My thesis work provides a new framework for planning in multiagent, stochastic, partially observable domains with little knowledge about other agents. The relevance of the contribution lies in the variety of practical applications this approach can help tackling, given the very generic assumptions about the environment and the other agents. In order to cope with this level of generality, Bayesian nonparametric techniques are employed for learning the models of other agents in the form of finite state controllers. Two scenarios are considered: one in which the agent learns the models of the other agents, using a sequence of observations, and subsequently uses this knowledge to plan optimally, and one in which learning and exploitation are interleaved in an online algorithm.

1. INTRODUCTION

An autonomous, rational agent operating in a stochastic, partially observable environment maximizes its expected utility, usually the discounted sum of future rewards, as in the case of partially observable Markov decision processes (POMDPs) [Russell and Norvig 2009; Kaelbling et al. 1998]. At least two sources of uncertainty must be dealt with: the world state's dynamics and the relation between the observations and the state of the environment. Another layer of uncertainty in multiagent environments is due to the actions of other agents which also affect the state of the world, and our agent's payoff. In order to achieve good performance it is useful to predict the actions of other agents. In my thesis work, I consider an agent that maintains explicit models of other agents in the form of finite state controllers (FSCs).

It is assumed that the agent does not know, a priori, the model of the other agents, nor their preferences (reward functions.) Therefore, it is necessary for the agent to learn these models from observed behavior, in order to be able to accurately predict other agents' actions, and ultimately maximize its expected utility. Since the complexity of the other agent's FSCs is unknown, and a priori unbounded, I consider a flexible Bayesian nonparametric [Hjort et al. 2010] method for learning a posterior distribution over the space of all possible FSCs. Such method allows the size of the learned representation to scale gracefully with the complexity of the observed data.

In my thesis, two possible scenarios are investigated. In the first, the protagonist agent observes a sequence of interactions of another agent with the environment, learns models of this behavior, and subsequently exploits this knowledge in a multiagent planning algorithm. In the second, the learning and exploitation phases are interleaved in an online planning algorithm.

My work is related to research on plan recognition [Charniak and Goldman 1993], and on goal-based POMDPs [Ramirez and Geffner 2011]. In game theory, deterministic finite automata (DFA) have been employed to represent strategies in the presence of bounded rationality [Rubinstein 1998] when the opponent's actions are assumed observable. [Carmel and Markovitch 1996] provides a heuristic for the on-line inference of a consistent DFA, which does not guarantee any bound on the complexity of the learned model. However, we do not use the classical game-theoretic solution concept of a Nash equilibrium, building on the growing body of work that uses the

\[1\] Here and in the rest of this paper, I consider one single agent interacting with the protagonist agent; however, the N-agents generalization is straightforward.
decision-theoretic solution concept [Oliehoek and Amato 2014] and behavioral game theory [Wright and Leyton-Brown 2012].

2. OFFLINE METHOD: LEARN, THEN EXPLOIT

In this scenario, the protagonist agent, denoted as $i$, first observes a sequence of interactions of the other agent ($j$) with the environment —while itself either executing a predefined policy or staying completely out of the way— and uses this information to learn a set of possible models of the other agent. Specifically, the agent learns over a class of FSCs named probabilistic deterministic finite state controllers (PDFCs), in which the transition between internal states of the modeled agent is deterministic, given an observation, while the action generated in each state is stochastic. Afterwards, the agent uses a planning algorithm to compute its new optimal policy in light of what has been learned. The two phases are briefly described in the rest of this section.

2.1.Offline Learning

Let us denote the sequence of observed interactions of agent $j$ with the world as $h_{1:T}$; the learning task is to infer the posterior distribution over all possible PDFCs $c \in C$, i.e.

$$p(c|h_{1:T}) \propto p(h_{1:T}|c) \ p(c) \quad (1)$$

where $p(c)$ is the prior distribution over $C$, and $p(h_{1:T}|c)$ is the likelihood awarded to the observed sequence assuming $j$’s model is $c$. There are different possible assumptions that can be made about what the observed sequence consists of. In what is arguably the simplest, albeit quite unrealistic case, $h_{1:T} = (\omega_j^{1:T}, a_j^{1:T})$, that is, the sequence of observations received by $j$ and its actions are perfectly observed by $i$. A relaxation of this assumption is when $j$’s actions and the state of the world are observable, i.e. $h_{1:T} = (s_1^{1:T}, a_1^{1:T})$. A more realistic and general case is when the learning agent $i$ does not observe directly the state and the actions, but instead receives information through its own observation function, i.e. $h_{1:T} = \omega_i^{1:T}$. Moreover, agent $i$ might be uncertain about $j$’s observation model. All these cases, and the learning problems they entail, are explored in depth in [Panella and Gmytrasiewicz 2014].

