Monte Carlo Bayesian Hierarchical Reinforcement Learning

(Extended Abstract)

Ngo Anh Vien
MLR Lab, University of Stuttgart, Germany
vien.ngo@ipvs.uni-stuttgart.de

Hung Ngo
IDSIA, USI-SUPSI, Switzerland
hung@idsia.ch

Wolfgang Ertel
Ravensburg-Weingarten University of Applied Sciences, Germany
ertel@hs-weingarten.de

ABSTRACT

In this paper, we propose to use hierarchical action decomposition to make Bayesian model-based reinforcement learning more efficient and feasible in practice. We formulate Bayesian hierarchical reinforcement learning as a partially observable semi-Markov decision process (POSMDP). The main POSMDP task is partitioned into a hierarchy of POSMDP subtasks; lower-level subtasks get solved first, then higher-level ones. We sample from a prior belief to build an approximate model for each POSMDP, then solve using Monte Carlo Value Iteration with Macro-Actions solver. Experimental results show that our algorithm performs significantly better than that of flat BRL in terms of both reward, and especially solving time, in at least one order of magnitude.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms,Performance

Keywords

Reinforcement Learning, Bayesian RL, Hierarchical BRL

1. INTRODUCTION

Bayesian Reinforcement Learning (BRL) can be reformulated as a partially observable Markov decision process (POMDP) problem [3]. The solution of POMDP, which is known to optimally disambiguating uncertainty, gives an optimal trade-off between exploration and exploitation in RL setting. Unfortunately, computing such Bayes-optimal behavior is intractable, severely limiting the applicability of the method. This paper aims to develop an alternative formulation which not only exploits the hierarchical structure of the problem, but also preserves the Bayesian optimality of the exploration/exploitation trade-off. We formulate Bayesian hierarchical RL (BHRL) as a partially observable semi-MDP (POSMDP). The main POSMDP task is partitioned into a hierarchy of smaller subtasks. Each subtask is again formulated as one POSMDP. We develop a Monte Carlo sampling-based algorithm to solve the POSMDP-formulated BHRL, called MC-BHRL, which first solves lower-level subtasks, then higher-level ones.

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

2. POSMDP FORMULATION FOR BHRL

As shown in [5], a POMDP with macro actions could be reformulated as a POSMDP. Here, we formulate BHRL as a hierarchy of multiple sub-POSMDPs, where a higher-level POSMDP consists of macro actions which are policies of lower-level POSMDP.

A POSMDP $\mathcal{P}$ is defined as a tuple $\langle \mathcal{S}_P, \mathcal{A}_P, \mathcal{O}_P, \mathrm{T}_P, \mathrm{R}_P \rangle$. The state space $\mathcal{S}_P = \{\mathcal{S}, \Theta\}$ consists of the MDP state space $\mathcal{S}$ and the unknown dynamics' parameter space Θ. Action space $\mathcal{A}_P$ consists of either primitive or macro actions. Observation space $\mathcal{O}_P = \mathcal{S}$. T_P is a joint distribution function of the state transition, the number of time steps k taken, and the observation, i.e., $\mathrm{T}_P(s, \theta, a, \theta', s', o, k) = p(\theta', s', k, o|s, \theta, a)$. The reward function is $\mathrm{R}_P(s, \theta, a, s', \theta') = R(s, a, s')$. The belief update, $b' = \tau(b, a, o)$, given an abstract action a and an observation o (i.e., s') is

$$b'(\theta') = Z_c \sum_{k=1}^{\infty} \gamma^{k-1} \int p(\theta', s', k, o|s, \theta, a) b(\theta) d\theta$$

where Z_c is a normalization constant. The Bellman's equations recursively updating V^π are

$$V_s^\pi(b) = r(b, s, a) + \gamma \sum_{o \in \mathcal{O}_P} p(o|\pi(b), b) V_{s'}^\pi(b').$$

The optimal policy π^* is defined to have the best value $V_s^*(b) \geq V_s^\pi(b), \forall \pi, b, s$, and can be found under the backup operator H:

$$HV_s(b) = \max_a \Big\{ r(b, s, a) + \gamma \sum_{o \in \mathcal{O}_P} p(o|a, b) V_{s'}(b') \Big\}.$$

We proved contraction and fixed-point properties of the new POSMDP formulation. These are followed by the piecewise linear and convex property.

