Markov Games of Incomplete Information for Multi-Agent Reinforcement Learning

Liam Mac Dermed, Charles L. Isbell, Lora Weiss
Georgia Institute of Technology
Robotics and Intelligent Machines Laboratory
Atlanta, Georgia 30332
liam@cc.gatech.edu, lora.weiss@gtri.gatech.edu, isbell@cc.gatech.edu

Abstract

Partially observable stochastic games (POSGs) are an attractive model for many multi-agent domains, but are computationally extremely difficult to solve. We present a new model, Markov games of incomplete information (MGIIs) which imposes a mild restriction on POSGs while overcoming their primary computational bottleneck. Finally we show how to convert a MGI into a continuous but bounded fully observable stochastic game. MGIIs represents the most general tractable model for multi-agent reinforcement learning to date.

Introduction

Multi-agent reinforcement learning (MARL) poses the problem of how to act optimally in scenarios with multiple self-serving agents. This is an important problem across a number of different domains, including: multi-robot interaction, reputation systems, evolutionary biology, mechanism design, etc. When the world is fully observable this problem is challenging. When this world is only partially observable (as is the case in many problems), the problem is overwhelming. Previous research has modeled these scenarios as partially observable stochastic games (POSG). POSGs are extremely complicated and attempts to solve POSGs have all met large computational barriers. In this paper we argue that POSGs are too general. Instead we present a slightly weaker model – Markov games of incomplete information (MGI). The huge advantage of MGIIs over POSGs is that MGIIs can be converted into a fully observable stochastic game, allowing them to be solved (much like POMDPs being converted into MDPs). MGIIs can approximate POSGs and exactly model many interesting problems while being (relatively) easier to solve.

Background: Multi-agent Models

We assume the reader is familiar with Markov decision processes (MDPs), partially observable MDPs (POMDPs), and normal form games. We start by giving the definition of stochastic games (SGs) followed by a definition of partially observable stochastic games (POSGs) and normal form games of incomplete information (Bayesian games). We then use these models to present a new model – Markov games of incomplete information (MGIIs). In the process we prove various properties of these models. In the next section we show how to convert MGIIs into SGs of complete information over a continuous but bounded belief space. In turn these continuous SGs can be discretized and solved using existing algorithms.

Multi-agent models extend single agent decision problems by using vectors (one element per player) to represent actions, rewards, beliefs, or observations rather than the scalars used in single agent models. These vectors are often referred to as joints of the individual variables. In our work, we assume agents are rational and attempt to maximize their long term expected utility with discount factor \( \gamma \) when decision problems are sequential. It is useful to refer to all elements of the joint except one particular player. For a joint-action \( \vec{a} \) where \( a_i \in A_i \) we refer to all actions except for player \( i \)'s as \( \vec{a}_{-i} \) and the set of possible such joint-actions as \( A_{-i} = \Pi_{j \in A_j \neq A_i} A_j \). If all other players besides player \( i \) take joint-action \( \vec{a}_{-i} \) and player \( i \) takes action \( \alpha \) then we write the full joint-action as \( \vec{a}_{-1, \alpha} \). A similar notation is used for types and observations.

Stochastic Games

Stochastic or Markov games (Littman 1994) extend MDPs to multiple agents. A stochastic game (SG) is fully observable and operates like an MDP: an agent starts in an initial state followed by repeatedly choosing actions, receiving rewards, and transitioning to new states. The difference is that there are multiple agents, each of which is independently choosing actions and receiving rewards while the state transitions with respect to the full joint-action of all players. Formally a SG consists of a tuple \( SG = \langle N, A, S, P, R, s(0) \rangle \).

- \( N \) is the set of players. \( n = |N| \)
- \( A = \Pi_{i=1}^n A_i \) is the set of joint-actions
- \( S \) is the set of states
- \( P : S \times A \rightarrow \Delta(S) \) is the probability transition function with \( P(s'|s, \vec{a}) \) being the probability of ending up in state \( s' \) after taking joint-action \( \vec{a} \) in state \( s \)
- \( R : S \times A \rightarrow \mathbb{R}^n \) is the joint-reward function
- \( s(0) \in S \) is the initial state
An example stochastic game is given in Figure 2. This game has four states with two terminal states. In the two middle states play alternates between the two players until one of the players decides to exit the game. This game is notable because there are no pure policies (players must randomize between actions) despite the fact that only one player acts in each of the two non-terminal states.

