Improved Planning for Infinite-Horizon Interactive POMDPs Using Probabilistic Inference

(Extended Abstract)

Xia Qu  
THINC Lab, Computer Science Dept.  
University of Georgia, Athens, GA  
quxia@uga.edu

Prashant Doshi  
THINC Lab, Computer Science Dept.  
University of Georgia, Athens, GA  
pdoshi@cs.uga.edu

ABSTRACT

We provide the first formalization of self-interested multiagent planning using expectation-maximization (EM). Our formalization in the context of infinite-horizon and finitely-nested interactive POMDPs (I-POMDP) is distinct from EM formulations for POMDPs and other multiagent planning frameworks. Specific to I-POMDPs, we exploit the graphical model structure and present a new approach based on block-coordinate descent for further speed up.

Categories and Subject Descriptors

1.2.11 [Distributed Artificial Intelligence]: Multiagent Systems

General Terms

Algorithms, Experimental

Keywords

expectation-maximization; multiagent systems; POMDP

1. PLANNING IN I-POMDP AS INERENCE

We may represent the policy of agent i for the infinite-horizon finitely-nested I-POMDP [1, 3] as a stochastic finite state controller (FSC), defined as: \( \pi_i = \{ N_i, T_i, C_i, V_i \} \) where \( N_i \) is the set of nodes in the controller. \( T_i : N_i \times A_i \times S_i \times N_i \rightarrow [0, 1] \) represents the node transition function: \( C_i : N_i \times A_i \times S_i \rightarrow [0, 1] \) denotes agent i’s action distribution at each node; and an initial distribution over the nodes is denoted by, \( V_i : N_i \rightarrow [0, 1] \). For convenience, we group \( V_i, T_i \) and \( C_i \) in \( f_i \). Define a controller at level \( l \) for agent i as \( \pi_{i,l} = \{ N_{i,l}, f_{i,l} \} \), where \( N_{i,l} \) is the set of nodes in the controller and \( f_{i,l} \) groups remaining parameters of the controller as mentioned before. Analogously to POMDPs [4], we formulate planning in multiagent settings formalized by I-POMDPs as likelihood maximization. The planning problem is modeled as a mixture of DBNs of increasing time from \( T=0 \) onwards. Transition and observation functions of I-POMDPs parameterize the chance nodes \( s \) and \( o \), respectively, \( Pr(r_{t-1}^i|a_{t-1}^i, a_j^t, s^t) \propto R_i(s_{t-1}^i, a_{t-1}^i, a_j^t) - R_{\max} - R_{\min} \). The networks include nodes, \( n_{i,l} \), of agent i’s level-l FSC. Therefore, functions in \( f_{i,l} \) parameterize the network as well, which are to be inferred. Additionally, the network includes the model nodes — one for each other agent — that contain the candidate level 0 models of the agent. Each model node provides the expected distribution over another agent’s actions. The straightforward approach is to infer a likely FSC for each level 0 model. However, this approach does not scale to many models. Proposition 1 shows that the dynamic \( Pr(a_j^t|s^t) \) is sufficient information for predictions.

PROPOSITION 1 (SUFFICIENCY). Distribution \( \prod_{t=0}^{T} Pr(a_j^t|s^t) \) across states, \( s^t \), is sufficient predictive information about other agent j to obtain the most likely policy of i.

Given Prop. 1, we seek to infer \( Pr(a_j^t|m_j^0) \) for each (updated) model of j at all steps, which is denoted as \( \phi_j^0 \). Other terms in the computation of \( Pr(a_j^t|s^t) \) are known parameters of the level 0 DBN. The likelihood maximization for the level 0 DBN is:

\[
\phi_j^0 = \arg\max_{\phi_j^0} \left( 1 - \gamma \right) \sum_{t=0}^{T} \sum_{s^t, a_j^t \in M_{j,0}^T} \gamma^t Pr(r_j^T = 1 | T, m_j, \phi_j^0)
\]

As the trajectory consisting of states, models, actions and observations of the other agent is hidden at planning time, we may solve the above likelihood maximization using EM.

