Scalable Bayesian Reinforcement Learning for Multiagent POMDPs

Christopher Amato
CSAIL
MIT
Cambridge, MA 02139
camato@csail.mit.edu

Frans A. Oliehoek
Department of Knowledge Engineering
Maastricht University
Maastricht, The Netherlands
frans.oliehoek@maastrichtuniversity.nl

Eric Shyu
CSAIL
MIT
Cambridge, MA 02139
eshyu@mit.edu

Abstract

Bayesian methods for reinforcement learning (RL) allow model uncertainty to be considered explicitly and offer a principled way of dealing with the exploration/exploitation tradeoff. However, for multiagent systems there have been few such approaches, and none of them apply to problems with state uncertainty. In this paper, we fill this gap by proposing a Bayesian RL framework for multiagent partially observable Markov decision processes that is able to take advantage of structure present in many problems. In this framework, a team of agents operates in a centralized fashion, but has uncertainty about the model of the environment. Fitting many real-world situations, we consider the case where agents learn the appropriate models while acting in an online fashion. Because it can quickly become intractable to choose the optimal action in naïve versions of this online learning problem, we propose a more scalable approach based on sample-based search and factored value functions for the set of agents. Experimental results show that we are able to provide high quality solutions to large problems even with a large amount of initial model uncertainty.

Keywords: Multiagent Learning, Bayesian Reinforcement Learning, POMDPs

Acknowledgements

Research supported in part by AFOSR MURI project #FA9550-091-0538.

1 Introduction

Bayesian reinforcement learning (RL) techniques are promising in that, in principle, they provide an optimal exploration/exploitation trade-off with respect to the prior belief. In the context of multiagent systems, Bayesian RL has been used in stochastic games [2] and factored Markov decision processes (MDPs) [18]. These approaches assume the state of the problem is fully observable (or can be decomposed into fully observable components). Unfortunately, no approaches have been proposed that can model and solve problems with partial observability. In fact, while planning in partially observable multiagent domains has had some success (e.g., [1, 11]), very few multiagent RL approaches of any kind consider partially observable domains (notable exceptions, e.g., [3, 13]).

We propose a framework for Bayesian learning in multiagent systems with state uncertainty using multiagent partially observable Markov decision processes (MPOMDPs) that can exploit the multiagent structure in these problems. MPOMDPs represent a centralized perspective where all agents share the same partially observable view of the world and can coordinate on their actions, but have uncertainty about the underlying environment. To model this problem, we extend the Bayes-Adaptive POMDP (BA-POMDP) [15]—which represents beliefs over possible model parameters using Dirichlet distributions—to the multiagent setting. The resulting framework can be used as a Bayesian online learning approach which represents the initial model using priors and updates probability distributions over possible models as the agent acts in the real world. In particular, we utilize sample-based planning based on Monte Carlo tree search (MCTS) which has shown promise performing planning in large POMDPs [16] and Bayesian learning in large MDPs [6].

Unfortunately, these methods become ineffective as the number of (joint) actions and observations scales exponentially in the number of agents. To combat this intractability, we propose exploiting structure in the value functions associated with the agents. That is, many multiagent problems possess structure in the form of locality of interaction: agents interact directly with a subset of other agents. This structure enables a decomposition of the value function into a set of overlapping factors, which can be used to produce high quality solutions [5, 9, 10]. We propose two techniques for incorporating such factored value functions into MCTS, thereby mitigating the additional challenges for scalability imposed by the exponential number of joint actions and observations. This approach is the first MCTS variant to exploit structure in multiagent systems, achieving better sample complexity and improving value function generalization by factorization.

2 Background

MPOMDPs form a framework for multiagent planning under uncertainty for a team of agents. At every stage, agents take individual actions and receive individual observations. However, in an MPOMDP, the assumption is that the team of agents is acting in a 'centralized manner', which means that we assume that all individual observations are shared via communication. We will restrict ourselves to the setting where such communication is free of noise, costs and delays.