Figure 1: The Breakup Game. Circles represent states, outgoing arrows represent deterministic actions. Unspecified rewards are zero.

Figure 2: The final achievable-sets for the breakup game ($\gamma = 0.9$). The state where player 1 moves is shown on the left while the state where player 2 moves is shown on the right.

The key idea needed to extend reinforcement learning into multi-agent domains is to replace the value-function, $V(s)$, in Bellman’s dynamic program with an achievable set function $V : S \rightarrow \{\mathbb{R}^n\} -$ a mapping from state to achievable-set. As a group of $n$ agents follow a joint-policy, each player $i \in I$ receives rewards. The discounted sum of these rewards is player’s utility, $u_i$. The vector $\vec{u} \in \mathbb{R}^n$ containing these utilities is known as the joint-utility, or value-vector. Thus, a joint-policy yields a joint-utility in $\mathbb{R}^n$. If we examine all mixed joint-policies starting from state $s$, discard those not in equilibrium, and compute all the joint-utilities of the remaining policies we will have a set of points in $\mathbb{R}^n$: the achievable set. An achievable set contains all possible joint-utilities that players can receive using policies in equilibrium. Each dimension represents the utility for a player (Figure 3). Figure 2 shows the final feasible-sets for the breakup game.

The achievable set of correlated equilibrium for SGs can be efficiently approximated to within bounded error using the QPACE algorithm (MacDermid et al. 2010). The algorithm scales linearly with respect to the number of states and polynomially with respect to the number of joint-actions, but exponentially with respect to the number of players.

Figure 2: The final achievable-sets for the breakup game ($\gamma = 0.9$). The state where player 1 moves is shown on the left while the state where player 2 moves is shown on the right.

The POSG framework is incredibly general and thus very difficult to solve. The authors know of no tractable attempt to compute optimal (or even reasonable) policies for general POSGs. Much work has been done on limited subsets of POSGs. Game theory has dealt heavily with games of incomplete information (a.k.a. Bayesian games) which can be thought of as finite horizon POSGs, but has typically had a descriptive focus instead of prescriptive algorithms that compute policies. Game theory has also dealt with stochastic games but very few game theoretic results exist at the intersection of stochastic and incomplete information games. Those results that do exist almost exclusively deal with two player zero-sum repeated games with incomplete information (Gilpin and Sandholm 2008). The multi-agent learning community has also studied the two player zero-sum case with much success (particularly for Poker (Sandholm 2010)), and is an active area of research. However, these results make extensive use of properties unique to zero-sum games and don’t extend to the general case. Likewise successes have been made in MARL for games of common payoff (Seuken and Zilberstein 2007), but these also do not generalize.

**Partially Observable Stochastic Games**

A partially observable stochastic game (POSG) is the multi-agent equivalent to a POMDP. A POSG is identical to a POMDP except instead of a single action, observation, and reward there is one for each player which are expressed together as a joint-action, joint-observation, and joint-reward. POSG consists of a tuple $PO = (N, A, S, O, P, R, s^{(0)})$.

- $N$ is the set of players. $n = |N|$.
- $A = \prod_{i=1}^{n} A_i$ is the set of joint-actions.
- $S$ is the set of states.
- $O = \prod_{i=1}^{n} O_i$ is the set of joint-observations.
- $P : S \times A \rightarrow \Delta(S \times O)$ is the probability transition function with $P(s', o|s, a)$ being the probability of ending up in state $s'$ with observations $\vec{o}$ after taking joint-action $\vec{a}$ in state $s$.
- $R : S \times A \rightarrow \mathbb{R}^n$ is the joint-reward function.
- $s^{(0)} \in \Delta(S)$ is the initial state distribution.

