbf{Q}$ is

$$\begin{aligned} f(\boldsymbol{\mu}|\mathbf{K},\mathbf{Q}) &= \prod_{i\in N}\prod_{k\in K_i}\prod_{q\in Q_i^k} f((\mu_i^k(a_i;q))_{a_i\in A_i}|Q_i^k) \\ &= \prod_{i\in N}\prod_{k\in K_i}\prod_{q\in Q_i^k} \frac{\Gamma(\nu|A_i|)}{[\Gamma(\nu)]^{|A_i|}} \prod_{a_i\in A_i} [\mu_i^k(a_i;q)]^{\nu-1}. \end{aligned}$$

The choice of ν determines how close to zero or one action probabilities are likely to be. For $\nu = 1$, the density is uniform on the $|A_i|$-dimensional simplex. Values $\nu > 1$ favor more moderate action probabilities such as $(\mu_i^k(C), \mu_i^k(D)) = (0.6, 0.4)$. Values $\nu \in (0,1)$ favor more extreme action probabilities such as $(\mu_i^k(C), \mu_i^k(D)) = (0.1, 0.9)$. The choice of ν expresses the user's belief about how close machines are to deterministic machines.

3.4 Conditional Posterior Distributions

In this section we derive some posterior distributions that will be useful for posterior simulation. The derivations are straighforward once we know the following property of the Dirichlet and multinomial distributions. If $\theta = (\theta_1, \ldots, \theta_n)$ is Dirichlet with parameter $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $c|\theta = (c_1, \ldots, c_n)|\theta$ is multinomial with parameters θ and $\sum_{j=1}^n c_j$, then $\theta|c$ is Dirichlet with parameter $(\alpha_1 + c_1, \ldots, \alpha_n + c_n)$ and the marginal probability mass function for c is

$$f(c) = \frac{\prod_{j=1}^n \Gamma(\alpha_j)}{\Gamma(\sum_{j=1}^n \alpha_j)} \cdot \frac{\Gamma(\sum_{j=1}^n \alpha_j + c_j)}{\prod_{j=1}^n \Gamma(\alpha_j + c_j)}.$$

The fact that the prior and data densities both factor by player implies that the posterior does as well, which simplifies affairs greatly.

Distribution $\boldsymbol{\pi}|\mathbf{K},\mathbf{Q},\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\phi},\mathbf{e}$

The machine probability mass functions π_i are conditionally independent, with each π_i distributed as a Dirichlet random variable with parameters $(\alpha + d_i^k(\phi_i))_{k\in K_i}$, where the *machine* *counts* $d_i^k(\phi_i)$ are given by[4]

$$d_i^k(\phi_i) \equiv \sum_{s\in S_i} \delta_{k,\phi_i(s)} \quad k \in K_i, \quad i \in N. \tag{2}$$

That is, the conditional posterior density for $\boldsymbol{\pi}$ is

$$\begin{aligned} f(\boldsymbol{\pi}|\mathbf{K},\mathbf{Q},\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\phi},\mathbf{e}) &= \prod_{i\in N} f(\pi_i|K_i,\phi_i) \qquad (3)\\ &= \prod_{i\in N} \frac{\Gamma(\sum_{k\in K_i}\alpha + d_i^k(\phi_i))}{\prod_{k\in K_i}\Gamma(\alpha + d_i^k(\phi_i))} \prod_{k\in K_i} [\pi_i(k)]^{\alpha + d_i^k(\phi_i) - 1}. \end{aligned}$$

Distribution $\boldsymbol{\mu}|\mathbf{K},\boldsymbol{\pi},\mathbf{Q},\boldsymbol{\lambda},\boldsymbol{\phi},\mathbf{e}$

The action probability mass functions $\mu_i^k(\cdot;q)$ are independant across players i, machine types k and machine states $q \in Q_i^k$, with each $(\mu_i^k(a_i,q))_{a_i\in A_i}$ distributed as a Dirichlet random variable with parameters $(\nu + c_i^k(a_i,q;\lambda_i^k,\phi_i))_{a_i\in A_i}$, where the *action counts* $c_i^k(a_i,q;\lambda_i^k,\phi_i)$ are given by (recall equation (1) defining $q(\cdot)$)

$$\begin{aligned} c_i^k(a_i,q;\lambda_i^k,\phi_i) &= \sum_{s\in\{S_i:\,\phi_i(s)=k\}} \sum_{r\in R_s} \sum_{t=1}^{T^r} \delta_{q,q(a_1^r,\ldots,a_{t-1}^r,\lambda_i^k)} \delta_{a_i,a_{ti}^r}, \qquad (4)\\ & \qquad a_i \in A_i,\ q \in Q_i^k,\ k \in K_i,\ i \in N. \end{aligned}$$

That is, the posterior density for $\boldsymbol{\mu}$ is given by

$$\begin{aligned} & f(\boldsymbol{\mu}|\mathbf{K},\boldsymbol{\pi},\mathbf{Q},\boldsymbol{\lambda},\boldsymbol{\phi},\mathbf{e})\\ &= \prod_{i\in N}\prod_{k\in K_i}\prod_{q\in Q_i^k} f((\mu_i^k(a_i;q))_{a_i\in A_i}|Q_i^k,\lambda_i^k,\phi_i,\mathbf{e}) \qquad (5)\\ &= \prod_{i\in N}\prod_{k\in K_i}\prod_{q\in Q_i^k} \frac{\Gamma(\sum_{a_i\in A_i}\nu + c_i^k(a_i,q;\lambda_i^k,\phi_i))}{\prod_{a_i\in A_i}\Gamma(\nu + c_i^k(a_i,q;\lambda_i^k,\phi_i))} \prod_{a_i\in A_i} [\mu_i^k(a_i;q)]^{\nu + c_i^k(a_i,q;\lambda_i^k,\phi_i) - 1}. \end{aligned}$$

[4] δ is the Kronecker delta function. For integers i and j, $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

Distribution $\mathbf{Q}, \boldsymbol{\lambda}|\mathbf{K}, \boldsymbol{\pi}, \boldsymbol{\phi}, \mathbf{e}$

We can marginalize out $\boldsymbol{\mu}$ in $\mathbf{e}, \boldsymbol{\mu}|\mathbf{K}, \boldsymbol{\pi}, \mathbf{Q}, \boldsymbol{\lambda}, \boldsymbol{\phi}$, given by $\mathbf{e}|\mathbf{K}, \boldsymbol{\pi}, \mathbf{Q}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\phi}$ and $\boldsymbol{\mu}|\mathbf{K}, \mathbf{Q}$, to obtain the probability mass function

$$f(\mathbf{e}|\mathbf{K}, \boldsymbol{\pi}, \mathbf{Q}, \boldsymbol{\lambda}, \boldsymbol{\phi}) = \prod_{i \in N} \prod_{k \in K_i} \prod_{q \in Q_i^k} \frac{[\Gamma(\nu)]^{|A_i|}}{\Gamma(\nu|A_i|)} \cdot \frac{\Gamma(\sum_{a_i \in A_i} \nu + c_i^k(a_i, q; \lambda_i^k, \phi_i))}{\prod_{a_i \in A_i} \Gamma(\nu + c_i^k(a_i, q; \lambda_i^k, \phi_i))}. \tag{6}$$

The pairs (Q_i^k, λ_i^k) of machine state set Q_i^k, and state transition function λ_i^k are conditionally independant across players i and machine types k. The posterior probability mass function for $\mathbf{Q}, \boldsymbol{\lambda}|\mathbf{K}, \boldsymbol{\pi}, \boldsymbol{\phi}, \mathbf{e}$ is

$$\begin{aligned} & f(\mathbf{Q}, \boldsymbol{\lambda}|\mathbf{K}, \boldsymbol{\pi}, \boldsymbol{\phi}, \mathbf{e}) \\ &= \prod_{i \in N} \prod_{k \in K_i} f(Q_i^k, \lambda_i^k|\phi_i, e_i) \qquad (7) \\ &\propto \prod_{i \in N} \prod_{k \in K_i} \frac{\theta_{|Q_i^k|}}{n(|A|, |Q_i^k|)} \cdot \frac{[\Gamma(\nu)]^{|A_i||Q_i^k|}}{[\Gamma(\nu|A_i|)]^{|Q_i^k|}} \prod_{q \in Q_i^k} \frac{\Gamma(\sum_{a_i \in A_i} \nu + c_i^k(a_i, q; \lambda_i^k, \phi_i))}{\prod_{a_i \in A_i} \Gamma(\nu + c_i^k(a_i, q; \lambda_i^k, \phi_i))}. \end{aligned}$$

