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 POMDP (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

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Algorithms, Experimental

Keywords

expectation-maximization; multiagent systems; POMDP

1. PLANNING IN I-POMDP AS INFERENCE

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: $\pi_i = \langle \mathcal{N}_i, \mathcal{T}_i, \mathcal{L}_i, \mathcal{V}_i \rangle$ where $\mathcal{N}_i$ is the set of nodes in the controller. $\mathcal{T}_i : \mathcal{N}_i \times A_i \times \Omega_i \times \mathcal{N}_i \rightarrow [0, 1]$ represents the node transition function; $\mathcal{L}_i : \mathcal{N}_i \times A_i \rightarrow [0, 1]$ denotes agent i's action distribution at each node; and an initial distribution over the nodes is denoted by, $\mathcal{V}_i : \mathcal{N}_i \rightarrow [0, 1]$. For convenience, we group $\mathcal{V}_i$, $\mathcal{T}_i$ and $\mathcal{L}_i$ in $\hat{f}_i$. Define a controller at level l for agent i as, $\pi_{i,l} = \langle \mathcal{N}_{i,l}, \hat{f}_{i,l} \rangle$, where $\mathcal{N}_{i,l}$ is the set of nodes in the controller and $\hat{f}_{i,l}$ 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-POMDP$_{i,l}$ parameterize the chance nodes s and o_i, respectively, $Pr(r_i^T | a_i^T, a_j^T, s^T) \propto \frac{R_i(s^T, a_i^T, a_j^T) - R_{max}}{R_{max} - R_{min}}$. The networks include nodes, $n_{i,l}$, of agent i's level-l FSC. Therefore, functions in $\hat{f}_{i,l}$ 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* $\prod_{t=0}^{T} \sum_{a_j^t \in A_j} Pr(a_j^t | s^t)$ *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 $Pr(a_j^t | m_{j,0}^t)$ for each (updated) model of j at all time 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:

$$\phi_{j,0}^* = \arg \max_{\phi_{j,0}} (1-\gamma) \sum_{T=0}^{\infty} \sum_{m_{j,0} \in M_{j,0}^T} \gamma^T Pr(r_j^T = 1 | T, m_{j,0}; \phi_{j,0})$$

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, $b_{i,1}^0$, 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,$m_{j,0}^t$), for computing the log likelihood. The transition function for the forward and backward steps is:

$$Pr(s^t, m_{j,0}^t | s^{t-1}, m_{j,0}^{t-1}) = \sum_{a_j^{t-1}, o_j^t} \phi_{j,0}(m_{j,0}^{t-1}, a_j^{t-1})$$
$$\times\, T_{m_j}(s^{t-1}, a_j^{t-1}, s^t)\, Pr(m_{j,0}^t | m_{j,0}^{t-1}, a_j^{t-1}, o_j^t) O_{m_j}(s^t, a_j^{t-1}, o_j^t)$$

where m_j in the subscripts is j's model at $t-1$. Forward filtering gives the probability of the next state as follows:

$$\alpha^t(s^t, m_{j,0}^t) = \sum_{s^{t-1}, m_{j,0}^{t-1}} Pr(s^t, m_{j,0}^t | s^{t-1}, m_{j,0}^{t-1})\, \alpha^{t-1}(s^{t-1}, m_{j,0}^{t-1})$$

where $\alpha^0(s^0, m_{j,0}^0)$ is the initial belief of agent i.

The smoothing by which we obtain the joint probability of the state and model at $t-1$ from the distribution at t is:

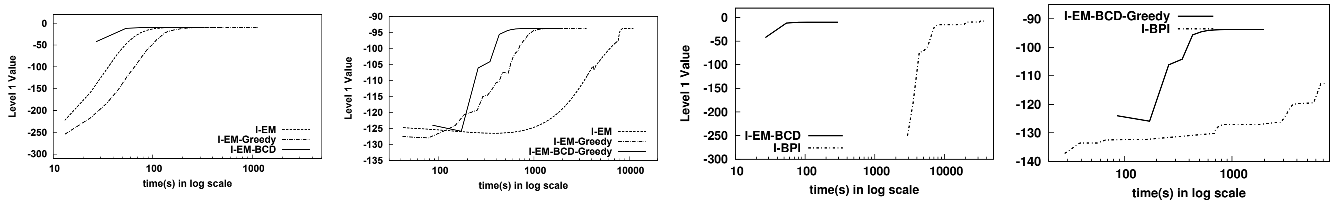
$$\beta^h(s^{t-1}, m_{j,0}^{t-1}) = \sum_{s^t, m_{j,0}^t} Pr(s^t, m_{j,0}^t | s^{t-1}, m_{j,0}^{t-1})\, \beta^{h-1}(s^t, m_{j,0}^t)$$