The POSG framework is incredibly general and thus very difficult to solve. The authors know of no tractable attempt to compute optimal (or even reasonable) policies for general POSGs. Much work has been done on limited subsets of POSGs. Game theory has dealt heavily with games of incomplete information (a.k.a. Bayesian games) which can be thought of as finite horizon POSGs, but has typically had a descriptive focus instead of prescriptive algorithms that compute policies. Game theory has also dealt with stochastic games but very few game theoretic results exist at the intersection of stochastic and incomplete information games. Those results that do exist almost exclusively deal with two player zero-sum repeated games with incomplete information (Gilpin and Sandholm 2008). The multi-agent learning community has also studied the two player zero-sum case with much success (particularly for Poker (Sandholm 2010)), and is an active area of research. However, these results make extensive use of properties unique to zero-sum games and don’t extend to the general case. Likewise successes have been made in MARL for games of common payoff (Seuken and Zilberstein 2007), but these also do not generalize.
Types and Beliefs

Agents living in a POSG view their world as lacking the Markov property; agents must utilize past observations to make optimal decisions. Naively an agent can retain the history of all observations received. However, such retention is impossible in the long run. In order to understand how to efficiently solve games of incomplete information we need to understand how to compactly represent the useful information of an agent’s history for decision making. We call an agent’s current sum of private information that agent’s type. The set of possible types is referred to as the type space. Types should not be confused with beliefs, which include the sum of private and public information. We make this distinction as public information is much easier to represent compactly. Note that in the single agent case there is no distinction between types and beliefs, as there is no notion of public vs. private information.

For POMDPs a sufficient (for optimal behavior) type space is the set of probability distributions over possible states. This is not the case for POSGs. Not only do agents have to worry about which state they are in, but also must worry about other player’s beliefs. Worse, players must reason about the beliefs that players hold about each other’s beliefs. These levels of meta-reasoning are called nested beliefs, and can potentially continue indefinitely (reasoning about beliefs of beliefs of beliefs etc...). This additional level of meta-reasoning makes the problem significantly more conceptually and computationally complex. An agent’s observations may include private information about other player’s beliefs, and in the worst case it might be impossible to losslessly reduce a player’s type beyond the full history. We refer to an agent’s belief about the states-of-nature as that agent’s level-zero belief (a probability distribution over states). To prevent having to reason over infinite levels of nested beliefs, we can assume a level of common knowledge. If a fact is common knowledge then all players not only know that fact, they know that other player’s know it and that all such nested knowledge is known. Most game theoretic results assume prior common knowledge.

There have been previous approaches that attempt to operate directly on the infinitely nested belief structure (Doshi and Gmytrasiewicz 2009), but these must necessarily be approximations of unknown accuracy (if we stop at the \( n^{th} \) nested belief the \( n^{th} + 1 \) could dramatically change the outcome). These results have gotten reasonable empirical results in a few limited domains but it seems unlikely these methods will eventually generalize to general-sum games or scale with the complexity of the game. The POSG model typically assumes that the model itself along with the initial state distribution \( s^{(0)} \) is common knowledge at the beginning of the game. At this initial point, players only have level-zero beliefs. However, for every transition of the game the non-common-knowledge belief levels increase by one. Every possible history could potentially lead to a different set of nested-beliefs, and thus the type-space grows combinatorially. Our solution, MGIIs, are defined to prevent this explosion of type-space and bound the belief-space.

Bayesian Games

A Bayesian game is the partially observable equivalent of a normal form game. A Bayesian stage-game is the implicit Bayesian game played at a given time step of the POSG. It consists of a tuple \( BG = (N, A, \Theta, \tau, R) \)

- \( N \) is the set of players, \( n = |N| \)
- \( A = \Pi_{i=1}^{n} A_i \) where \( A_i \) is the actions for player \( i \)
- \( \Theta = \Pi_{i=1}^{n} \Theta_i \) is the set of types (one for each different possible private signal \( \theta_i \in \Theta_i \))
- \( \tau \in \Delta(\Theta) \) is a probability distribution over joint types \( \Theta \) assigning \( \theta \) w.p. \( \tau(\theta) \)
- \( R : \Theta \times A \rightarrow \mathbb{R}^n \) is the reward function

There are a number of different game theoretic solutions to Bayesian games, each of which make slightly different assumptions (Forges 2006). We distinguish between three different dimensions of solutions and argue for a particular solution concept based on which assumptions we believe to be more general and applicable.