Distribution $\boldsymbol{\phi}|\mathbf{p}, \mathbf{e}$

The machine assignments $\phi_i(s)$ are conditionally independant across players i and subjects s. The posterior probability mass function for $\boldsymbol{\phi}$ is

$$\begin{aligned} f(\boldsymbol{\phi}|\mathbf{p}, \mathbf{e}) &= \prod_{i \in N} \prod_{s \in S_i} f(\phi_i(s)|\pi_i, (Q_i^k, \mu_i^k, \lambda_i^k)_{k \in K_i}, \mathbf{e}) \qquad (8) \\ &= \prod_{i \in N} \prod_{s \in S_i} \frac{\pi_i(k) \prod_{r \in R_s} \prod_{t=1}^{T^r} \mu_i^k(a_{ti}^r; q_{it}^k(a_1^r, \ldots, a_{t-1}^r))}{\sum_{\kappa \in K_i} \pi_i(\kappa) \prod_{r \in R_s} \prod_{t=1}^{T^r} \mu_i^\kappa(a_{ti}^r; q_{it}^\kappa(a_1^r, \ldots, a_{t-1}^r))}. \end{aligned}$$

3.5 Posterior Simulation

Since both the prior and the posterior factor by player, the $|N|$ populations are *a priori* and *a posteriori* independent, and we can independently simulate the unknown quantities for each player i. We now fix $i \in N$ arbitrarily.

Here we will condition on K_i and describe how to simulate the distribution $\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i}, \phi_i | K_i, e_i$. Later, in Section 3.7, we discuss how to use a method for computing marginal likelihoods to obtain $f(K_i|e_i)$.

We describe seven different parameter update blocks, each of which stochastically updates one or more of the unknown quantities, conditioning only on the current values of other unknown quantities. We can chain together any number of these blocks (repetitions are allowed) to form a *sweep*. The sweep in thus a stochastic update of the vector of unknown quantities conditioning only on the current value of the vector. It therefore defines a Markov transition kernel. Under conditions described below, we obtain an ergodic Markov chain whose invariant distribution is the posterior distribution $\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i}, \phi_i | K_i, e_i$, and therefore we can use the chain to simulate the posterior.

Two blocks preserve the distribution $\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i}, \phi_i | K_i, e_i$. Block B_π updates the machine probability mass function π_i and block B_ϕ updates the machine assignment function ϕ_i.

Four blocks preserve the marginal distribution $\pi_i, (Q_i^k, \lambda_i^k)_{k \in K_i}, \phi_i | K_i, e_i$. Blocks $B_{Q,\lambda}(k)$ and $B'_{Q,\lambda}(k)$ update, for the given $k \in K_i$, the machine state set Q_i^k and the state transition function λ_i^k. Block $B_\lambda(k)$ updates, for the given $k \in K_i$, the state transition function λ_i^k. Block B'_ϕ updates the machine assignment function ϕ_i.

Finally, block $B_\mu(k)$ draws, for the given $k \in K_i$, the action probability mass function μ_i^k from the distribution $\mu_i^k | K_i, \pi_i, (Q_i^k, \lambda_i^k)_{k \in K_i}, \phi_i, e_i$. In this way, a block preserving $\pi_i, (Q_i^k, \lambda_i^k)_{k \in K_i}, \phi_i | K_i, e_i$ followed by a $B_\mu(k)$ block together preserve the distribution $\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i} | K_i, e_i$.

A sweep is a sequence of blocks, with possible repetitions. If a sweep satisfies the following conditions:

1. It includes B_π and B_ϕ and for each $k \in K_i$, $B_\mu(k)$ and one of $B_{Q,\lambda}$ or $B'_{Q,\lambda}$.

2. A block preserving $\pi_i, (Q_i^k, \lambda_i^k)_{k \in K_i}, \phi_i | K_i, e_i$ cannot be followed by a block preserving

$\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i}, \phi_i | K_i, e_i$.

3. The last block of the sweep must either be a $B_\mu(k)$ block or preserve $\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i}, \phi_i | K_i, e_i$.

then the Markov transition kernel is ergodic and its unique invariant distribution is $\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i}, \phi_i | K_i$ Then Birkhoff's ergodic theorem implies that posterior simulation sample moments converge almost surely to posterior population moments whenever the latter exist.

The fact that $\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i} | K_i, e_i$ is *an* invariant distribution of the Markov transition kernel follows from Conditions 2 and 3. All blocks required by Condition 1 have updates whose support is the same as the support of the appropriate conditional distribution, and together they update the entire parameter vector. The chain is therefore Harris recurrent and aperiodic. This in turn implies that the chain is ergodic and has a unique invariant distribution.

We now describe in detail the various blocks.

Blocks preserving $\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i}, \phi_i | K_i, e_i$

By (3), the distribution $\pi_i | K_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i}$ is Dirichlet. Block B_π draws π_i directly from this distribution using standard procedures.

By (8), the probability mass function for $\phi_i | K_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i}$ is a product of $|S_i|$ probability mass functions, each of which has only $|K_i|$ mass points. Block B_ϕ draws ϕ_i directly from this distribution using standard procedures.

Blocks preserving $\pi_i, (Q_i^k, \lambda_i^k)_{k \in K_i}, \phi_i | K_i, e_i$

Block B'_ϕ is a random walk Metropolis update which proposes a ϕ_i^* identical to ϕ_i except for a random mutation and adopts the proposal with a certain probability. The proposal consists of drawing a random subject s^* from the uniform distribution on S_i, then drawing $\phi_i^*(s^*)$ from the uniform distribution on $K_i \backslash \phi_i(s^*)$. For all $s \in S_i \backslash \{s^*\}$, we set $\phi_i^*(s) = \phi_i(s)$.

The proposal is a random walk on the space of machine assignments. We accept the proposal ϕ_i^* with probability

$$\min\left[1, \frac{\pi_i(k^*)}{\pi_i(k)} \cdot \frac{B(k,\phi_i^*)}{B(k,\phi_i)} \cdot \frac{B(k^*,\phi_i^*)}{B(k^*,\phi_i)}\right],$$

where $k = \phi_i(s)$, $k^* = \phi_i^*(s)$, and for all $k \in K_i$ and all machine assignment functions $\phi_i \colon S_i \to K_i$,

$$B(k,\phi_i) \equiv \prod_{q \in Q_i^k} \frac{\Gamma(\sum_{a_i \in A_i} \nu + c_i^k(a_i, q; \lambda_i^k, \phi_i))}{\prod_{a_i \in A_i} \Gamma(\nu + c_i^k(a_i, q; \lambda_i^k, \phi_i))}.$$

Acceptance means we move to the proposed state, replacing ϕ_i with ϕ_i^*. If we do not accept, ϕ_i remains unchanged.

For given $k \in K_i$, block $B_{Q,\lambda}$ is an independance Metropolis Hastings update with the prior distribution $f(Q_i^k, \lambda_i^k)$ as the proposal distribution. This involves the following steps. We first draw the proposal Q_i^{k*}, λ_i^{k*} from the joint prior distribution of Q_i^k and λ_i. We then accept with probability

$$\min\left[1, \frac{\theta_{Q_i^k}}{\theta_{Q_i^{k*}}} \cdot \frac{[\Gamma(\nu)]^{|A_i|(|Q_i^{k*}|-|Q_i^k|)}}{[\Gamma(\nu|A_i|)]^{|Q_i^{k*}|-|Q_i^k|}} \cdot \frac{\prod_{q\in Q_i^{k*}} \frac{\Gamma(\sum_{a_i\in A_i} \nu + c_i^{k*}(a_i,q;\lambda_i^{k*},\phi_i))}{\prod_{a_i\in A_i} \Gamma(\nu + c_i^{k*}(a_i,q;\lambda_i^{k*},\phi_i))}}{\prod_{q\in Q_i^k} \frac{\Gamma(\sum_{a_i\in A_i} \nu + c_i^{k}(a_i,q;\lambda_i^{k},\phi_i))}{\prod_{a_i\in A_i} \Gamma(\nu + c_i^{k}(a_i,q;\lambda_i^{k},\phi_i))}}\right].$$

Acceptance means we move to the proposed state, replacing Q_i^k and λ_i^k with Q_i^{k*} and λ_i^{k*}. If we do not accept, λ_i^k remains unchanged.