where h denotes the number of time steps until T (horizon), and $\beta^0(s^T, m_{j,0}^T) = E_{a_j^T | m_{j,0}^T}[Pr(r_j^T = 1 | s^T, m_{j,0}^T)]$.

Messages α^t and β^h give the probability of a state at some time slice in the DBN. As we consider a mixture of BNs, we seek the probabilities for all states in the mixture model. Subsequently, we may compute the forward and backward messages at all states for the entire mixture model in one sweep.

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. $\phi'_{j,0}$:

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*



(a) 5-agent tiger: all methods (b) 2-agent ML: all methods (c) 5-agent tiger: I-EM-BCD, I-BPI (c) 2-agent ML: I-EM-BCD-Greedy, I-BPI

Figure 1: FSCs improve with time for I-POMDP$_{i,1}$ in the (a) 5-agent tiger, and (b) 2-agent ML. I-EM-BCD **converges significantly quicker than** I-BPI **to similar-valued FSCs for (c) multiagent tiger, and (d) ML problems. All were run on Linux with Intel Xeon 2.6GHz CPUs and 32GB RAM.**

$$\phi'_{j,0}(a_j^t, m_{j,0}^t) \propto \phi_{j,0}(a_j^t, m_j^t) \sum\nolimits_{s^t} R_{m_j}(s^t, a_j^t)\, \widehat{\alpha}(s^t, m_{j,0}^t)$$
$$+ \sum\nolimits_{s^t, s^{t+1}, m_{j,0}^{t+1}, o_j^{t+1}} \frac{\gamma}{1-\gamma} \widehat{\beta}(s^{t+1}, m_{j,0}^{t+1})\, \widehat{\alpha}(s^t, m_{j,0}^t)$$
$$\times T_{m_j}(s^t, a_j^t, s^{t+1})\, Pr(m_{j,0}^{t+1}|m_{j,0}^t, a_j^t, o_j^{t+1})\, O_{m_j}(s^{t+1}, a_j^t, o_j^{t+1})$$

At strategy levels, $l \geq 1$, we seek $\pi_{i,l}^*$ which maximizes the likelihood and it is iteratively obtained using EM.

E-step at level 1 In a multiagent setting, the hidden variables additionally include what the other agent may observe and how it acts over time. However, a key insight is that Proposition 1 allows us to limit attention to the conditional distribution over other agents' actions given the state. In the T-step DBN mixture model, observed evidence includes the reward, r_i^T, at the end and the initial belief. We seek the likely distributions, $\mathcal{V}_i$, $\mathcal{T}_i$, and $\mathcal{L}_i$, across time slices. We may again realize the full joint in the expectation using a forward-backward algorithm on a hidden Markov model whose state is $(s^t, n_{i,l}^t)$. The transition function of this model is,

$$Pr(s^t, n_{i,l}^t|s^{t-1}, n_{i,l}^{t-1}) = \sum\nolimits_{a_i^{t-1}, a_{-i}^{t-1}, o_i^t} \mathcal{L}_i(n_{i,l}^{t-1}, a_i^{t-1})\, \mathcal{T}_i(n_{i,l}^{t-1},$$
$$a_i^{t-1}, o_i^t, n_{i,l}^t)\, T_i(s^{t-1}, a_i^{t-1}, a_{-i}^{t-1}, s^t)\, O_i(s^t, a_i^{t-1}, a_{-i}^{t-1}, o_i^t)$$

The forward message, $\alpha^t = Pr(s^t, n_{i,l}^t)$, represents the probability of being at some state of the DBN at time t:

$$\alpha^t(s^t, n_{i,l}^t) = \sum_{s^{t-1}, n_{i,l}^{t-1}} Pr(s^t, n_{i,l}^t|s^{t-1}, n_{i,l}^{t-1})\, \alpha^{t-1}(s^{t-1}, n_{i,l}^{t-1})$$

where, $\alpha^0(s^0, n_{i,l}^0) = \mathcal{V}_i(n_{i,l}^0) b_{i,l}^0(s^0)$.