The first distinction we make is between Nash and correlated equilibria. Nash equilibria (NE) assume players choose actions independently while correlated equilibria (CE) allow player’s actions to be dependent via a shared random variable (and thus conditionally independent given the random variable). CE are more general than NE (all NE are CE) and thus will produce better results as for any NE there will be a CE that Pareto dominates it. CE are also computationally much less demanding to compute. Therefore agents have an incentive to use CE instead of NE when available. While CE require a shared random variable (a.k.a. correlation device), such a requirement is not particularly demanding as agents in most real world scenarios can find or create such variables (e.g., electromagnetic or sound noise at a particularly volatile frequency, sunspots, a dice roll, a mediator, or cryptographic techniques using communication). It might seem implausible for agents to have the ability to agree on a particular shared random variable but if they can’t select a random variable then they also won’t be able to perform equilibrium selection or for that matter even agree on a solution concept. Determining a correlation device as well as equilibrium selection and solution concept agreement are thus all part of the same problem and out of the scope of this paper (as it is implicitly for all pure game theoretic methods). Because the requirements are low and reasonable while providing superior results at lower computational complexity, we will focus on correlated equilibria.

Within correlated equilibria concepts a second distinction is made with regard to the level of communication permitted. Unlimited communication which incurs no cost to the agents (cheap-talk) is sometimes present. In games of incomplete information cheap-talk can expand the set of equilibria as agents can potentially share their private information with each other. This leads to a solution concept known as communication equilibrium (Forges 2006). However the problem of determining a communication policy (without an unbiased mediator) is itself a complicated problem with doubts that it is possible at all (Kar, Ray, and Serrano 2010).
We also want to maintain as general a solution as possible so we believe that any communication should be explicitly included in the model. We will therefore assume agents have access to a shared random variable, but we will not assume cheap-talk (thus agents can not inform each other about their private information within a stage-game, only explicitly through their actions across stages).

The third distinction we make is between strategic and agent form solutions. Strategic form solutions assume that a player simultaneously controls the policy of all types of that player while agent form treats each player type as a separate player. Strategic form allows multiple types of a player to switch their policies simultaneously while agent normal form only allows a single player type to change their actions. In effect, strategic form must guard against coalitions between types of the same player, and is thus more restrictive than agent form (all strategic form equilibria are agent form equilibria, but not visa-versa). Agent-normal form will be more conservative than agent form (all strategic form equilibria are agent form equilibria, but not visa-versa). Agent-normal form will therefore provide better results. Agent-form is also easier to compute for these two reasons we believe the agent-normal form is a more appropriate solution concept.

Taking into account the three distinctions listed above we believe the appropriate solution concept for each Bayesian stage-game is the agent-normal form correlated equilibrium. For a description of various Bayesian-Nash solution concepts see (Myerson 1991), while other correlated solution concepts are examined in (Forges 2006).

An agent normal form correlated equilibrium takes the form of a probability distributions across actions, one for each joint-type, such that agent-types don’t have an incentive to deviate, and that a player’s action is conditionally independent of the types of other players (enforcing that no private information is revealed by the shared random variable). We use the notation defined in section 3 to give a formal definition:

**Definition 0.0.1.** A set of probability distributions $P \in \Theta \times \Delta(A)$ is an agent normal form correlated equilibrium of Bayesian game $(N, A, \Theta, \tau, R)$ iff:

For each player $i$ with type $\theta_i$, distinct actions $a, b \in A_i$,

$$\sum_{\bar{a}, \bar{\theta}_{-i}} \sum_{\bar{a}, \bar{\theta}_{-i}} \tau(\theta_{-i,i})P_{\theta_{-i,i}}(\bar{a}_{-i,\alpha})R(\theta_{-i,i}, \bar{a}_{-i,\alpha}) \geq$$

$$\sum_{\bar{a}, \bar{\theta}_{-i}} \sum_{\bar{a}, \bar{\theta}_{-i}} \tau(\theta_{-i,i})P_{\theta_{-i,i}}(\bar{a}_{-i,\alpha})R(\theta_{-i,i}, \bar{a}_{-i,\beta})$$  \hspace{1cm} (1)

For all players $i, j$ with types $\theta_i, \theta_j$, and action $\alpha$ for player $i$: $P_{\theta_j} = P_{\theta_j}(\theta_i, \alpha)$, or:

$$\sum_{\bar{a}, \bar{\theta}_{-ij}} \sum_{\bar{a}, \bar{\theta}_{-ij}} P_{\theta_{-ij,i,j}}(\bar{a}_{-i,\alpha}) =$$

$$\sum_{\bar{a}, \bar{\theta}_{-ij}} \sum_{\bar{a}, \bar{\theta}_{-ij}} \tau(\theta_{ij,i,j})$$  \hspace{1cm} (2)