Block $B'_{\lambda,Q}$ is also an independance Metropolis-Hastings update, but features a proposal distribution more closely resembling the posterior distribution $Q_i^k, \lambda_i^k | K_i, \phi_i^k, e_i$.

Denote by Q_i^{k*}, λ_i^{k*} the proposal. Q_i^{k*} is drawn from the prior distribution for Q_i^k:

$$f(Q_i^{k*}|K_i, \phi_i, e_i) = \theta_{|Q_i^{k*}|}.$$

We construct λ_i^{k*} stochastically, one value $\lambda_i^{k*}(a,q)$ at a time. To establish an order for these values, we introduce a lexicographic order on $A \times \mathbb{N}$. It is useful to think of $\mathbb{N}$ as an extention of Q_i^{k*}. For all $a, a' \in A$ and $q, q' \in \mathbb{N}$,

$$(a,q) > (a',q') \text{ iff } (q > q') \text{ or } (q = q' \wedge a > a').$$

We also introduce some convenient related notation. For all $a \in A$ and $q \in \mathbb{N}$, we denote by $(a,q)_+$ the successor of (a,q) and by $(a,q)_-$ its predecessor.

After each successive draw of a value $\lambda_i^k(a,q)$, we have a "history" $L(a,q)$ of drawn values:

$$L(a,q) \equiv (\lambda(a',q'))_{(a',q') \leq (a,q)}.$$

We now introduce a measure of how well an incomplete transition function, given by history $L(a,q)$, fits the data. We first describe an infinite state machine which agrees with the incomplete transition function for values up to and including $\lambda(a,q)$, but otherwise has no recurrent states. The initial state is 1 and the transition function $\bar{\lambda}_{L(a,q)} \colon A \times \mathbb{N} \to \mathbb{N}$ is given by

$$\bar{\lambda}_{L(a,q)}(a',q') = \begin{cases} \lambda_i^k(a',q') & (a',q') \leq (a,q) \\ |Q_i^{k*}| + 1 & (a',q') = (a,q)_+ \\ \lambda_i^k((a',q')_-) + 1 & (a',q') > (a,q)_+ \end{cases}.$$

We now consider the question of fit. Recalling (6), the probability mass function of the data e_i^k attributable to subjects of player i assigned to machine $k \in K_i$, given the state set Q_i^k, transition function λ_i^k and machine assignment function ϕ_i, but with action probabilities μ_i^k marginalized out, is

$$f(e_i^k | Q_i^k, \lambda_i^k, \phi_i) = \prod_{q \in Q_i^k} \frac{[\Gamma(\nu)]^{|A_i|}}{\Gamma(\nu |A_i|)} \cdot \frac{\Gamma(\sum_{a_i \in A_i} \nu + c_i^k(a_i, q; \lambda_i^k, \phi_i))}{\prod_{a_i \in A_i} \Gamma(\nu + c_i^k(a_i, q; \lambda_i^k, \phi_i))}.$$

This naturally extends to infinite state machines with $\mathbb{N}$ replacing Q_i^k in this equation and in the definition of action counts c_i^k of equation (4). The resulting infinite product is no cause for alarm. With a finite amount of data, the counts c_i^k will be non-zero only for a finite number of values $q \in \mathbb{N}$, and thus only a finite number of factors will take on values other than 1.

We are now ready to describe the proposal distribution, which is designed to favor machines with higher degrees of fit. We have the following probability mass function for λ_i^k:

$$h(\lambda_i^{k*}|K_i, Q_i^{k*}, \phi_i^k, e_i) = \prod_{(a,q)\in A\times Q_i^{k*}} h(\lambda_i^{k*}(a,q)|L((a,q)_-), K_i, Q_i^{k*}, \phi_i^k, e_i),$$

where

$$h(\lambda_i^{k*}(a,q)|L((a,q)_-), K_i, Q_i^{k*}, \phi_i^k, e_i) \propto$$

$$\begin{cases} \infty & \lambda_i^{k*} = q+1,\ a = |A|,\ \max_{(a',q')<(a,q)} \lambda_i^{k*}(a',q') = q \\ f(e_i^k|Q_i^k, \bar{\lambda}_{L(a,q)}, \phi_i) & \lambda_i^{k*} \in \{1,\ldots,\max_{(a',q')<(a,q)} \lambda_i^{k*}(a',q')+1\} \\ 0 & \text{otherwise.} \end{cases}$$

The value ∞ forces us to choose $\lambda_i^{k*} = q+1$ if otherwise the resulting machine would have unreachable states. The support $\{1,\ldots,\max_{(a',q')<(a,q)} \lambda_i^{k*}(a',q')+1\}$ ensures that the non-initial states are correctly ordered.

For every $k \in K$, we draw Q_i^{k*} and λ_i^{k*} as described above, and accept with probability

$$\min\left[1, \frac{h(\lambda_i^k|K_i, Q_i^k, \phi_i, e_i)}{h(\lambda_i^{k*}|K_i, Q_i^{k*}, \phi_i, e_i)} \cdot \frac{\theta_{Q_i^k}}{\theta_{Q_i^{k*}}} \cdot \frac{[\Gamma(\nu)]^{|A_i|(|Q_i^{k*}|-|Q_i^k|)}}{[\Gamma(\nu|A_i|)]^{|Q_i^{k*}|-|Q_i^k|}} \cdot \frac{\prod_{q\in Q_i^{k*}} \frac{\Gamma(\sum_{a_i\in A_i} \nu + c_i^{k*}(a_i,q;\lambda_i^{k*},\phi_i))}{\prod_{a_i\in A_i}\Gamma(\nu+c_i^{k*}(a_i,q;\lambda_i^{k*},\phi_i))}}{\prod_{q\in Q_i^k} \frac{\Gamma(\sum_{a_i\in A_i} \nu + c_i^{k}(a_i,q;\lambda_i^{k},\phi_i))}{\prod_{a_i\in A_i}\Gamma(\nu+c_i^{k}(a_i,q;\lambda_i^{k},\phi_i))}}\right].$$

For given $k \in K_i$, block B_λ is a random walk Metropolis update which proposes a λ_i^{k*} identical to λ_i^k except for a random mutation. The proposal consists of drawing a random pair (a,q) from the uniform distribution on $A \times Q_i^k$, then drawing $\lambda_i^{k*}(a,q)$ from the uniform distribution on $Q_i^k \backslash \lambda_i^k(a,q)$. For all pairs $(a',q') \in (A\times Q_i^k)\backslash\{(a,q)\}$, we set $\lambda_i^{k*}(a',q') = \lambda_i^k(a',q')$. Then if necessary, we permute the non-initial states so that λ_i^{k*} satisfies the order condition for non-initial states.

The proposal is a random walk on the space of transition functions satisfying condition 2 (order of non-initial states) but not necessarily 3 (no unreachable states). We accept the proposal λ_i^{k*} with probability

$$\min\left[1, \frac{\prod_{q\in Q_i^{k*}} \frac{\Gamma(\sum_{a_i\in A_i} \nu + c_i^{k*}(a_i;q))}{\prod_{a_i\in A_i}\Gamma(\nu+c_i^{k*}(a_i;q))}}{\prod_{q\in Q_i^k} \frac{\Gamma(\sum_{a_i\in A_i} \nu + c_i^{k}(a_i;q))}{\prod_{a_i\in A_i}\Gamma(\nu+c_i^{k}(a_i;q))}} \cdot 1_{\Lambda(A,Q_i^k)}(\lambda_i^{k*})\right].$$

Note that the indicator function $1_{\Lambda(A,Q_i^k)}$, which comes from the prior on λ_i^k, means we reject any proposal with unreachable states.