The backward message gives the probability of observing the reward in the final $T-1^{th}$ time step given some state of the Markov model, $\beta^t(s^t, n_{i,l}^t) = Pr(r_i^T = 1|s^t, n_{i,l}^t)$:

$$\beta^h(s^t, n_{i,l}^t) = \sum_{s^{t+1}, n_{i,l}^{t+1}} Pr(s^{t+1}, n_{i,l}^{t+1}|s^t, n_{i,l}^t)\, \beta^{h-1}(s^{t+1}, n_{i,l}^{t+1})$$

where, $\beta^0(s^T, n_{i,l}^T) = \sum_{a_i^T, a_{-i}^T} Pr(r_i^T = 1|s^T, a_i^T, a_{-i}^T) \times \mathcal{L}_i(n_{i,l}^T, a_i^T) \prod_{-i} Pr(a_{-i}^T|s^T)$, and $1 \leq h \leq T$ is the horizon. Here, $Pr(r_i^T = 1|s^T, a_i^T, a_{-i}^T) \propto R_i(s^T, a_i^T, a_{-i}^T)$.

M-step at level 1 We update the parameters, $\mathcal{L}_i$, $\mathcal{T}_i$ and $\mathcal{V}_i$, of $\pi_{i,l}$ to obtain $\pi'_{i,l}$ based on the expectation in the E-step. In order to update, $\mathcal{L}_i$, we partially differentiate the expected log likelihood with respect to $\mathcal{L}'_i$. $\mathcal{L}'_i$ on maximizing the log likelihood is:

$$\mathcal{L}'_i(n_{i,l}^t, a_i^t) \propto \mathcal{L}_i(n_{i,l}^t, a_i^t) \sum\nolimits_{T=0}^{\infty} \prod\nolimits_{-i} \sum\nolimits_{s^T, a_{-i}^T} \frac{\gamma^T}{1-\gamma}$$
$$\times Pr(r_i^T = 1|s^T, a_i^T, a_{-i}^T)\, Pr(a_{-i}^T|s^T)\, \alpha(s^T, n_{i,l}^T)$$

Node transition probabilities $\mathcal{T}_i$ and node distribution $\mathcal{V}_i$ for $\pi'_{i,l}$, is updated analogously to $\mathcal{L}_i$.

Block-Coordinate Descent for Speed Up Block-coordinate descent (BCD) [5] is an iterative scheme to gain faster non-asymptotic rate of convergence in the context of large-scale N-dimensional optimization problems. In this scheme, within each iteration, a set of variables referred to as coordinates are chosen and the objective function is optimized with respect to one of the coordinate blocks while the other coordinates are held *fixed*. We empirically show that grouping the number of time slices, t, and horizon, h, in computing α and β, respectively, at each level, into coordinate blocks of equal size is beneficial. *In other words, we decompose the mixture model into blocks containing equal numbers of Bayesian networks.*

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-BCD) and coupled with a greedy M-step (I-EM-BCD-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-BCD consistently improves on I-EM: the corresponding value improves 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-BCD converges to values similar to those of I-BPI almost *two orders of magnitude faster*. Beginning with 5 nodes, I-BPI adds 4 more nodes to obtain the same level of value as EM for the tiger problem. For money laundering, I-EM-BCD-Greedy converges to controllers whose value is at least 1.5 times better.

REFERENCES

[1] P. J. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multi-agent settings. *JAIR*, 24:49–79, 2005.

[2] B. Ng, C. Meyers, K. Boakye, and J. Nitao. Towards applying interactive POMDPs to real-world adversary modeling. In *IAAI*, pages 1814–1820, 2010.

[3] E. Sonu and P. Doshi. Scalable solutions of I-POMDPs using generalized and bounded policy iteration. *JAAMAS*, 2014.

[4] M. Toussaint and A. Storkey. Probabilistic inference for solving discrete and continuous state markov decision processes. In *ICML*, pages 945–952, 2006.

[5] P. Tseng and C. O. L. Mangasarian. Convergence of a block coordinate descent method for nondifferentiable minimization. *Opt. Theory Appl.*, pages 475–494, 2001.