LEMMA 1. *The backup operator is a contraction mapping,*

$$\|HU - HV\|_\infty \leq \gamma \|U - V\|_\infty$$

where U and V are two value functions.

THEOREM 2.1. *The optimal value function V^* is a single fixed point of the backup operator H: $V^* = HV^*$.*

THEOREM 2.2. *The t-step optimal value function is convex and piecewise linear, which is represented as*

$$V_t(b) = \sup_{\alpha_t^i \in \Gamma_t} \langle \alpha_t^i, b \rangle$$

Table 1: Taxi domain. We report the average rewards, success rates, and the offline and online running time in seconds.

Algorithm	Rew.	Suc.	Offline	Online
MC-BHRL ($M = 1K$)	-14.42	74%	2000	4
MC-BHRL ($M = 2K$)	-10.29	85%	4000	4
MC-BRL ($M = 1K$)	-19.76	11%	2000	4
	-19.20	12%	10000	5
MC-BRL ($M = 2K$)	-15.01	15%	4000	4
	-14.39	17%	10000	5

where Γ_t is a continuous set of α-functions $\alpha_t^i : \mathcal{S}_P \to \Re$.

3. MONTE CARLO BHRL

MC-BHRL is a recursive Monte-Carlo sampling-based method for solving BHRL formulated as POSMDP. We adopt the similar idea from MC-BRL method [9] to sample from a prior to approximate the exact continuous POSMDP, then solve the approximated POSMDP using the Macro-MCVI solver [5, 1]. The subtasks are solved in bottom-up ordering as in Algorithm 1. To solve each POSMDP $\mathcal{P}$ in the offline stage, we sample M primitive models from a prior distribution $b_0(\theta)$, then create a new approximate POSMDP $\hat{\mathcal{P}} = \langle \mathcal{S}_{\hat{P}}, \mathcal{A}_{\hat{P}}, \mathcal{O}_{\hat{P}}, \mathrm{T}_{\hat{P}}, \mathrm{R}_{\hat{P}}, \gamma, b^0_{\hat{P}} \rangle$. This approach is easily extended to apply for RL setting in POMDP environments, similar to the method in [9].

Algorithm 1 MC-BHRL Planning

1: **Require:** An action hierarchy $\mathcal{H}$.
2: **for** each subtask $a \in \mathcal{H}$ **do** solve in bottom-up ordering
3: Formulate a POSMDP $\mathcal{P}$ with lower-level macro actions.
4: Sample M primitive models $\{\hat{\theta}^1, \cdots, \hat{\theta}^M\} \sim b_0(\theta, \mathcal{P})$.
5: Form a new approximate POSMDP $\hat{\mathcal{P}}$.
6: Use Macro-MCVI to solve POSMDP $\hat{\mathcal{P}}$ for a policy $\hat{\pi}^*_a$.
7: **end for**
8: **return** $\hat{\pi}^*_a,\ \forall a \in \mathcal{H}$.

4. EXPERIMENTS

We empirically evaluate the performance of MC-BHRL on three domains. First, we use a familiar variant of Taxi domain from [2] which is a fully observable MDP task. The other two are RL tasks in POMDP: Cheese-Taxi [6] and large Cheese-Taxi [7].

For Taxi domain, the results for comparison are reported in Table 1. The online time is for running 500 episodes. The performance results of MC-BRL and MC-BHRL are averaged over 10 offline simulations. Each simulation is online evaluated with 500 trials (episodes) to report an average total reward. This is a difficult task for a BRL algorithm due to a very large continuous state space (400-dimensional), so MC-BHRL could not find a near optimal policy in limited time. Since MC-BRL could not find a better policy than a random policy even after quite long time, MC-BHRL still shows promising results over MC-BRL in terms of running time, success rate and average reward. The average reward is still less than zero while the success rate of MC-BHRL is high. This is partly because we let the macro action run until reaching its maximum length if not terminated (a movement takes a cost of -1.0), and partly because it found a non-shortest path at each success.