A block drawing μ_i^k from its conditional posterior distribution

By (5), the density for $\mu_i^k|K_i, \pi_i, Q_i^k, \lambda_i^k, \phi_i, e_i$ is a product of independent Dirichlet distributions. Block B_μ draws each μ_i^k directly from its Dirichlet distribution using standard procedures.

3.6 Machine Type Identification

In this section we discuss an identification issue concerning machine type labels. Any permutation of machine types of a population p_i gives an observationally equivalent population. So the population in the example of Section 2, where m^1 is 85% grim trigger, m^2 is 85% tit-for-tat and their respective probabilities are $\pi(1) = 0.6$ and $\pi(2) = 0.4$, is observationally equivalent to a population where m^1 is 85% tit-for-tat, m^2 is 85% grim trigger, $\pi(1) = 0.4$ and $\pi(2) = 0.6$.

Note that we have already resolved a similar issue with state labels: if we permute the states of a regular state transition function we obtain a non-regular state transition function, which the prior assigns zero probability. This is an example of identification through labelling restrictions, where the restriction is expressed by conditions 2 and 3 on λ.

The non-identification of machine types is not always a problem. Many questions can be answered without recourse to machine type identification restrictions. The posterior distribution of the number of machine states (a measure of complexity) of the machine to which a particular subject is assigned can be obtained without such recourse.

Sometimes, however, it is convenient to introduce labelling restrictions. Suppose $S_i = \{1, 2, 3\}$ is the set of subjects for player i and $K_i = \{1, 2\}$ is the set of machine types for i. With no labelling restrictions, we might infer that machine assignments $(\phi_i(1), \phi_i(2), \phi_i(3))$

Table 1: Example of Posterior Probabilities of Machine Assignments

Values of $(\phi_i(1), \phi_i(2), \phi_i(3))$	Posterior Probability
(1,1,1), (2,2,2)	0.05
(1,1,2), (2,2,1)	0.05
(1,2,1), (2,1,2)	0.10
(1,2,2), (2,1,1)	0.30

have the posterior distribution given in Table 2.

Each row gives an equivalence class of two observationally equivalent machine assignments. Symmetry requires their posterior probabilities to be equal, and this common probability is also tabulated. The partition of the subject set implied by the machine assignments is important, not the label values.

We identify machine types by selecting one map from each equivalence class to represent the class. If, for example, we choose the first map of each row, we are labelling as machine 1 the machine to which subject 1 is assigned, and machine 2 as the other machine.

Alternatively, we can choose the high probability map $(1, 2, 2)$ from the fourth row and the three maps which differ from it by only one subject assignment: $(2, 2, 2)$, $(1, 1, 2)$ and $(1, 2, 1)$. We can think of machine 1 as the machine to which subjects 2 and 3 tend to be assigned and to which subject 1 tends not to be assigned. This kind of identification is useful if posterior probability is dominated by a "central" machine assignment, a few "close" machine assignments, and the observationally equivalent machine assignments obtained by permuting machine type labels.

We now make precise a identification strategy based on this general idea. This strategy involves a *central* machine assignment: $\bar{\phi}_i \colon S \to K_i$, which we describe how to obtain below. Each class of observationally equivalent machine assignments is represented by the one closest in Hamming distance to $\bar{\phi}$, with ties resolved using a lexicographic order. The Hamming distance between two machine assignment functions is the number of subjects for which their values are not equal.

The central machine assignment $\bar{\phi}_i$ is a fixed point satisfying

$$\bar{\phi}_i(s) = \arg\max_{k \in K_i} \Pr[\phi_i(s) = k | K_i, e_i] \quad \text{for every } s \in S_i,$$

where the probabilities are computed after identifying machine types as described in the previous paragraph.

3.7 Marginal Likelihood Computation

To compute the posterior mass function $f(K_i|e_i)$, it is useful to have the values $f(e_i|K_i)$ for all K_i with positive prior probability, since we can then compute $f(K_i|e_i)$ using Bayes rule:

$$f(K_i|e_i) = \frac{f(K_i)f(e_i|K_i)}{\sum_{\kappa=1}^{\infty} \Pr[K_i = \kappa] f(e_i|K_i = \kappa)}.$$

We use the method of Gelfand and Dey to compute the values $f(e_i|K_i)$. We require a properly normalized density function $\hat{f}(\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i} | K_i, e_i)$, resembling the posterior density $f(\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i} | K_i, e_i)$. With M draws $(\pi_i^m, (Q_i^{k,m}, \lambda_i^{k,m}, \mu_i^{k,m})_{k \in K_i})_{m \in \{1,\ldots,M\}}$ simulated from the posterior distribution, we compute

$$\hat{g} = \frac{1}{M} \sum_{m=1}^{M} \frac{\hat{f}(\pi_i^m, (Q_i^{k,m}, \lambda_i^{k,m}, \mu_i^{k,m})_{k \in K_i} | K_i)}{f(e_i, \pi_i^m, (Q_i^{k,m}, \lambda_i^{k,m}, \mu_i^k)_{k \in K_i} | K_i)}.$$

Provided that our Markov chain is ergodic and its invariant distribution is the posterior distribution $\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i}, \phi_i | K_i, e_i$, the sample mean $\hat{g}$ converges almost surely to the following population mean g:

$$\begin{aligned} g &\equiv E\left[\frac{\hat{f}(\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i} | K_i, e_i)}{f(e_i, \pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i} | K_i)} | K_i, e_i\right] \\ &= \frac{1}{f(e_i|K_i)} \cdot E\left[\frac{\hat{f}(\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i} | K_i, e_i)}{f(\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k \in K_i} | K_i, e_i)} | K_i, e_i\right]. \end{aligned} \tag{9}$$

The posterior expectation in (9) is the following integral over the parameter space Θ:

$$\int_\Theta \frac{\hat{f}(\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k\in K_i}|K_i, e_i)}{f(\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k\in K_i}|K_i, e_i)} \cdot f(\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k\in K_i}|K_i, e_i)\, d\nu(\vartheta)$$
$$= \int_\Theta \hat{f}(\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k\in K_i}|K_i, e_i)\, d\nu(\vartheta) = 1,$$

where ν is the appropriate measure. Therefore g is the reciprocal of the marginal likelihood $f(e_i|K_i)$.

The choice of $\hat{f}$ is important, since the error of approximation depends on the distribution of the ratio $\hat{f}/f$. The ratio is bounded below by zero, so reducing the error of approximation involves controlling the right tail of this distribution. Roughly speaking, $\hat{f}$ should approximate f, but errors of understatement are less serious than errors of overstatement.

Our choice of $\hat{f}$ is based on the central machine assignment functions $\bar{\phi}_i$ discussed in Section 3.6:

$$\begin{aligned}
&\hat{f}(\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k\in K_i}|K_i, e_i) \\
&= f(\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k\in K_i}|K_i, e_i, \phi_i = \bar{\phi}_i, |Q_i^k| \leq \bar{Q}) \\
&= f(\pi_i|K_i, \phi_i = \bar{\phi}_i) \\
&\cdot \prod_{k\in K_i} f(Q_i^k, \lambda_i^k|\phi_i = \bar{\phi}_i, |Q_i^k| \leq \bar{Q}, e_i) \cdot f(\mu_i^k|Q_i^k, \lambda_i^k, \phi_i = \bar{\phi}_i, e_i)
\end{aligned}$$

Equation (3) gives the normalized density $f(\pi_i|K_i, \phi_i = \bar{\phi}_i)$ and equation (5) gives the normalized densities $f(\mu_i^k|Q_i^k, \lambda_i^k, \phi_i = \bar{\phi}_i, e_i)$. Equation (7) gives a probability mass function proportional to $f(Q_i^k, \lambda_i^k|\phi_i = \bar{\phi}_i, |Q_i^k| \leq \bar{Q}, e_i)$, but not the normalization constant. We calculate this normalization constant numerically by summing over all machines with up to $\bar{Q}$ states. The choice of $\bar{Q}$ reflects a tradeoff. The computational cost of the summation increases with $\bar{Q}$. However, as $\bar{Q}$ decreases, the variance of $\hat{f}/f$ increases due to the increased probability of the event $\hat{f}/f = 0$.