Formally, an MPOMDP is a tuple $\langle I, S, \{A_i\}, T, R, \{Z_i\}, O, h\rangle$ with: I, a finite set of agents; S, a finite set of states with designated initial state distribution b_0; $A = \times_i A_i$, the set of joint actions, using action sets for each agent, i; T, a set of state transition probabilities: $T^{s\vec{a}s'} = \Pr(s'|s,\vec{a})$, the probability of transitioning from state s to s' when the set of actions $\vec{a}$ are taken by the agents; R, a reward function: $R(s,\vec{a})$, the immediate reward for being in state s and taking the set of actions $\vec{a}$; $Z = \times_i Z_i$, the set of joint observations, using observation sets for each agent, i; O, a set of observation probabilities: $O^{\vec{a}s'\vec{z}} = \Pr(\vec{z}|\vec{a},s')$, the probability of seeing the set of observations $\vec{o}$ given the set of actions $\vec{a}$ was taken which results in state s'; h, the number of steps before termination or horizon. An MPOMDP can be reduced to a special type of POMDP in which there is a single centralized controller that takes joint actions and receives joint observations [14].

Most research concerning POMDPs has considered the task of *planning*: given a full specification of the model, determine an optimal (joint) policy, π, mapping past (joint) observation histories (which can be summarized by distributions $b(s)$ over states called beliefs) to (joint) actions. Such an optimal (joint) policy can be extracted from an optimal Q-value function, $Q(b,a) = \sum_s R(s,a) + \sum_z P(z|b,a) \max_{a'} Q(b',a')$, by acting greedily, in a way similar to the situations in regular MDPs [17]. Computing $Q(b,a)$, however, is complicated by the fact that the space of beliefs is continuous [7].

While POMDP planning methods can find solutions effectively given a problem model, for many real-world applications, the model is not (perfectly) known in advance, requiring the agents to learn about their environment during execution. To deal with such partially observable multiagent learning problems, we build on the framework of Bayes-Adaptive POMDPs [15]. This approach utilizes Dirichlet distributions to model uncertainty over both transitions and observations.

Intuitively, if the agent could observe both states and observations, it could maintain vectors ϕ and ψ of counts for transitions and observations respectively. That is, $\phi^a_{ss'}$ is the transition count representing the number times state s' resulted from taking action a in state s and $\psi^a_{s'z}$ is the observation count representing the number of times observation z was seen after taking action a and transitioning to state s'. While the agent cannot observe the states and has uncertainty about the actual count vectors, *this uncertainty can be represented using the regular POMDP formalism*. That is, the count vectors are included as part of the hidden state of a special POMDP, called BA-POMDP.

3 BA-MPOMDPs

The BA-POMDP can be extended to the multiagent setting in a straightforward manner as the Bayes-Adaptive multiagent POMDP (BA-MPOMDP). The BA-MPOMDP model allows a team of agents to learn about its environment while acting in a Bayesian fashion and is applicable in any multiagent RL setting where there is instantaneous communication. Since a BA-MPOMDP can be simply seen as a BA-POMDP where the actions are joint actions and the observations are joint observations, the theoretical results related to BA-POMDPs also apply to the BA-MPOMDP model.

Formally, a BA-MPOMDP is a tuple $\langle I, S_{BM}, \{A_i\}, T_{BM}, R_{BM}, \{Z_i\}, O_{BM}, h\rangle$ where $I, \{A_i\}, \{Z_i\}, h$ are as before. The state of the BA-MPOMDP now includes the Dirichlet parameters (i.e., the count vectors): $s_{BM} = \langle s, \phi, \psi\rangle$. As such, the set of states is given by $S_{BM} = S \times \mathcal{T} \times \mathcal{O}$ where $\mathcal{T} = \{\phi \in \mathbb{N}^{|S||A||S|} | \forall (s,\vec{a}) \sum_{s'} \phi^{\vec{a}}_{ss'} > 0\}$ is the space of all possible transition counts and similarly $\mathcal{O}$ is the space of all possible observation parameters: $\mathcal{O} = \{\psi \in \mathbb{N}^{|S||A||Z|} | \forall (s,\vec{a}) \sum_{\vec{z}} \psi^{\vec{a}}_{s'\vec{z}} > 0\}$ where $|A|$ is the number of joint actions and $|Z|$ is the number of joint observations.