**POSGs as a Sequence of Bayesian Games**

A stochastic game can be thought of as a model of the world where agents play a sequence of normal-form games where the next game in the sequence depends on the previous game and the actions taken. A SG can then be solved by solving each normal-form stage game augmented to include expected utility. It would be nice to use the same process to solve a POSG by turning it into a series of Bayesian games. Unfortunately unlike how a SG can easily be seen as a sequence of normal-form games, a POSG does not naturally correspond to a sequence of Bayesian games. However, a non-trivial transformation can be applied to a POSG to compute successive Bayesian games which have the same decision theoretic properties as the original POSG. Emery-Montemerlo et.al [2004] presented a method for achieving this transformation. The method keeps track of two distributions and updates them each time-step: the probably of seeing a particular observation history $\theta(1)$ (which are defined as the player types) and the probability that the underlying state is $s(1)$ given $\theta(1)$.

After $t + 1$ time steps the current Bayesian game, that is equivalent to the decision problem faced by players in the POSG, can be computed as follows: $N$ and $A$ are the same as in the POSG. The player types, $\Theta(1)$, are all possible observation histories as discussed in section 4. The common knowledge distribution over types, $\tau(1)$, can be computed recursively given the sequence of states and actions as follows. The probability that joint-type $\theta(1) = (\bar{\theta}_1, \theta(1))$ at time $t + 1$ is given by:

$$\tau(t+1)(\bar{\theta}, \theta(t)) = \tau(t)(\bar{\theta})Pr[\bar{\theta}|\theta(t)]$$  \hspace{1cm} (3)

$$Pr[\bar{\theta}|\theta(t)] = \sum_{s(t+1)} P(s(t+1), \bar{\theta}(t), \theta(t))Pr[\theta(t)|s(t)]$$  \hspace{1cm} (4)

Because $\theta(t)$ is the history of joint observations we can compute $P(s\theta(t))$ by treating the POSG as a hidden Markov model and performing filtering with observations $\theta(t)$ (recall that $\theta(t)$ includes observed joint-actions). We can compute the one step reward function:

$$R(t+1)(\bar{\theta}, \theta(t)) =$$

$$\sum_{s(t)} Pr[\theta(t)|s(t)] \sum s' P(s'|s(t), \bar{a}(t))R(s', \bar{a}(t))$$  \hspace{1cm} (5)

This reward function provides the short-term stage-game reward, however agents actually want to maximize their long term rewards (utility). Calculating the exactly utility of successor states is very difficult. A standard trick is to instead use estimated utilities of each successor state, $V((\bar{\theta}, \theta(t)))$. Using $V$ the expected utility for each joint-action $\bar{a}$ can be calculated resulting in the augmented reward $R(\bar{a}, \theta) = \sum s P(s|\theta)R(s, \bar{a}) + \sum \bar{a} V((\bar{a}, \theta(t))) P(\bar{a}|\theta)$. The challenge then lies in computing the state utility estimations $V((\bar{\theta}, \theta(t)))$. A common tactic when the world is fully observable is to apply dynamic programming and iteratively compute $V$ based on the previous estimation of $V$ (e.g. value iteration). Unfortunately the state-space of Bayesian games is unbounded as it includes all possible histories, making it impossible to iterate over all states and thus infeasible to compute $V$ in this way. If we did have a $V$ we would have a Bayesian game and could compute equilibria of this game. Emery-Montemerlo et.al 2004 constructed an
approximation of $V$ by removing all types with low probability combined with a heuristic to guess $V$. While this provides a tractable algorithm that might perform reasonably on some problems if the heuristic is good and types don’t become too diluted, it likely will produce arbitrarily poor solutions to general POSG problems.

Our solution is to define a new model, Markov games of incomplete information (MGIIs), that by definition will produce a bounded type-space for the Bayesian stage-games. This will allow $V$ to be estimated using dynamic programming with existing algorithms.