Note that the true posterior distribution $\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k\in K_i}|K_i, e_i$ is a mixture of distributions $\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k\in K_i}|K_i, \phi_i, e_i$, where the mixing distribution is $\phi_i|K_i, e_i$. One component of this mixture is $\pi_i, (Q_i^k, \lambda_i^k, \mu_i^k)_{k\in K_i}|K_i, \phi_i = \bar{\phi}_i, e_i$, which is proportional to $\hat{f}$ on the region where $|Q_i^k| \leq \bar{Q}$. We see that the ratio $g\hat{f}/f$ is bounded above by the reciprocal of the probability $\Pr[\phi_i = \bar{\phi}_i, |Q_i^k| < \bar{Q}|K_i, e_i]$. In practice, we find that the procedure for choosing $\bar{\phi}$ described in Section 3.6, and a feasible choice $\bar{Q}$ makes this probability fairly high, and we obtain very satisfactory standard errors for marginal likelihoods.

4 Experimental Design and Procedures

The stage game we study is the well known Prisoner's Dilemma. In our game the punishment (mutual defection) payoff is 60, the temptation (play d when the opponent plays C) is 180, the sucker (play c when the opponent plays D) is 0, and the reward (play c when the opponent plays C) is 90. Recall that both the unique stage-game equilibrium and sub-game perfect equilibrium in the finitely repeated game (by backward induction) are mutual defection in each round.

By the folk theorem, we know that in the indefinitely repeated game cooperative equilibria can arise via the threat that deviations off of the equilibrium path are punished forever by defection. To compute the minimum discount factor required to achieve cooperation it is sufficient to ensure that a one-period deviation is not profitable:

$$\sum_{t=0}^{\infty} \delta^t 90 \geq 180 + \sum_{t=1}^{\infty} \delta^t 60.$$

Thus for any discount factor $\delta \geq 0.75$ cooperative equilibria exist. In our experiments we simulated a discount factor of 0.80 by introducing a constant and independent probability of continuing the repeated game after each stage game. We set the continuation probability just above the threshold of 0.75 so that cooperation is possible but not necessarily obvious. With a continuation probability of 0.80 the expected length of a supergame is five rounds, thus it is possible to run laboratory experiments where we can observe many supergames.

In the experiments, subjects were randomly and anonymously paired to play twenty supergames of indefinite length. After each supergame subjects were randomly matched with a new opponent. The supergame lengths were determined in advance using the stated distribution.[5] Each subject experienced the same sequence of supergame lengths, which mitigated any differences in behavior due to different experiences with the random process. Subjects were informed of this procedure (but not the outcome) in advance.

The subjects were seated at a computer terminal displaying the earnings table for a row and a column player. All subjects viewed the game as if they were the row player. The choices of each stage game were recorded at the bottom of the screen, along with a label that identified each supergame, and subjects were able to review these data at any time. A screen informed the subjects whenever a new pairing was about to begin.

Subjects took a brief quiz that indicated whether or not they understood the earnings table, and whether or not they understood the constant and independent continuation probability. For the latter, specifically, they were asked what chances out of ten there would be that a pairing would continue if it had already lasted 1, 5, 10, and 100 periods. The answer in each case is eight and the sessions did not begin until all subjects correctly answered the questions. Subjects correctly answering these questions should have a better understanding regarding the stochastic process that governed the end of the supergames and less of a tendency toward the gambler's fallacy.

A total of 44 English-speaking university students in Montreal participated in four experimental sessions.[6] The experiment was programmed and conducted using z-Tree software (Fischbacher 1999). The experiments were run in June and July, 2004, at the Bell Experimental Laboratory for Commerce and Economics at the Centre for Research and Analysis on Organizations. Subjects earned CAD $10.00 for showing up on time and participating fully (which compensated for travel to the off-campus laboratory). Subjects averaged an

[5] The random number generator and seed are available upon request. The minimum supergame length was 1; the maximum supergame length was 25; the average was 4.975.

[6] We use only the first session here.

additional CAD \$18.27 during the experiment that lasted between one and one and a half hours. To control for possible wealth effects accumulating during the (on average) 100 decisions made by each subject, they were paid for the sum of their earnings in a randomly selected five supergames. Alternative opportunities for pay in Montreal for our subjects is considered to be approximately CAD \$8.00 per hour.

5 Results

5.1 Simulation Results

To illustrate our procedure, we generate artificial data by having machines from two populations play Prisoner's Dilemma against each other, and then show what features of the original populations we can recover.

Population 1 is the population described in Section 2. Machine $k = 1$ is 85% grim trigger, machine $k = 2$ is 85% tit-for-tat, and both are equally likely. Population $i = 2$ also has two equally probable machines, both with a single state. Machine $k = 1$ is 60% defect and machine $k = 2$ is 60% cooperate.

We have eight subjects from each population, and each subject from population $i = 1$ plays each subject from population $i = 2$ exactly once. Unlike in the experimental data, we draw an independent supergame duration T for every pair of subjects. As with the experimental data, T is exponential with mean 5, representing a constant continuation probability of 0.8.

Rather than draw machine types randomly from the population, we "stratify" the subjects so that there are four subjects of each machine type in both populations. Specifically, we set $\phi_i(1) = \phi_i(3) = \phi_i(5) = \phi_i(7) = 1$ and $\phi_i(2) = \phi_i(4) = \phi_i(6) = \phi_i(8) = 2$ for $i \in \{1, 2\}$.

We choose machines for population $i = 1$ whose deterministic counterparts (machines playing the probable state with probability one) are supported by equilibrium predictions or past investigations of repeated games (e.g., Axelrod 1985). Population $i = 2$ machines generate variation in play that reveals the machine types of population $i = 1$.

We choose the following prior parameters. We set $\alpha = 1$, implying, for each $K_i \in \{1, 2, 3\}$, a uniform density for $\pi_i | K_i$ on the K_i-dimensional simplex. We set the priors for the numbers of machines states and the numbers of machine types to be uniform on $\{1, 2, 3\}$. That is, $\theta_1 = \theta_2 = \theta_3 = \omega_1 = \omega_2 = \omega_3 = 1/3$ and $\theta_j = \omega_j = 0$ for all $j > 3$. We set $\nu = 0.6$, which favors extreme action probabilities. The density for the 2-dimensional Dirichlet (i.e. the Beta) distribution with parameter vector $(0.6, 0.6)$ is shown in Figure 1.

We present results only for population $i = 1$, the more interesting case.

How well do we recover K_1, the number of machines? The true value is 2, and we obtain the posterior distribution $(0.00, 0.23, 0.77)$ on $\{1, 2, 3\}$. We discover heterogeneity, but we have difficulty learning whether or not there is a third machine type. Since we observe the behavior of only 8 subjects, it is difficult to know how many more than 2 of machine types there are. Since the posterior probability $\Pr[K_1 = 1 | e_1]$ is negligible, we discuss results conditional only on $K_1 = 2$ and $K_1 = 3$. We can think of the whole posterior distribution as being a mixture of the two conditional distributions, with the mixing weights 0.23 and 0.77.

How well do we recover ϕ_1, the machine assignment function? We first condition on $K_1 = 2$. Later, we will condition instead on $K_1 = 3$. Only two partitions of subjects have non-negligible posterior probability. The partition with subjects 1, 3, 5 and 7 assigned to one machine and subjects 2, 4, 6 and 8 assigned to the other (which is the true partition) has posterior probability 0.995. The partition with subjects 1, 2, 3, 5, 7 assigned to one and subjects 4, 6 and 8 to the other has posterior probability 0.005. Clearly, we recover ϕ_1 quite well. We classify all subjects correctly. We are nearly certain about our classification of all subjects except subject 2 and even for this subject, our classification has posterior probability 0.995.