For the Cheese-Taxi and large Cheese-Taxi domain, the comparison results are reported in Table 2. PolCA [6] and Flat-DDN [7] are hierarchical POMDP solvers; their performance results are with an assumption of a known POMDP. The Cheese-Taxi problem is small enough so that both algorithms could quickly find a near-optimal policy. However, MC-BHRL with a pre-defined action hierarchy is hundred times faster than MC-BRL, a flat BRL solver.

Table 2: Cheese-Taxi and Large Cheese-Taxi domains.

Algorithm	Rewards	Time (seconds)
Cheese-Taxi		
MC-BHRL ($M = 10$)	$\mathbf{8.434 \pm 0.16}$	2
MC-BRL ($M = 10$)	-32.4 ± 2.24	2
	6.257 ± 0.10	15
	8.223 ± 0.09	450
PolCA	$\backsim 8.50$	N/A
Large Cheese-Taxi		
MC-BHRL ($M = 1000$)	-7.56 ± 1.16	1000
MC-BHRL ($M = 2000$)	$\mathbf{4.23 \pm 0.49}$	1000
MC-BRL ($M = 1000$)	-22.6 ± 5.53	1000
	-22.0 ± 5.70	10000
MC-BRL ($M = 2000$)	-20.4 ± 4.46	1000
	-18.6 ± 5.09	10000
Flat-DDN	8.40	N/A

There is a performance gap between MC-BHRL and an optimal policy of Flat-DDN, though MC-BHRL has 100% completed the tasks. However it found longer paths due to the use of policy graph. Each time a NAVIGATE macro action is used, the agent implementing NAVIGATE's policy graph starts as if it just starts to do disambiguation, and ignores all knowledge from previous called NAVIGATE(s). This problem might be resolved with modifications of Macro-MCVI solver.

5. DISCUSSION & CONCLUSION

We have proposed an efficient and simple method to solve BRL problems by exploiting action decomposition. We formulated the underlying BHRL problem as a POSMDP, which is then approximated and solved by Macro-MCVI. There are a number of potential extensions to MC-BHRL. An automatic hierarchy discovery may also be integrated into MC-BHRL. With POSMDP formulation for BHRL, online POSMDP solvers can be applied to solve BHRL, e.g. Monte Carlo Tree Search approaches [8, 4].

6. REFERENCES

[1] H. Bai, D. Hsu, W. S. Lee, and N. A. Vien. Monte carlo value iteration for continuous-state pomdps. In *WAFR*, pages 175–191, 2010.

[2] T. G. Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *J. Artif. Intell. Res. (JAIR)*, 13:227–303, 2000.

[3] M. Duff. *Optimal learning: Computational procedures for Bayes-adaptive Markov decision processes*. PhD thesis, University of Massassachusetts Amherst, 2002.

[4] A. Guez, D. Silver, and P. Dayan. Efficient bayes-adaptive reinforcement learning using sample-based search. In *NIPS*, pages 1034–1042, 2012.

[5] Z. W. Lim, D. Hsu, and L. W. Sun. Monte Carlo value iteration with macro-actions. In *NIPS*, pages 1287–1295, 2011.

[6] J. Pineau. *Tractable Planning Under Uncertainty: Exploiting Structure*. PhD thesis, Robotics Institute, Carnegie Mellon University, 2004.

[7] W. H. Turkett. *Robust Multiagent Plan Generation and Execution with Decision Theoretic Planners*. PhD thesis, Department of Computer Science and Engineering, University of South Carolina, 1998.

[8] N. A. Vien and W. Ertel. Monte carlo tree search for bayesian reinforcement learning. In *ICMLA (1)*, pages 138–143, 2012.

[9] Y. Wang, K. S. Won, D. Hsu, and W. S. Lee. Monte carlo bayesian reinforcement learning. In *ICML*, 2010.