In order to define T_{BM}, O_{BM}, the transition and observation probabilities for the BA-MPOMDP, we need the expected transition and observation probabilities induced by (the count vectors of) a state: $T^{s\vec{a}s'}_{\phi} = \mathbf{E}[T^{s\vec{a}s'}|\phi] = \phi^{\vec{a}}_{ss'}/N^{s\vec{a}}_{\phi}$, $O^{\vec{a}s'\vec{z}}_{\psi} = \mathbf{E}[O^{\vec{a}s'\vec{z}}|\psi] = \psi^{\vec{a}}_{s''\vec{z}}/N^{\vec{a}s'}_{\psi}$, where $N^{sa}_{\phi} = \sum_{s''} \phi^{a}_{ss''}$, and $N^{as'}_{\psi} = \sum_{z'} \psi^{a}_{s'z'}$. The transition probabilities $P(\langle s', \phi', \psi'\rangle | \langle s, \phi, \psi\rangle, a)$ can be defined using a vector $\delta^{a}_{ss'}$ which is 1 at the index of a, s and s' and 0 otherwise: $T_{BM}((s,\phi,\psi),\vec{a},(s',\phi',\psi')) = T^{s\vec{a}s'}_{\phi} O^{\vec{a}s'z}_{\psi}$ if $\phi' = \phi + \delta^{\vec{a}}_{ss'}$ and $\psi' = \psi + \delta^{\vec{a}}_{s'z}$(and 0 otherwise). Similarly, for observations, we define $\delta^{\vec{a}}_{s'\vec{z}}$ to be a vector with value 1 at the index $\vec{a}$, s' and $\vec{z}$ and 0 otherwise: $O_{BM}((s,\phi,\psi),\vec{a},(s',\phi',\psi'),z) = 1$ if $\phi' = \phi + \delta^{\vec{a}}_{ss'}$ and $\psi' = \psi + \delta^{\vec{a}}_{s'z}$(and 0 otherwise). The reward model remains the same (since it is assumed to be known), $R_{BM}((s,\phi,\psi),\vec{a}) = R(s,\vec{a})$. We assume the initial state distribution b_0 and initial count vectors ϕ_0 and ψ_0 are given.

4 Monte Carlo Tree Search for Multiagent POMDPs

Monte Carlo Tree Search for POMDPs A successful recent online planning method, called partially observable Monte Carlo planning (POMCP) [16], extends Monte Carlo tree search (MCTS), and in particular the UCT algorithm [8], to solving POMDPs. At every stage, the algorithm performs online planning, given the current belief, by incrementally building a lookahead tree that contains (statistics that represent the) $Q(b,a)$. The algorithm, however, avoids expensive belief updates by creating nodes not for each belief, but simply for each action-observation history h. In particular, it samples hidden states s only at the root node (called 'root sampling') and uses that state to sample a trajectory that first traverses the lookahead tree and then performs a (random) rollout. The return of this trajectory is used to update the statistics for all visited nodes. When traversing the tree, actions are selected to maximize the 'upper confidence bounds': $U(h,a) = Q(h,a) + c\sqrt{\log(N+1)/n}$. Here, N is the number of times the history has been reached and n is the number of times that action a has been taken in that history. When the exploration constant c is set correctly, POMCP can be shown to converge in the limit. Moreover, the method has demonstrated good performance in large domains with a limited numbers of simulations.

Because the BA-MPOMDP formalism constructs an (infinite state[1]) POMDP, POMCP could be applied here too. Doing so means that during online planning, a lookahead tree will be constructed that has nodes corresponding to *joint* action observation histories $\vec{h}$, and where statistics are stored that represent the expected values $Q(\vec{h},\vec{a})$ and upper confidence bounds $U(\vec{h},\vec{a})$. A shortcoming of Monte Carlo tree search methods is that they are not directly suitable for multiagent problems due to the large number joint actions and joint observations, which are exponential in the number of agents.

The large number of joint observations is problematic, since it will lead to a lookahead tree with very high branching factor and a breakdown of particle filtering to estimate the belief (necessitating starting from the initial belief again, or acting using a separate policy such as a random one). The large number of actions results in exponentially many joint actions that have to be selected at least a few times to drive down their confidence bounds (i.e., exploration bonus).

Coordination Graphs In many cases, the effect of a joint action is factorizable as the effects of the action of individual agents or small groups of agents. For instance, consider a team of agents that is tasked with fighting fire at a number of burning houses, as illustrated in Fig. 1(a). In such a setting, the overall transition probabilities can be factored as a product of transition probabilities for each house [12], and the transitions of a particular house may depend only on the amount of water deposited on that house (rather than the exact joint action).

We can consider agents' interactions in the form of a coordination graph which represents interactions between subsets of agents and permits factored linear value functions as an approximation to the joint value function. Specifically—for the moment assuming a stateless problem—an action-value function can be approximated by $Q(\vec{a}) = \sum_e Q_e(\vec{a}_e)$, where each component e is a value specified over only a (possibly overlapping) subset of agents. Note that in this and later

[1]Since there can be infinitely many count vectors, the state space is infinite, but a finite approximate model can be used [15].