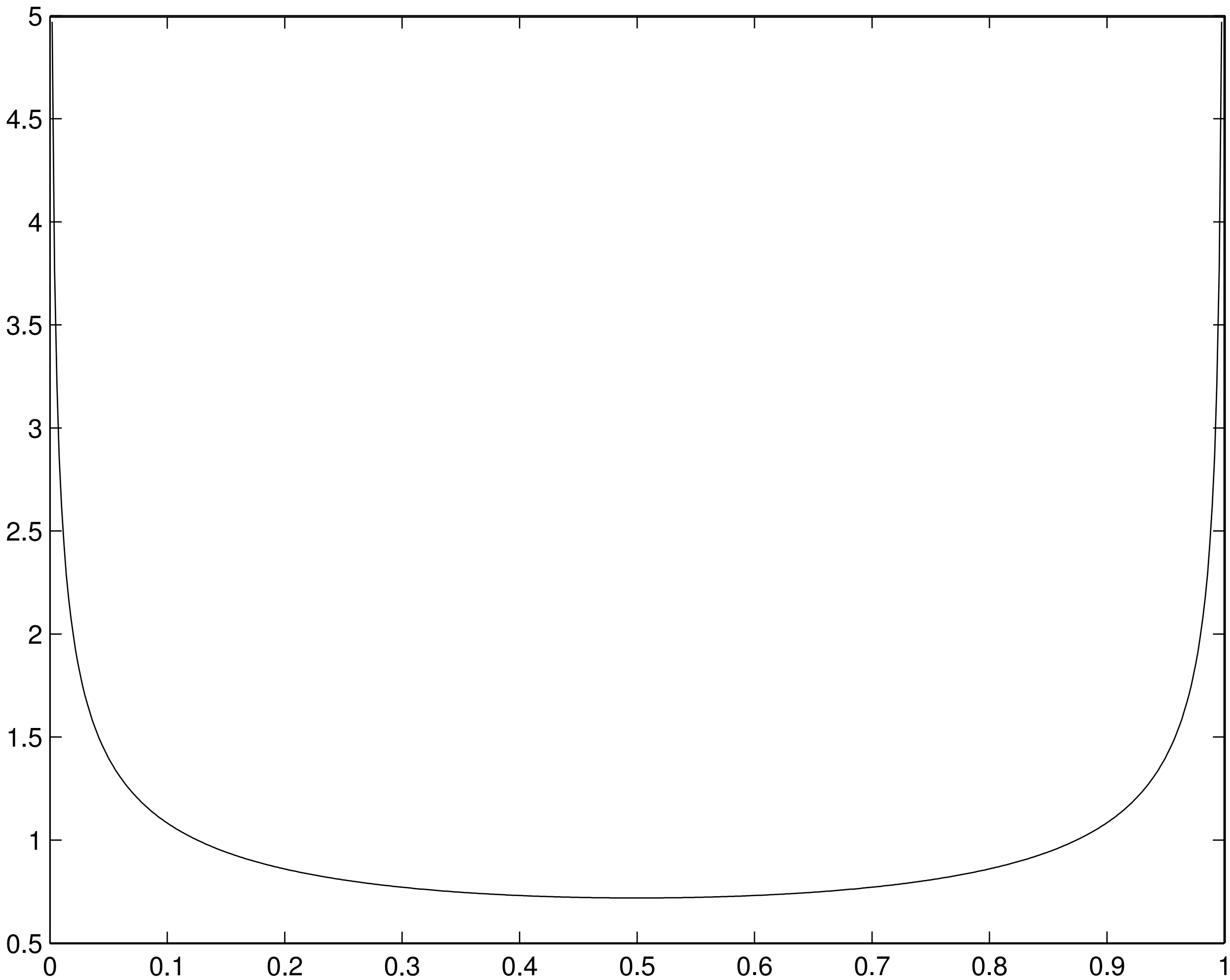

Figure 1: Density of Beta(0.6,0.6) distribution

Table 2: Posterior probabilities $\Pr[\phi_1(s) = k | K_i = 3, e_i]$

Subject (s)	$k = 1$	$k = 2$	$k = 3$
1	0.891	0.000	0.109
2	0.000	0.952	0.048
3	0.767	0.009	0.224
4	0.000	0.998	0.002
5	0.963	0.000	0.037
6	0.000	0.997	0.003
7	0.957	0.000	0.043
8	0.000	0.997	0.003

The central machine assignment, obtained using the method of Section 3.6, is $\bar{\phi}_1(1) = \bar{\phi}_1(3) = \bar{\phi}_1(5) = \bar{\phi}_1(7) = 1$ and $\bar{\phi}_1(2) = \bar{\phi}_1(4) = \bar{\phi}_1(6) = \bar{\phi}_1(8) = 2$. The is exactly the same as the true ϕ_1, but note that it could just as easily have been $\bar{\phi}_1(1) = \bar{\phi}_1(3) = \bar{\phi}_1(5) = \bar{\phi}_1(7) = 2$ and $\bar{\phi}_1(2) = \bar{\phi}_1(4) = \bar{\phi}_1(6) = \bar{\phi}_1(8) = 1$, in which case we would need to permute the results below in the obvious way. According to the method of Section 3.6, we identify as machine $k = 1$ the one with the highest count of subjects 1, 3, 5, and 7 assigned to it plus subjects 2, 4, 6 and 8 *not* assigned to it. Machine $k = 2$ is the other one.

We now condition on $K_1 = 3$. The central machine assignment is $\bar{\phi}_1(1) = \bar{\phi}_1(3) = \bar{\phi}_1(5) = \bar{\phi}_1(7) = 1$ and $\bar{\phi}_1(2) = \bar{\phi}_1(4) = \bar{\phi}_1(6) = \bar{\phi}_1(8) = 2$, as before: $\bar{\phi}$ assigns *no* subject to machine $k = 3$. This identifies machine $k = 1$ as the one with the highest count of subjects 1, 3, 5, and 7 assigned to it plus subjects 2, 4, 6 and 8 not assigned to it; machine $k = 2$ as the one with the highest count of subjects 2, 4, 6, and 8 assigned to it plus subjects 1, 3, 5 and 7 not assigned to it; and machine $k = 3$ as the remaining machine. Table 2 tabulates the posterior probabilities of each assignment.

Results for the posterior distribution of π_1, the machine probability mass function, are somewhat misleading, since we did not draw from any distribution, but instead arranged for equal proportions in the sample. Conditional on $K_1 = 2$, we compute a posterior mean (standard deviation) for $\pi_1(1)$, the probability that a random subject of population 1 is

Table 3: Posterior probabilities $\Pr[Q_1^k = Q|K_1, e_1]$ of machine state numbers

K_1	k	$Q = 1$	$Q = 2$	$Q = 3$
2	1	0.000	0.745	0.254
2	2	0.000	0.892	0.108
3	1	0.000	0.656	0.344
3	2	0.000	0.889	0.111
3	3	0.203	0.375	0.422

machine type 1, of 0.50 (0.15). The relative uncertainty reflects the fact that we only observe eight subjects from the population. Conditional on $K_1 = 3$, we have a posterior mean for $(\pi_1(1), \pi_1(2), \pi_1(3))$ of $(0.416, 0.451, 0.133)$.

How well do we recover numbers of states? The true values for both machines is 2. The posterior distribution of machine state numbers is given in Table 3. Given $K_1 = 2$, we are all but certain that machine $k = 1$ has memory, but still somewhat unsure of the exact complexity. We are nearly sure that machine $k = 2$ also has memory, and somewhat more confident that there are only 2 states. For $K_1 = 3$, the distributions for machines $k = 1$ and $k = 2$ change slightly. Since no subject is assigned to machine $k = 3$ with much probability, we learn little about it: the posterior distribution of its number of states is close to the (uniform) prior. If we further condition on at least one of the eight subjects being assigned to it, the probability that the machine has one state drops almost to zero.