**Markov Games of Incomplete Information (MGIIs)**

A MGIIs is defined as a sequence of Bayesian games where both the next game and the types of players in that game are random variables dependent on the previous game and type. This is in contrast to POSGs which do not naturally correspond to a sequence of Bayesian games, and more similar to how SGs are defined in terms of successional normal-form games. We also impose a special Markov property on players’ types to insure that players need only reason about level-zero beliefs. This property is similar to the Markov property for states, in that it insures that all relevant historical information is included in the type, and is easy satisfy by explicitly including any past types that are still informative with the current type.

A MGIIs consists of a tuple $\langle N, A, S, P, R, O, s^{(0)}, o^{(0)} \rangle$

- $N$ is the set of players, $n = |N|$.
- $A = \Pi_{i=1}^n A_i$ is the set of actions.
- $O = \Pi_{i=1}^n O_i$ is the set of private signals.
- $S$ is the set of states.
- $P : S \times O \times A \rightarrow \Delta(S \times O)$ is the probability transition function with $P(s', \sigma|s, \sigma', \delta)$ being the probability of ending up in state $s'$ with signals $\sigma'$ after joint-action $\delta$ in state $s$ with signals $\sigma$.

$P$ must have the following Markov property:

$\forall s, s', \sigma, \sigma, \delta, \delta', j, i \in O_j :$

$$Pr[\sigma_{-i}^{j-1}|s', s, \sigma', \delta', \delta_{-j,i}] = Pr[\sigma_{-i}^{j-1}|s', s, \sigma', \delta_{-j,i}]$$

(a player’s signal is independent of other players’ previous signals given what is common knowledge and other players’ new signal)

- $R : S \times O \times A \rightarrow \mathbb{R}^n$ is the reward function.
- $s^{(0)} \in S$ is the initial state with known types $o^{(0)} \in O$.

In a MGIIs a player’s observation is both their private signal and the current state, but not other players’ signals. Unlike POSGs, MGIIs are explicitly defined in terms of successive Bayesian games where the signal is each player’s type. We don’t call signals “types” because that would denote capturing all private information. In fact, previous signals are still informative, but as we will show any information provided is common knowledge. Also notice that unlike POSGs the union of all player’s observations (if they shared their observations with each other) would reveal the true state. However, despite these differences the important distinction between MGIIs and POSGs lies in the Markov property enforced on signals in $P$. To demonstrate this point we compare MGIIs without the Markov property to POSGs and show that they are equivalent models.

**Lemma 0.0.1.** MGIIs without the Markov property on signals have equivalent representational power as POSGs.

**Proof.** While MGIIs are more complicated to define than POSGs, we show that MGIIs without the Markov property and POSGs can be converted into each other using polynomially similar space. It’s easy to convert a MGIIs into a POSG. The unobserved true states are the union of $s$ and $\sigma$ while the observations to each player $i$ are $s, \sigma_i$. The remaining variables are trivially similar.

Turning a POSG into an MGIIs (without the Markov restriction on $P$) is slightly harder. We define the MGIIs as having only a single state, and one additional player (who we will call nature). Nature has one signal for each true state of the POSG, but has only one action. Thus nature can give no information to the players and hides the true state. Each player gets observations (their type) and rewards as in the POSG but based on natures signal which acts as the unobserved underlying state of the POSG. The transition function $P$ falls out naturally but becomes quite large in the constructed MGIIs requiring $O(|S| \cdot |A| \cdot |O|^2)$ entries while the POSG only requires $O(|S| \cdot |A| \cdot |O|)$, (assuming the number of successor states is small). The POSG representation is more efficient because it assumes observations are conditionally independent of previous observations given the true-state, while the MGIIs representation of the POSG transitions based on all the players’ observations, not just natures’ observation. Players in this new MGIIs will receive the same observations and rewards as in the original POSG. \qed
private signals, \( \vec{o} \) and their private type. Thus a player’s belief will consist of a common knowledge representation. This fact is the key difference between POSGs and MGIs and allows us to convert MGIs into SGs of complete information – a model which has both an tractable exact solution (Murray and Gordon 2007) and a tractable solution with bounded error (Mac Dermed and Isbell 2009).