As mentioned above, Bayesian nonparametric methods are employed, that allow flexibility over the number of states in the learned controller. The Dirichlet Process is used to define a distribution over the infinite space of PDFCs. Intuitively, the concentration parameter of the process indirectly controls the number of states and how the transitions between them occur, while the base measure acts as a prior for the action-generating distributions of each state. Since the considered problem is too complex to be amenable to conjugate analysis, a Gibbs sampler is developed that provides as output a set of samples from the posterior distribution over PDFCs. The key component of the sampling schema is the use of the Chinese Restaurant Process conditional distribution, that is the sampling counterpart of the Dirichlet Process and is used when determining the values of the transition function, given all the other variables involved. According to what assumption about the observed sequence is made, a number of other variables need to be sampled in each iteration of the Gibbs sampler. Perhaps unsurprisingly, some of the formulas used to sample the unobserved values in the considered time series are ad-hoc variations of distributions used for standard Hidden Markov Models, with some additional complexity. For more details about the learning phase, see [Panella and Gmytrasiewicz 2014].

2.2. Offline Planning

Once a set of models of agent $j$ has been learned, it can be used to perform optimal planning in a multiagent POMDP algorithm. Specifically, I consider a version of the
Learning and Planning in Multiagent POMDPs Using Finite-State Models of Other Agents

The idea is to expand the set of states of the environment $S$ by including the set $M$ of models of the other agent that have been learned. Since a version of the value function and the belief update equation to cope with this augmented space can be derived, and that the piecewise linearity and convexity of the value function carries over from the single-agent POMDP case, it is possible to adapt existing POMDP algorithms to solve this type of I-POMDPs. One problem is that the resulting state space can easily be too large for standard methods to work in reasonable time. However, the augmented state space exhibits a great deal of conditional independence and context-specific independence. This allows algorithms that exploits state factorization, such as symbolic Perseus, to successfully be employed in solving our problem.

More details on the planning algorithm are available in [paper currently under review].

3. ONLINE METHOD: LEARN WHILE EXPLOITING

In some cases, agent $i$ might not wish or be allowed to passively observe the behavior of agent $j$ before exploiting such information in a second stage. Perhaps the most important and fascinating case that makes it impossible to divide the process in two phases is when agent $j$ is in turn learning a model of $i$. When that happens, the interaction of the two agent becomes part of the very learning process, as it happens in multiagent learning tasks. From the point of view of agent $i$, the problem is in fact akin to model-based reinforcement learning; however, the agent does not learn a model of the world’s dynamics (here assumed known,) but instead the explicit model of another agent.

The methodology I propose to tackle this problem is an online Monte-Carlo tree search algorithm, extending the approach described in [Silver and Veness 2010]. As in the previous case, the set of states of the world is extended with a set of possible models of the other agent. Similarly to particle filtering, the extended state space is sampled and the particles are repeatedly projected forward to approximate the sequence of future states for a given horizon, and choose the optimal action at each step. I augment this process by interleaving a learning operation that performs a given number of iterations of the Gibbs sampling in order to update the “particles” representing the other agent, in light of the newly collected evidence. This operation can be performed at each time step, or whenever the likelihood awarded by these samples to the observed sequence falls below a certain threshold. This method is currently being developed and implemented.

4. CONCLUSIONS AND FUTURE WORK

I have presented a methodology for learning and planning in multiagent, stochastic, partially observable environments. Only few existing approaches attempts at solving this kind of problem at this level of generality, making this work a substantial contribution in the context of multiagent systems.

There are many opportunities for future work, part of which will be included in my dissertation. One is to consider other types of models for the other agents, such as fully stochastic finite state controllers. Moreover, it will be interesting to explore theoretical properties of the emerging multiagent learning framework when both agents are simultaneously learning and acting. Another interesting development is to use the learning approach to infer properties of the other agent’s reward function, hence generating transferable knowledge that can be used in different contexts of interaction.
REFERENCES