Finally, how well do we recover the state transition functions and action probability mass functions? Once again, we first condition on $K_1 = 2$. Machine $k = 1$ has the exact state transition function of its true value (grim trigger transitions) with posterior probability 0.892. If it does have this transition function, the cooperate probability in the initial state has a posterior mean (standard deviation) of 0.876 (0.026), and the defect probability in the other state has a posterior mean (standard deviation) of 0.864 (0.039). Recall that both true values are 0.85. No other transition function has posterior probability greater than 0.02.

The prior probability of each of the 240 unique 2-state transition functions is 0.00139.

Machine $k = 2$ has exact tit-for-tat transitions (the true transitions) with probability 0.892. With this transition function, the cooperate probability in the initial state has a posterior mean (standard deviation) of 0.863 (0.049), and the defect probability in the other state has a posterior mean (standard deviation) of 0.865 (0.135). No other transition function has probability greater than 0.02.

Now we condition on $K_1 = 3$. Machine $k = 1$ has exact grim trigger transitions with posterior probability 0.589. With this transition function, the cooperate probability in the initial state has a posterior mean (standard deviation) of 0.865 (0.037), and the defect probability in the other state has a posterior mean (standard deviation) of 0.857 (0.134). With another 0.058 probability, the transition function for machine $k = 1$ is like grim trigger except that the machine returns to the initial state if it cooperates against a defecting opponent. Since cooperating in this state is the unlikely action, we have less data informing us about this state transition than we have informing us about other transitions. No other transition function has posterior probability greater than 0.02.

Machine $k = 2$ has exact tit-for-tat transitions with posterior probability 0.886. With this transition function, the cooperate probability in the initial state has a posterior mean (standard deviation) of 0.875 (0.027), and the defect probability in the other state has a posterior mean (standard deviation) of 0.869 (0.131). No other transition function has posterior probability greater than 0.02.

Machine $k = 3$ has probability 0.203 of having 1 state and therefore no non-degenerate transitions, probability 0.034 of having exact grim trigger transitions, probability 0.026 of having the same near-grim-trigger transitions as above and probability 0.015 of having exact tit-for-tat transitions. No other transition function has posterior probability greater than 0.02.

5.2 Experimental Results

We use the same prior for inferring populations from experimental data. Because of the symmetry of the game, we treat all experimental subjects as being from population 1.

The posterior distribution, on $\{1, 2, 3\}$, for the number of machines K_1 is $(0.00, 0.28, 0.72)$. We are nearly certain that there is heterogeneity, but less sure about the degree. Given the small number of subjects, this is understandable. Again and for the same reasons, we ignore the case $K_1 = 1$ and treat the cases $K_1 = 2$ and $K_1 = 3$ separately.

The central machine assignment given $K_1 = 2$ is

$$\bar{\phi}_1(1) = \bar{\phi}_1(2) = \bar{\phi}_1(3) = \bar{\phi}_1(4) = \bar{\phi}_1(5) = 1 \quad \bar{\phi}_1(6) = \bar{\phi}_1(7) = \bar{\phi}_1(8) = 2,$$

which establishes machine $k = 1$ as the machine associated with subjects 1 through 5 and machine $k = 2$ as the machine associated with subjects 6 through 8. The central machine assignment given $K_1 = 3$ is

$$\bar{\phi}_1(1) = \bar{\phi}_1(2) = \bar{\phi}_1(3) = \bar{\phi}_1(4) = \bar{\phi}_1(5) = 2 \quad \bar{\phi}_1(6) = \bar{\phi}_1(7) = \bar{\phi}_1(8) = 3,$$

with no subjects assigned to machine $k = 1$. We see that machine $k = 2$ tends to be assigned the same subjects as machine $k = 1$ for $K_1 = 2$, and machine $k = 3$ tends to be assigned the same subjects as machine $k = 2$ for $K_1 = 2$. Table 4 tabulates posterior probabilities of assignments for both $K_1 = 2$ and $K_1 = 3$. Some of the features of the joint distribution are obscured by reporting only these marginal distributions. For example, the assignments of subjects 1 and 3 are highly correlated. If one is assigned the "5 through 8" machine, then the other is more likely to be so assigned.

For the posterior mean of machine type probabilities, we have $E[(\pi(1), \pi(2)|K_1 = 2, e_1]$ $= (0.609, 0.391)$ and $E[(\pi(1), \pi(2), \pi(3)|K_1 = 2, e_1] = (0.117, 0.528, 0.355)$.

The posterior distribution of machine state numbers is given in Table 5. As with the simulated data, we are quite sure that all machines have memory. Again we have a machine,

Table 4: Posterior probabilities $\Pr[\phi_1(s) = k|K_i, e_i]$ of state assignments

K_1	Subject (s)		$k = 1$	$k = 2$
2	1		0.965	0.035
2	2		1.000	0.000
2	3		0.970	0.030
2	4		1.000	0.000
2	5		1.000	0.000
2	6		0.000	1.000
2	7		0.000	1.000
2	8		0.153	0.847
		$k = 1$	$k = 2$	$k = 3$
3	1	0.040	0.887	0.073
3	2	0.058	0.942	0.000
3	3	0.038	0.897	0.065
3	4	0.017	0.982	0.001
3	5	0.078	0.922	0.000
3	6	0.016	0.000	0.984
3	7	0.004	0.001	0.995
3	8	0.035	0.182	0.783

Table 5: Posterior probabilities $\Pr[Q_1^k = Q | K_1, e_1]$ of state numbers

K_1	k	$Q = 1$	$Q = 2$	$Q = 3$
2	1	0.000	0.000	1.000
2	2	0.000	0.538	0.461
3	1	0.254	0.319	0.426
3	2	0.000	0.002	0.998
3	3	0.000	0.516	0.484

this time $k = 1$, having a low probability of any subject being assigned to it. As before, the posterior distribution of its number of states differs little from the prior distribution, but conditioning on some subject being assigned to it, the probability of having only one state drops to nearly zero.

Given $K_1 = 2$, several state transition functions have posterior probability greater than 0.02. High probability transition functions for $k = 1$ come in three clusters, illustrated in Table 6. Entries in the table give either the state value in $\{1, 2, 3\}$ or an X indicating a "don't care". A cluster in the table consists of all transition functions obtained by independently assigning values to the don't cares.

Cluster 1 consists of nine transition functions that agree on all transitions except $\lambda((c, C), 2)$ and $\lambda((d, D), 3)$. The posterior probabilities of the nine transition functions in cluster 1 range from 0.056 to 0.060, and together account for 0.525 of posterior probability. Numerical standard errors suggest that these probabilities are different and not due to numerical sampling error, but this is not conclusive. Cluster 2 consists of three transition functions that agree on all transitions except $\lambda((d, D), 3)$. Their posterior probabilities range from 0.028 to 0.029 and together account for 0.086 of posterior probability. Cluster 3 consists of nine more transition functions, with probabilities ranging from 0.024 to 0.026 and accounting for a posterior probability of 0.224.

The three clusters together account for 0.835 of posterior probability. The 21 high probability transition functions, like all 343,000 state transition functions with three states, each have a prior probability of 9.71×10^{-7}.

Not only are the transition functions in each cluster very similar, the clusters themselves are quite similar. All 21 high probability transition functions agree on eight of the twelve transitions, including all four transitions from the initial state.

The action probability mass functions associated with these 21 high probability transition functions are very similar. The posterior mean (standard deviation) of the cooperate probability is approximately 0.42 (0.05) in the initial state, 0.03 (0.01) in state 2 and 0.95 (0.03) in state 3.

The $k = 1$ machine exhibits at least three interesting behavioral characteristics. First, it is complex, consisting of three states. Second, it has a buffer state (state 1) that delays entering the defect state (2) after a defection is observed and the cooperative state (3) after cooperation is observed. This state might be thought of as a cautious state: the machine does not always immediately punish a defection, but sometimes waits for a second consecutive defection before punishing. Likewise, the machine sometimes waits for a second act of cooperation before starting to cooperate itself. Second, it exhibits reciprocity, a feature for which there is much experimental evidence (e.g., Fehr and Gaechter, 2000). Third, cooperative actions result in cooperative responses with high probability, and non-cooperative actions are likely to engender non-cooperative actions.