(a) (b)

Figure 1: (a) Illustration of an MPOMDP in which a team of agents has to fight a fire, and (b) Illustration of the coordination graph (as a factor graph) with houses represented as h1,. . . ,h4 and agents represented as a1,. . . ,a3. Each agent should coordinate with the adjacent agents in the graph.

formulations, a normalization term of $1/|e|$ can be used to scale the Q-values back to the range of the original problem, but the maximizing actions remain the same.

In cases where such a factorization holds, the maximization $\max_{\vec{a}} \sum_e Q_e(\vec{a}_e)$ can be performed efficiently via variable elimination (VE) [5], or max-sum [4, 9]. These algorithms are not exponential in the number of agents (although VE is exponential in the induced with), and therefore enable significant speed-ups for larger number of agents.

Factored Statistics The first technique we introduce, called *Factored Statistics* directly applies the idea of coordination graphs inside MCTS. Rather than maintaining one set of statistics in each node that expresses the expected value for each joint action $Q(\vec{h}, \vec{a})$, we maintain several sets of statistics, each one expressing the value for a set of agents $Q_e(\vec{h}, \vec{a}_e)$. As such, the Q-value function is approximated by $Q(\vec{h}, \vec{a}) \approx \sum_e Q_e(\vec{h}, \vec{a}_e)$.

Since this method retains fewer statistics and performs joint action selection more efficiently via VE, we expect that this approach will be more efficient than plain application of POMCP to the BA-MPOMDP. However, the complexity due to joint observations is not directly addressed: because joint histories are used, reuse of nodes and the ability to create nodes in the tree for the necessary observations seen during execution may be limited.

Factored Trees The second technique, called *Factored Trees*, additionally tries to overcome this burden of the large number of joint observations. This is done by further decomposing the joint histories into local histories over factors. That is, in this case, the Q-values are approximated by $Q(\vec{h}, \vec{a}) \approx \sum_e Q_e(\vec{h}_e, \vec{a}_e)$. This approach further reduces the number of statistics maintained and increases the reuse of nodes in MCTS and the chance that nodes in the trees will exist for observations that are seen during execution. As such, it aims to increase performance by utilizing more generalization (now also over local histories), as well as producing more robust particle filters.

Finally, we note that this type of factorization has major implications for the implementation of the approach: rather than constructing a single tree, we now need to construct a number of trees in parallel, one for each factor (or edge in the coordination graph) e. A node of the tree of an component e now stores the required statistics: $N_{\vec{h}_e}$, the count for the local history, $n_{\vec{a}_e}$, the counts for actions taken in the local tree and Q_e for the tree.

5 Experimental Results

We performed an evaluation on the firefighting problem from Section 4. Each experiment was run for a given number of simulations (the number of samples used at each step to choose an action) and averaged over a number of runs (resetting the state and count vectors to their initial values). To determine the value of acting with the true model known, we provide results from POMCP [16] and to show the result of acting solely based on the initial prior given by the initial count vectors, we provide results from a 'No learning' method. This no learning approach uses the BA-MPOMDP in the same way as the other methods, but never updates the count vectors, causing it to retain the same uncertain distribution over models. We also provide results for the value produced by uniform random action selection. The values given are the average undiscounted returns for the horizon (i.e., number of steps in the problem) shown. Experiments were run on a single core of a 2.5 GHz machine with 8GB of memory.

The fire fighting domain [12] consists of four agents and five houses, each with 3 different fire levels. Fires are suppressed more quickly if a larger number of agents choose that particular house. Fires also spread to neighbor's houses and can start at any house with a small probability. Priors were used that had high confidence in (near) correct transition probabilities and low confidence in incorrect (near uniform) observation probabilities.