E-step at level 0 The “data” in the level 0 DBN consists of the initial belief over the state and models, \( b_j^0 \), and the observed reward at \( T \). Analogously to EM for POMDPs, this motivates forward filtering-backward smoothing on a network with joint state, \((s^t, m_j^0)\), for computing the log likelihood. The transition function for the forward and backward steps is:

\[
Pr(s^t, m_j^0|s^{t-1}, m_j^0) = \sum_{a_j^t, s^{t-1}} Pr(s^{t-1}, m_j^0|s^{t-1}, m_j^0) T_{m_j^0}(s^{t-1}, a_j^t, m_j^0) K_{m_j^0}(s^{t-1}, a_j^t, m_j^0)
\]

As the trajectory consisting of states, models, actions and observations of the other agent is hidden at planning time, we may solve the above likelihood maximization using EM.

M-step at level 0 We obtain the updated \( \phi_j^0 \) from the full log likelihood by separating the terms and maximizing it w.r.t. \( \phi_j^0 \);
\[ \phi^{(O)}(a^{T}_{i}, s^{T}_{i}, n^{T}_{i+1}) = \sum_{a^{T-1}_{T-i}, a^{T-1}_{i}} \beta^{(O)}(s^{T-1}, s^{T-1}_{i}, a^{T-1}_{i}, a^{T-1}_{i}) \alpha^{(T-1)}(s^{T-1}_{i}, n^{T-1}_{i}) \]

\[ \beta^{(O)}(s^{T}, n^{T}_{i}) = \sum_{s^{T-1}, n^{T-1}_{i}} \alpha^{(T-1)}(s^{T-1}, n^{T-1}_{i}) \gamma \beta^{(O)}(s^{T-1}, n^{T-1}_{i}, a^{T-1}_{i}, a^{T-1}_{i}) \]

\[ \alpha^{(O)}(s^{T}, n^{T}_{i}) = \sum_{s^{T-1}, n^{T-1}_{i}} \beta^{(O)}(s^{T-1}, n^{T-1}_{i}, a^{T-1}_{i}, a^{T-1}_{i}) \alpha^{(T-1)}(s^{T-1}_{i}, n^{T-1}_{i}) \]

\[ \beta^{(O)}(s^{T}, n^{T}_{i}) = \sum_{s^{T-1}, n^{T-1}_{i}} \alpha^{(T-1)}(s^{T-1}, n^{T-1}_{i}) \gamma \beta^{(O)}(s^{T-1}, n^{T-1}_{i}, a^{T-1}_{i}, a^{T-1}_{i}) \]

2. EXPERIMENTS

Four variants of EM are evaluated as appropriate: the exact EM inference-based planning (labeled as I-EM); replacing the exact M-step with its greedy variant (I-EM-Greedy) [4]; iterating EM based on coordinate blocks (I-EM-BPI) and coupled with a greedy M-step (I-EM-BPI-Greedy). We use 2 problem domains: the noncooperative multiagent tiger problem [1] with a total of 5 agents and 50 models for each other agent. A larger noncooperative 2-agent money laundering (ML) problem [2] forms the second domain.

In Fig. 1(a, b), we compare the variants on both problems. Each method starts with a random seed, and the converged value is significantly better than a random FSC for all methods and problems. Increasing the sizes of FSCs gives better values in general but also increases time; using FSCs of sizes 5 and 3 for the 2 domains respectively demonstrated a good balance. I-EM-BPI consistently improves on I-EM: the corresponding value improvements by large steps initially (fast non-asymptotic rate of convergence). We compare the quickest of the I-EM variants with previous best algorithm, I-BPI [3] (Figs. 1(c, d)), allowing the latter to escape local optima as well by adding nodes. Observe that FSCs improved using I-EM-BPI converges to values similar to those of I-BPI almost two orders of magnitude faster. Beginning with 5 nodes, I-BPI adds more nodes to obtain the same level of value as EM for the tiger problem. For money laundering, I-EM-BPI-Greedy converges to controllers whose value is at least 1.5 times better.

REFERENCES