High probability transition functions for $k = 2$ come in one cluster, illustrated in Table 7. The cluster consists of 16 transition functions, which agree on 4 transition functions and differ on the other 4. The cluster includes the tit-for-tat but not the grim trigger transition function. Their probabilities range from 0.030 to 0.033 and together account for 0.495 of posterior probability. The action probability mass functions associated with these 16 high probability transition functions are again very similar. The posterior mean (standard deviation) of the cooperate probability is approximately 0.02 (0.01) in the initial state, and

Table 6: Three high probability clusters for machine $k = 1$

Cluster	Action (a)	$\lambda(a,1)$	$\lambda(a,2)$	$\lambda(a,3)$
	(d, D)	2	2	X
1	(c, D)	2	2	1
	(d, C)	3	1	3
	(c, C)	3	X	3
	(d, D)	2	2	X
2	(c, D)	2	2	2
	(d, C)	3	1	3
	(c, C)	3	3	3
	(d, D)	2	2	X
3	(c, D)	2	2	1
	(d, C)	3	1	1
	(c, C)	3	X	3

Table 7: One high probability cluster for machine $k = 1$

Action (a)	$\lambda(a,1)$	$\lambda(a,2)$
(d, D)	1	X
(c, D)	X	1
(d, C)	2	X
(c, C)	2	X

0.98 (0.03) in the other. Note that in the initial state, defection is the high probability action. Usually the initial state in a tit-for-tat machine is thought of as a cooperate state.

Machines $k = 1$ and $k = 2$ both have, with high posterior probability, a tit-for-tat quality of imitating the opponent's action in the previous period.

We don't report results conditional on $K_1 = 3$ except to point out that machine $k = 2$ is very similar to the $k = 1$ machine for $K_1 = 2$ and machine $k = 3$ is very similar to the $k = 2$ machine for $K_1 = 2$.

6 Conclusions

We presented a new method to draw inference about the number and types of repeated-game strategies from choice data in experiments. We base our strategy model on finite automata, which model a broad array of strategic behavior.

We demonstrated our ability to recover features of machines, commonly found in repeated game theory, that generate artificial data. We generated an amount of data similar to what we might expect to collect in the laboratory. We showed that our procedure does well recovering the characteristics of the underlying machines. We applied our procedure to new experiments in which subjects simply played many indefinitely repeated prisoner's dilemma games. The repeated game admits both cooperative and noncooperative equilibria.

The results give us a new picture of play in repeated prisoner's dilemma games. We find strong evidence for heterogeneity. We find evidence that people use strategies that punish and reward behavior of the opponent. And interestingly, the strategies we infer do not contain harsh enough punishments to support cooperation in equilibrium.

Our strategy model and inference procedure open doors for new avenues of investigation of play in repeated games. For example, we can study equilibrium selection in a new (and more natural) way by examining the expected payoff of inferred strategies when played against the population, and comparing this to the expected payoff for the best-response to the inferred population. And we are currently examining empirically based strategy models across several different types of stage games in an effort to discover behavioral regularities between them.

For another example, with the ability to base repeated-game strategies on empirical observation we are poised for fresh contributions to the literature on learning in games (Camerer and Ho, 1999; Cheung and Friedman, 1997; Erev and Roth, 1998), in which boundedly rational learning models are employed to understand and predict play over time (much longer periods of time than the in the supergames in our experiments). These dynamic models of behavior assume that subjects make choices from a set of strategies that typically contains only stage-game actions. However, it seems likely that subjects are learning to play repeated-game strategies. We should be able to augment the learning models with the repeated-game strategies we recover using our procedure, thus broadening the class of games to which the learning theories are applicable, and improving our predictions of strategic choice behavior in dynamic environments.

Our strategy model is quite general; it can be applied to games with multiple actions and player types, and it contains both equilibrium and non-equilibrium strategies. As such we aim to make our procedure available as a tool to a least compliment existing methods of statistical inference for a wide class of games. With our procedure we take a step toward a deeper understanding of the ghost in repeated-game machine.

References

[1] Aoyagi, M. and G. Frechette (2004) "Collusion in Repeated Games with Imperfect Monitoring", working paper, Harvard Business School.

[2] Aumann, R. (1981) "Survey of Repeated Games", in R.J. Aumann et al., (eds), *Essays in Game Theory and Mathematical Economics in Honor of Oscar Morgenstern*, Bibliographisches Institut, Zurich, Mannheim, 11-42.

[3] Axelrod, R. (1984) "*The Evolution of Cooperation*, in Basic Books, New York.

[4] Binmore, K. and L. Samuelson (1992) "Evolutionary stability in repeated games played by finite automata", *Journal of Economic Theory*, 57:2, 278-305.

[5] Camerer, C. and T.H. Ho (1999), "Experience-weighted attraction (EWA) learning in normal-form games", *Econometrica*, 67:827-874.

[6] Cheung Y.W. and D. Friedman (1997) "EIndividual Learning in Normal Form Games: Some Laboratory Results", *Games and Economic Behavior*, 19:46-76.

[7] Costa-Gomes, M., V. Crawford, and B. Broseta (2001) "Cognition and Behavior in Normal-form Games: An Experimental Study", *Econometrica*, 69:5, 1193-1235.

[8] El-Gamal, M. and D. Grether (1995) "Are People Bayesian? Uncovering Behavioral Strategies", *Journal of the American Statistical Association*, 90:432, 1137-1145.

[9] Engle-Warnick, J. and R. Slonim (2003) "Inferring Repeated-Game Strategies from Actions: Evidence from Trust Game Experiments", working paper, McGill University.

[10] Erev, I. and A. Roth (1998) "Predicting How People Play Games: Reinforcement Learning in Experimental Games with Unique, Mixed Strategy Equilibria", *American Economic Review*, 88:848-881.

[11] Feher, E. and S. Gaechter (2000) "The Economics of Reciprocity", *Journal of Economic Perspectives*, 15:159-181.

[12] Fischbacher, U. (1999), "z-Tree - Zurich Toolbox for Readymade Economic Experiments - Experimenter's Manual", Working Paper Nr. 21, Institute for Empirical Research in Economics, University of Zurich.

[13] Geweke, J. and M. Keane (1999) "Bayesian Inference for Dynamic Discrete Choice Models Without the Need for Dynamic Programming", in Mariano, Schuermann and Weeks, (eds), *Simulation Based Inference and Econometrics: Methods and Applications*, Cambridge University Press, Cambridge.

[14] Geweke, J. and M. Keane (2001) "Computationally Intensive Methods for Integration in Econometrics", in J. Heckman and E. Leamer, (eds), *Handbook of Econometrics*, Vol. 5, North Holland, 3463-3568.

[15] Houser, D., M. Keane and K. McCabe (2002) "Behavior in a Dynamic Decision Problem: An Analysis of Experimental Evidence Using a Bayesian Type Classification Algorithm", *Econometrica*, 72:3, 781-822.

[16] Johnson, E., C. Camerer, S. Sen, and T. Rymon (2001) "Detecting Failures of Backward Induction: Monitoring Information Search in Sequential Bargaining", *Journal of Economic Theory*, 104:1:16-47.

[17] McKelvey, R. and T. Palfrey (1995) "Quantal Response Equilibria for Normal Form Games", *Games and Economic Behavior*, 10:6-38.

[18] Miller, J. (1996) "The Coevolution of Automata in the Repeated Prisoner's Dilemma", *Journal of Economic Behavior and Organization*, 29:1, 87-112.

[19] Rubinstein, A. (1986) "Finite Automata Play the Repeated Prisoner's Dilemma", *Journal of Economic Theory*, 38, 83-96.

[20] Selten, R., M. Mitzkewitz and G. R. Ulrich (1997) "Duopoly Strategies Programmed by Experienced Traders", *Econometrica*, 65, 517-555.

[21] Stahl, D. and P. Wilson (1995) "On Players' Models of Other Players: Theory and Experimental Evidence", *Journal of Economic Behavior and Organization*, 10, 218-254.

[22] Tanner, M. A. and W. H. Wong (1987) “The calculation of posterior distriubtions by data augmentation”, *Journal of the Americal Statistical Association*, 82, 528-549.