Results are shown in Table 1 where the benefits of the factored approaches are seen. For a small number of samples (which is crucial on large problems, $|S| = 243$, $|A| = 81$, $|Z| = 16$) the factored tree method learns very quickly, providing significantly better values. Using factored statistics will learn more slowly, but the value function is closer to optimal due to the use of the full history. BA-MPOMDP and No learning perform poorly due to the incorrect prior and insufficient

	Horizon 10	
	50 Simulations	250 Simulations
POMCP (true)	-87.3 ± 2.09	-50.6 ± 9.50
Factored statistics	-47.1 ± 7.85	-21.8 ± 1.94
Factored tree	-41.2 ± 3.85	-29.8 ± 5.64
BA-MPOMDP	-85.6 ± 1.66	-51.6 ± 5.94
No learning	-86.7 ± 1.59	-54.6 ± 6.41
Random	-81.6 ± 4.09	-81.6 ± 4.09

	Horizon 50
	50 Simulations
POMCP (true)	-425.7 ± 4.65
Factored statistics	-403.9 ± 10.78
Factored tree	-210.2 ± 12.01
BA-MPOMDP	-436.9 ± 4.66
No learning	-436.9 ± 4.66
Random	-437.6 ± 5.73

Table 1: Undiscounted return (and standard error) for horizon 10 and 50 fire fighting problems averaged over 10 runs

samples to choose high-quality actions. After more samples (as seen by 250 samples), the performance of the flat models improve, but the factored methods still perform better. POMCP(true) with 100000 simulations was able to achieve values for horizon 10 of -19.83 ± 0.96 and 50 of -62.1 ± 6.96. Note that the 'No learning' method gives the value of using the prior without learning, showing the benefit of the learning approaches. In particular, the factored approaches are able to improve learning through generalization and make better use of the statistics and the particle filter.

6 Conclusions

We present the first method to utilize multiagent structure to produce a scalable method for multiagent Bayesian reinforcement learning with state uncertainty. To combat exponential growth of the number of joint actions and observations, we propose two methods for decomposing the agents using a coordination graph to reduce 1) the number of joint actions and 2) the number of joint histories considered. These methods are used in conjunction with a leading POMDP method, POMCP [16], to generate a MCTS-based sample-based planner for our Bayes-Adaptive MPOMDP model. Our experimental results demonstrate that the proposed techniques allow agents to both learn faster (with fewer simulations) and produce higher quality solutions. We expect that these approaches can serve as the basis for many future work directions in multiagent learning as well as be used to solve BA-POMDPs with large action and observation spaces.

References

[1] C. Amato, G. Chowdhary, A. Geramifard, N. K. Ure, and M. J. Kochenderfer. Decentralized control of partially observable Markov decision processes. In *CDC*, 2013.

[2] G. Chalkiadakis and C. Boutilier. Coordination in multiagent reinforcement learning: A Bayesian approach. In *AAMAS*, 2003.

[3] Y.-H. Chang, T. Ho, and L. P. Kaelbling. All learning is local: Multi-agent learning in global reward games. In *NIPS 16*, 2004.

[4] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In *AAMAS*, 2008.

[5] C. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored MDPs. In *NIPS*, 15, 2001.

[6] A. Guez, D. Silver, and P. Dayan. Efficient Bayes-adaptive reinforcement learning using sample-based search. In *NIPS 12*, 2012.

[7] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *AIJ*, 101, 1998.

[8] L. Kocsis and C. Szepesvári. Bandit based monte-carlo planning. In *ECML*, 2006.

[9] J. R. Kok and N. Vlassis. Collaborative multiagent reinforcement learning by payoff propagation. *JMLR*, 7, 2006.

[10] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked distributed POMDPs: a synthesis of distributed constraint optimization and POMDPs. In *AAAI*, 2005.

[11] F. A. Oliehoek. Decentralized POMDPs. In M. Wiering and M. van Otterlo, editors, *Reinforcement Learning: State of the Art*. Springer, 2012.

[12] F. A. Oliehoek, M. T. J. Spaan, S. Whiteson, and N. Vlassis. Exploiting locality of interaction in factored Dec-POMDPs. In *AAMAS*, 2008.

[13] L. Peshkin, K.-E. Kim, N. Meuleau, and L. P. Kaelbling. Learning to cooperate via policy search. In *UAI*, pages 489–496, 2000.

[14] D. V. Pynadath and M. Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *JAIR*, 16, 2002.

[15] S. Ross, J. Pineau, B. Chaib-draa, and P. Kreitmann. A Bayesian approach for learning and planning in partially observable Markov decision processes. *JAIR*, 12, 2011.

[16] D. Silver and J. Veness. Monte-carlo planning in large POMDPs. In *NIPS 23*, 2010.

[17] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[18] W. T. L. Teacy, G. Chalkiadakis, A. Farinelli, A. Rogers, N. R. Jennings, S. McClean, and G. Parr. Decentralized Bayesian reinforcement learning for online agent collaboration. In *AAMAS*, 2012.