**Solving a MGII**

A solution to a MGII (or POSG) takes the form of a policy tree (a.k.a. strategy) for each agent. A policy tree is a function mapping every sequence of observations (including any implicit observation of one’s own action) to a distribution across actions. Note that policy trees need not and cannot be represented explicitly. In this paper we take a game theoretic perspective and assume that the players are completely rational. Assuming agents are rational dictates that a solution must be an equilibrium. While we have described what an equilibrium of a Bayesian stage-game looks like, an equilibrium for the full MGII does not directly follow. Given fixed policy trees for each player the world becomes a hidden Markov model and the utility for each player can be calculated (or at least sampled to arbitrary precision). We can therefore decide if it is beneficial (for any sequence of observations) for an agent to change one of their actions in the tree if all other actions stay the same. If no player can benefit by changing an action, then the policy-trees are in equilibrium.

A fundamental result from single agent RL tells us that it is sufficient to only worry about changing a single action, and not all actions at once. This amounts to insuring that when agents choose which action to execute in a stage-game they consider only their immediate reward but also the long term utility of the result of their action given the other player’s policy-trees. The expected joint-utility for each joint-action and joint-type is known as the continuation utility, \( cu_{\vec{o},\vec{a}} \). Because the agents consider both their immediate reward and their continuation utility they are in essence playing an augmented Bayesian stage game with payoffs \( R(s, \vec{o}, \vec{a}) + cu_{\vec{o},\vec{a}} \). If each of these augmented stage games are in equilibrium, and the continuation utility equals the true utility of following the policy trees than the policy trees are an equilibrium of the full MGII.

Figure 5 gives an example of an incomplete information game and the resulting set of agent normal form correlated equilibria.

**Example: Multi-Agent Tiger**

The single agent tiger problem (Cassandra, Kaelbling, and Littman 1994) has been used often to test and explain POMDP algorithms. In the problem there are two doors. One holds a tiger (-100 reward), the other some money (+10 reward) with even probability of which door holds which. A contestant must choose the correct door, but can wait and listen for growling noises (which are reported with 85% accuracy). Figure 6 depicts this game. There have been a number of multi-agent versions of this game (Doshi and Gmytrasiewicz 2009; Emery-Montemerlo et al. 2004; Zettlemeyer, Milch, and Kaelbling 2009). However, we choose to contract yet another version because previous problems mostly hinge on continous belief spaces which MGIs can’t model exactly.

In our multi-agent version, the game is like the single agent version except the two rooms have only a low wall
The game has two non-trivial states representing the two types of the player: Door 1 and Door 2. In each state, the player can choose to either stay or leap. The game has two non-trivial states representing the two types of the player: Door 1 and Door 2. In each state, the player can choose to either stay or leap. The player's decision is based on the common knowledge of the other player's action.

Figure 6: A POMDP representing the single agent tiger problem. Actions are in red, observations are in boxes. Grey numbers are transition probabilities or rewards.

Figure 7: A MGII representing a multi-agent tiger problem. There are two states, growl-1 and growl-2, which correspond to the two possible common knowledge observations. Both types represent identical Bayesian games. The outcome of joint-actions is given for each tiger type (door-1 or door-2).

The player's decision is based on the common knowledge of the other player's action. The player's decision is based on the common knowledge of the other player's action. The player's decision is based on the common knowledge of the other player's action. The player's decision is based on the common knowledge of the other player's action.

Converting a MGII into a SG

Given a MGII we define the following SG:

- \( N' = \bigcup_{i \in N} O_i \), one player for each player type. Recall that out solution concept is agent-normal-form which operates as if each agent type acts independently.
- \( A' = \Pi_{i=1}^n \Delta(\bar{A}) \), each player type submits a probability distribution across actions in the MGII as their action in the SG. While in fully observable domains, in games of incomplete information the distribution of actions not taken by players matters, as the relative ratio of actions across player types informs other players as to the actual type of the player.
- \( S' = S \times \Delta(\theta) \) states consist of the original state along with a common knowledge probability distribution, \( \tau \), across types \( \theta \).
- \( P' = S' \times A' \to \Delta(S') \):

\[
P'(s', \tau'|s, \tau, \bar{A}) = \sum_{\bar{a}} Pr[\bar{a}|\bar{A}] \sum_{\theta} \tau(\theta) \sum_{\theta'} P'(s', \theta'|s, \tau, \bar{a}) Pr[\tau'|s, \theta', \tau, \bar{a}, \bar{A}]
\]

where: \( Pr[\bar{a}|\bar{A}] \) is the probability of joint-action \( \bar{a} \) given actions distributions \( \bar{A}, \bar{a} \), \( \bar{a} \subseteq \bar{a} \) is the vector of actions
taken just by agent-types $\theta$, and $\tau(\theta)$ is the current probability mass assigned to joint-types $\theta$.

$\tau$ transitions deterministically given the public observations $s, s', \tau, a, A$. $A$ is public because agents are rational, and thus able to predict the behavior of other agents.

$$P_r[\tau'|s, s', \tau, a, A] = \sum_{\theta_1} P_r[\theta|A] P_r[\theta']$$

$$= \sum_{\theta_1} \sum_{\theta_2} \sum_{\theta_3} P_r[\theta_1|A] P_r[\theta_2|\theta_1] P_r[\theta'|\theta_2]$$

$$= \sum_{\theta_1} \sum_{\theta_2} \sum_{\theta_3} P_r[\theta_1|A] P_r[\theta_2|\theta_1, a_2] P_r[\theta'|\theta_2, \theta_1, a_2]$$

$$= \sum_{\theta_1} \sum_{\theta_2} \sum_{\theta_3} P_r[\theta_1|A] P_r[\theta_2|\theta_1, a_2] P(s', \theta'|s, \theta_1, a_2)$$

$P_r[\tau'|s, s', \tau, a, A] = 0$ Otherwise.

- $R' = R'(s, \tau, A)_{\theta_i \in N'} = \sum_{\theta_1 \in \Delta(\theta_1), \theta_1 = 0} \tau(\theta_1) \sum_{\theta_2} P_r[\theta_2|A] R(s, \theta, a_2)$

- $s(0) = \{s(0), \tau(0)\}$ where $\tau(0)(\theta(0)) = 1$ and 0 elsewhere.

We now argue that an equilibrium of this SG (let's call it $E$) maps to an equilibrium of the original MGII. $E$ takes the form of a joint-policy mapping states to joint-action distributions. Every observation trace of the original MGII corresponds to a state in the SG, and thus maps to a joint-action distribution of $E$. In particular the transition function $P'$ of the SG insures that the common-knowledge probability distribution across types $\tau \in \Delta(\theta)$ of the successor states matches equation 3 when $\tau^{(0)}(\theta^{(0)})$ is given by the current state’s common knowledge distribution. In other words, for a MGII the sequence of Bayesian games that a POSG follows as laid out by Emery-Montmerleto et.al 2004 corresponds to the SG’s states. The reward in the SG for each player type is defined to be the expected reward given the probability distribution across other player types. This is the same expected immediate reward in equations 1 and 1 in the corresponding Bayesian game. Because the sequence of Bayesian games is the same for both the MGII and the SG, the continuation utility of each augmented stage-game in the SG for each player-type will be the same as the continuation utilities in the MGII. This mapping allows us to compare the equilibrium constraints of the SG with the MGII Bayesian stage games. The reward $R'$ given in the SG is equal to the expected reward in both sides of equation 1, thus the fully observable rationality constraints $\sum_{a_1 = \omega_1} R'(a) Q_{a_1} \geq \sum_{a_1 = \omega_1} R'(a) Q_{a_1}$ equals the Bayesian rationality constraints (equation 1). Finally, the SG is designed so that all player-types are playing independently and thus can not collaborate with each other (as per an agent-normal form equilibrium). This also insures that the true type of players can not be disclosed, since no true type exists in the SG. This satisfies the non-disclosure constraints (equation 2). Therefore each state corresponds to an augmented stage-game of the original MGII and an equilibrium of the SG $E$ satisfies the equilibrium inequalities of all such Bayesian-games and corresponds to an agent-normal form correlated equilibrium of the original MGII.

**Conclusion**

We presented a new model for multi-agent sequential decision problems with hidden state: Markov games of incomplete information (MGII). MGII naturally represent sequences of Bayesian games and forces a Markov property on private information such that the only useful information which is not common knowledge is a player’s most recent private signal (a player’s type). Furthermore, all common knowledge in MGII can be compactly represented as a probability distribution across types. This feature permits MGII’s to be converted into a continuous but bounded stochastic game, much as how POMDPs can be converted into continuous belief-space MDPs. This continuous SG can be approximated by discretizing the space. While such an approximate has no error guarantees, we believe future research will be able to merge algorithms that solve POMDPs with those that solve SGs to create tractable and accurate algorithms for MGII.

**References**


